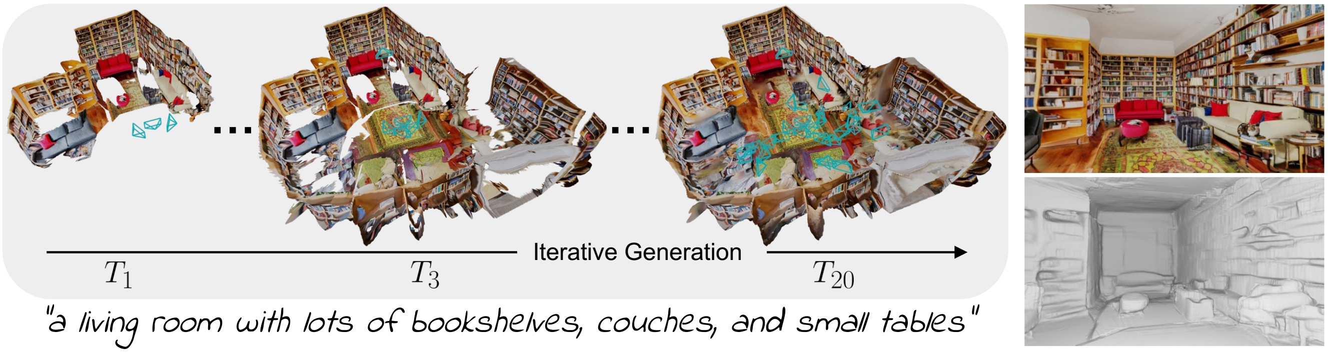
Text2Room: Extracting Textured 3D Meshes from 2D Text-to-Image Models

Lukas Höllein[1]* Ang Cao[2]* Andrew Owens[2] Justin Johnson[2] Matthias Nießner[1]
[1]Technical University of Munich [2]University of Michigan


T_1
T_3
Iterative Generation
T_{20}
"a living room with lots of bookshelves, couches, and small tables"

(a) 3D Mesh Generation from Text

(b) Rendered Image + Mesh

Figure 1. **Textured 3D mesh generation from text prompts.** We generate textured 3D meshes from a given text prompt using 2D text-to-image models. (a) The scene is iteratively created from different viewpoints (marked in blue). (b) Our generated mesh contains compelling textures and geometry. We remove the ceiling in the top-down views for better visualization of the scene layout.

Abstract

We present Text2Room†, a method for generating room-scale textured 3D meshes from a given text prompt as input. To this end, we leverage pre-trained 2D text-to-image models to synthesize a sequence of images from different poses. In order to lift these outputs into a consistent 3D scene representation, we combine monocular depth estimation with a text-conditioned inpainting model. The core idea of our approach is a tailored viewpoint selection such that the content of each image can be fused into a seamless, textured 3D mesh. More specifically, we propose a continuous alignment strategy that iteratively fuses scene frames with the existing geometry to create a seamless mesh. Unlike existing works that focus on generating single objects [57, 42] or zoom-out trajectories [18] from text, our method generates complete 3D scenes with multiple objects and explicit 3D geometry. We evaluate our approach using qualitative and quantitative metrics, demonstrating it as the first method to generate room-scale 3D geometry with compelling textures from only text as input.

1. Introduction

Mesh representations of 3D scenes are a crucial component for many applications, from AR/VR asset creation to computer graphics, yet creating these 3D assets remains a painstaking process that requires considerable expertise. In the 2D domain, recent works have successfully created high-quality images from text using generative models, such as diffusion models [66, 59, 68]. These methods significantly reduce the barriers to creating images that contain a user's desired content, effectively helping towards the democratization of content creation. An emerging line of work has sought to apply similar methods to create 3D models from text [9, 57, 30, 42, 39], yet existing approaches come with a number of significant limitations and lack the generality of 2D text-to-image models.

One of the core challenges of generating 3D models is coping with the lack of available 3D training data, as 3D datasets are vastly smaller than those available in many other applications, such as 2D image synthesis. For example, methods that directly use 3D supervision, such as Chen *et al.* [9], are often limited to datasets of simple shapes, such as ShapeNet [8]. To address these data limitations, recent methods [57, 30, 42, 39, 89] lift the expressive power of 2D text-to-image models into 3D by formulating 3D generation as an iterative optimization problem in the image domain. This allows them to generate 3D ob-

* joint first authorship
† https://lukashoel.github.io/text-to-room

jects stored in a radiance field representation, demonstrating the ability to generate arbitrary (neural) shapes from text. However, these methods cannot easily be extended to create room-scale 3D structure and texture. The challenge of generating large scenes is ensuring that the generated output is dense and coherent across outward-facing viewpoints, and that these views contain all of the required structures, such as walls, floors, and furniture. Additionally, a mesh remains a desired representation for many end-user tasks, such as rendering on commodity hardware (which requires an additional conversion step as presented in Lin *et al.* [42]).

To address these shortcomings, we propose a method that extracts scene-scale 3D meshes from off-the-shelf 2D text-to-image models. Our method iteratively generates a scene through inpainting and monocular depth estimation. We produce an initial mesh by generating an image from text, and backproject it into 3D using a depth estimation model. Then, we iteratively render the mesh from novel viewpoints. From each one, we fill in holes in the rendered images via inpainting, then fuse the generated content into the mesh (Fig. 1a).

Our iterative generation scheme has two important design considerations: how we choose the viewpoints, and how we merge generated scene content with the existing mesh. We first select viewpoints from predefined trajectories that will cover large amounts of scene content, then adaptively select viewpoints that close remaining holes. When merging generated content with the mesh, we align the two depth maps to create smooth transitions, and remove parts of the mesh that contain distorted textures. Together, these decisions lead to large, scene-scale 3D meshes with compelling textures and consistent geometry (Fig. 1b), that can represent a wide range of rooms.

To summarize, our contributions are:

- Generating 3D meshes of room-scale indoor scenes with compelling textures and geometry from any text input.
- A method that leverages 2D text-to-image models and monocular depth estimation to lift frames into 3D in an iterative scene generation. Our proposed depth alignment and mesh fusion steps, enable us to create seamless and undistorted geometry and textures.
- A two-stage tailored viewpoint selection that samples camera poses from optimal positions to first create the room layout and furniture and then close any remaining holes, creating a watertight mesh.

2. Related Work

Text-based Generation has seen significant advances due to large-scale image-text datasets [74, 73, 14, 72] and scalable generative model architectures [16, 67, 61, 33], enabling synthesis of novel images from text [20, 3, 55].

Recently, diffusion models [79, 24, 81, 82, 83] achieved impressive results on image synthesis [15, 66, 68, 51, 59] through improvements like latent space denoising [66, 85], faster sampling [24, 80, 53, 35], and better guidance [25].

In particular, *text-to-image* methods like Stable Diffusion [66], Imagen [68], GLIDE [51] and DALL·E 2 [59] yield diverse, high-fidelity, and controllable [6, 95] outputs. Text-based generation has been extended to other modalities including audio [36, 17, 29, 71], video [76, 92, 86, 26], and 4D fields [77]. We use *text-to-image* models by lifting their generated output into complete 3D scene meshes.

Text-to-3D. Several methods use 3D data for supervised training of text-to-3D models [9, 52, 5]; however this direction remains challenging due to the lack of large-scale aligned datasets of text and 3D.

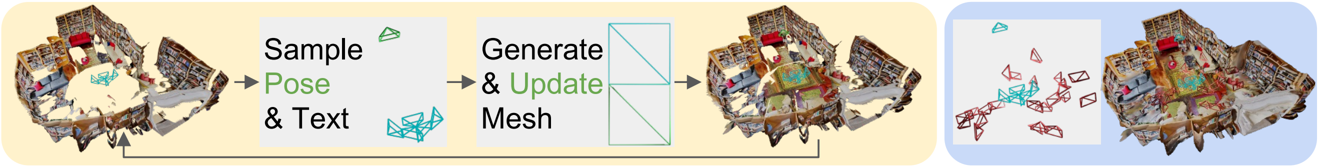
Alternative approaches use 2D vision-language models like CLIP [58] to create 3D content by formulating the generation as an optimization problem in the image domain [87, 30, 39, 49, 31] or as object alignment [70]. Related methods refine existing 3D input through text guidance in a similar fashion [47, 10, 88, 63].

Recent methods [57, 42, 46, 89, 45] combine large text-to-image diffusion models [66, 68] and neural radiance fields [48] to generate 3D objects without training. Other approaches train custom diffusion models on a similar text-to-3D task [40, 50, 12]. In contrast, we use a fixed text-to-image model and extract a 3D mesh representing entire scenes of many objects and structural elements like walls.

3D-Consistent View Synthesis from a Single Image. Several methods have been proposed that perform novel-view-synthesis from a single image [65, 91, 75, 62, 19]. Others optimize a neural 3D representation of an object, that can be viewed from arbitrary novel view points [93, 90, 1]. Another line of work performs *perpetual view generation* [78, 43, 41, 7], synthesizing videos via a *render-refine-repeat* pattern from a single RGB image that depict a scene along a forward-facing camera trajectory. In very recent concurrent work, Fridman *et al.* [18] create 3D scenes from text, but focus on this type of 3D-consistent "zoom-out" video generation. Instead, we generate complete, textured 3D room geometry from arbitrary trajectories.

3. Method

Our method creates a textured 3D mesh of a complete scene from text input. To this end, we continuously fuse generated frames from a 2D text-to-image model at different poses into a joint 3D mesh, creating the scene over time. The core idea of our approach is a two-stage tailored viewpoint selection, that first generates the scene layout and objects and then closes remaining holes in the 3D geometry (Section 3.4). We visualize this workflow in Figure 2. For each pose in both stages, we apply an iterative scene generation scheme to update the mesh (Section 3.1). We first align each frame with the existing geometry with a depth align-



(a) Scene Generation Stage

(b) Scene Completion Stage

Figure 2. **Method overview**. We iteratively create a textured 3D mesh in two stages. (a) First, we sample predefined poses and text to generate the complete scene layout and furniture. Each new pose (marked in green) adds newly generated geometry to the mesh (depicted by green triangles) in an iterative scene generation scheme (see Figure 3 for details). Blue poses/triangles denote viewpoints that created geometry in a previous iteration. (b) Second, we fill in the remaining unobserved regions by sampling additional poses (marked in red) after the scene layout is defined.

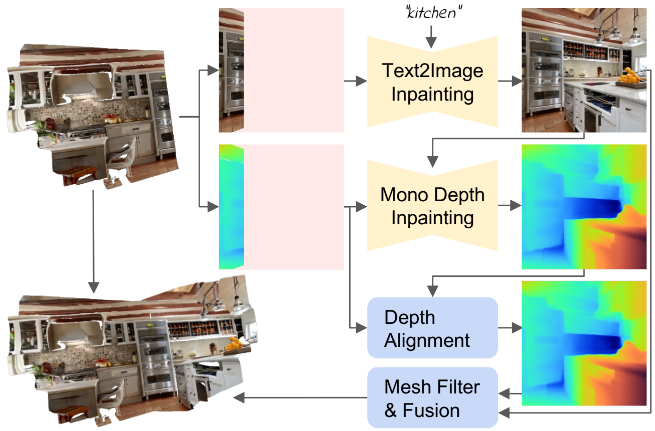


Figure 3. **Iterative scene generation**. For each new pose, we render the current mesh to obtain partial RGB and depth renderings. We complete both, utilizing respective inpainting models and the text prompt. Next, we perform depth alignment (see Section 3.2) and mesh filtering (see Section 3.3) to obtain an optimal next mesh patch, that is finally fused with the existing geometry.

ment strategy (Section 3.2). Next, we triangulate and filter the novel content to merge it into the mesh (Section 3.3).

3.1. Iterative 3D Scene Generation

Our scene is represented as a mesh $\mathcal{M} = (\mathcal{V}, \mathcal{C}, \mathcal{S})$ where the vertices $\mathcal{V} \in \mathbb{R}^{N\times 3}$, vertex colors $\mathcal{C} \in \mathbb{R}^{N\times 3}$ and the face set $\mathcal{S} \in \mathbb{N}_0^{M\times 3}$ are generated over time. Input to our method is a set of arbitrary text prompts $\{P_t\}_{t=1}^{T}$ that corresponds to our selected poses $\{E_t\}_{t=1}^{T} \in \mathbb{R}^{3\times 4}$ in both stages. Inspired by recent methods [43, 41], we iteratively build up the scene, following a *render-refine-repeat* pattern. We summarize this iterative scene generation process in Figure 3. Formally, for each step of generation t, we first render the current scene from a novel viewpoint:

$$I_t, d_t, m_t = r(\mathcal{M}_t, E_t), \tag{1}$$

where r is a classical rasterization function without shading, I_t is the rendered image, d_t the rendered depth and m_t the image-space mask, that marks pixels without observed content. We then use a fixed text-to-image model $\mathcal{F}_{t2i}$ to inpaint unobserved pixels according to the text prompt:

$$\hat{I}_t = \mathcal{F}_{t2i}(I_t, m_t, P_t). \tag{2}$$

Next, we inpaint unobserved depth by applying a monocular depth estimator $\mathcal{F}_d$ in our depth alignment (see Section 3.2):

$$\hat{d}_t = \textit{predict-and-align}(\mathcal{F}_d, I_t, d_t, m_t). \tag{3}$$

Finally, we combine the novel content $\{\hat{I}_t, \hat{d}_t, m_t\}$ with the existing mesh by our fusion scheme (see Section 3.3):

$$\mathcal{M}_{t+1} = \textit{fuse}(\mathcal{M}_t, \hat{I}_t, \hat{d}_t, m_t, E_t). \tag{4}$$

3.2. Depth Alignment Step

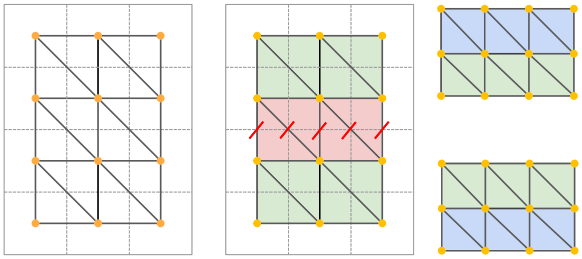
To lift a 2D image I into 3D, we predict the per-pixel depth. To correctly combine old and new content, it is necessary that both align with each other. In other words, similar regions in a scene like walls or furniture should be placed at similar depth. However, directly using the predicted depth for backprojection leads to hard cuts and discontinuities in the 3D geometry, since the depth is inconsistent in scale between subsequent viewpoints (see Figure 7a).

To this end, we perform depth alignment in two-stages. First, we use a state-of-the-art depth inpainting network [4] that takes ground-truth depth d for known parts in the image as input and aligns the prediction to it: $\hat{d}_p = \mathcal{F}_d(I, d)$.

Inspired by Liu *et al.* [43] we then improve the result by optimizing for scale and shift parameters $\gamma, \beta \in \mathbb{R}$, aligning predicted and rendered disparity in the least squares sense:

$$\min_{\gamma,\beta} \left\| m \odot \left(\frac{\gamma}{\hat{d}_p} + \beta - \frac{1}{d} \right) \right\|^2, \tag{5}$$

where we mask out unobserved pixels via m. We can then extract the aligned depth as $\hat{d} = (\frac{\gamma}{\hat{d}_p} + \beta)^{-1}$. Finally, we smooth $\hat{d}$ by applying a 5×5 Gaussian kernel at the mask edges (see supplemental material for more details).



(a) Pixel Triangulation (b) Face Filtering (c) Mesh Fusion

Figure 4. **Visualization of our mesh fusion step.** (a) We triangulate an image, such that 4 neighboring pixels (orange dots) create two faces. (b) We filter a face (marked in red), if its surface normal forms a small grazing angle with the viewing direction or if any edge in world space is too long. (c) We fuse the remaining faces (marked in green) with the existing geometry (marked in blue).

3.3. Mesh Fusion Step

At each step, we insert new content $\{\hat{I}_t, \hat{d}_t, m_t\}$ into the scene. For that, we first backproject the image-space pixels into a world-space point cloud:

$$\mathcal{P}_t = \{E_t^{-1} K^{-1} \cdot \hat{d}_t[u,v] \cdot (u,v,1)^T\}_{u=0,v=0}^{W,H}, \qquad (6)$$

where $K \in \mathbb{R}^{3\times3}$ are the camera intrinsics and W, H are image width and height, respectively. We then use a simple triangulation scheme (Figure 4a), where each four neighboring pixels $\{(u,v), (u{+}1,v), (u,v{+}1), (u{+}1,v{+}1)\}$ in the image form two triangles. Since the estimated depth is noisy, this naïve triangulation creates stretched out 3D geometry (see Figure 7b). To alleviate this problem, we propose two filters that remove stretched out faces (Figure 4b).

First, we filter faces based on their edge length. We remove a face if the Euclidean distance of any face edge is larger than a threshold δ_{edge}. Second, we filter faces based on the angle between surface normal and viewing direction:

$$\mathcal{S} = \{(i_0, i_1, i_2) | n^T v > \delta_{sn}\} \qquad (7)$$

where $\mathcal{S}$ is the face set, (i_0, i_1, i_2) are the vertex indices of the triangle, δ_{sn} is the threshold, $n \in \mathbb{R}^3$ is the normalized face normal, and $v \in \mathbb{R}^3$ is the normalized view direction in world space from the camera center towards the average pixel location from which the triangle originated. This avoids creating texture for large regions of the mesh from a comparatively small number of pixels from an image.

Finally, we fuse together the newly generated mesh patch and the existing geometry (Figure 4c). All faces that are backprojected from pixels falling into the inpainting mask m_t are stitched together with their neighboring faces, which are already part of the mesh. Precisely, we continue the triangulation scheme at all edges of m_t, but use the existing vertex positions of $\mathcal{M}_t$ to create the corresponding faces.

3.4. Two-Stage Viewpoint Selection

A key part of our method is the choice of text prompts and camera poses from which the scene is synthesized. Users can in principle choose these inputs arbitrarily to create any desired indoor scene. However, the generated scene can degenerate and contain stretch and hole artifacts, if poses are chosen carelessly (see Figure 7 and supplemental material). To this end, we propose a two-stage viewpoint selection strategy, that samples each next camera pose from optimal positions and refines empty regions subsequently.

Generation Stage. In the first stage, we create the main parts of the scene, including the general layout and furniture. We subsequently render *predefined* trajectories in different directions that eventually cover the whole room. We found generation works best, if each trajectory starts off from a viewpoint with mostly unobserved regions. This generates the outline of the next chunk, while still being connected to the rest of the scene (e.g., see Figure 3). Then, we complete the 3D structure of that chunk by moving and rotating into it subsequently until the end of the trajectory.

Additionally, we ensure an optimal observation distance for each pose. We translates camera positions $T_0 {\in} \mathbb{R}^3$ along the look-at direction $L {\in} \mathbb{R}^3$ uniformly: $T_{i+1} {=} T_i {-} 0.3L$. We stop if the mean rendered depth is larger than 0.1 or discard the camera after 10 steps. This avoids views too close to existing geometry. For example, the green pose in Figure 2a is moved back as far as possible into the existing geometry such that it views most of the empty floor region.

We create closed room layouts following this principle, by choosing trajectories that generate the next chunks in a circular motion, roughly centered around the origin. We found it helpful to discourage the text-to-image generator from generating furniture in unwanted regions by engineering the text prompts accordingly. For example, for poses looking at the floor or ceiling, we choose text prompts that only contain the words "floor" or "ceiling", respectively.

Completion Stage. After the first stage, the scene layout and furniture is defined. However, it is impossible to choose sufficient poses *a-priori*. Since the scene is generated on-the-fly, the mesh contains holes that were not observed by any camera (see Figure 7c). We complete the scene by sampling additional poses *a-posteriori*, looking at those holes.

Inspired by trajectory optimization [23, 64], we voxelize the scene into dense uniform cells. We sample random poses in each cell, discarding those being too close to existing geometry. We select one pose per cell that views most unobserved pixels (e.g., see the red poses in Figure 2b).

Next, we inpaint the scene from all chosen camera poses following Section 3.1. Similar to Fridman *et al.* [18], we observe it is important to clean the inpainting masks, because our text-to-image generator can generate better results for large connected regions. Thus, we first inpaint small holes with a classical inpainting algorithm [84] and dilate the remaining holes. We additionally remove all faces that fall into the dilated region and are close to the rendered depth. Please see the supplemental material for more details.

Finally, we run Poisson surface reconstruction [34] on the scene mesh. This closes any remaining holes after completion and smoothens out discontinuities. The result is a watertight mesh of the generated scene, that can be rendered with classical rasterization.

4. Results

Implementation Details. We implement mesh rasterization and fusion with Pytorch3D [60]. As our text-to-image model $\mathcal{F}_{t2i}$, we utilize a Stable Diffusion [66] model, that is finetuned on the image inpainting task, using additional mask input. We generate a single inpainting proposal and employ a state-of-the-art guided diffusion sampler [44]. As our monocular depth estimator $\mathcal{F}_d$, we employ an IronDepth [4] model, that is trained on indoor scenes from the ScanNet dataset [13] and augment it for depth inpainting according to Bae *et al.* [4]. We set δ_{edge}=0.1 and δ_{sn}=0.1 in all our experiments. During generation, we use 20 different trajectories with 10 frames each sampled between the respective start and end poses. We construct prompts using the guidelines suggested by Pierre [56]. Creating one scene takes approximately 50 minutes on one RTX 3090 GPU.

Baselines. To the best of our knowledge, there are no direct baselines that generate textured 3D room geometry from text. We compare against four related methods (please see the supplemental material for more details about baselines).

- *PureClipNeRF* [39]: We compare against text-to-3D methods for generating objects [57, 42, 30, 39, 89] and choose Lee *et al.* [39] as open-source representative.
- *Outpainting* [59, 54]: We combine outpainting from a Stable Diffusion [66] model with depth estimation and triangulation to create a mesh from an enlarged viewpoint.
- *Text2Light* [11]: We generate RGB panoramas from text using Chen *et al.* [11]. Estimating 3D mesh structure from a panorama is difficult. Related approaches estimate room layout [94], perform view synthesis [37, 27, 22, 28] or predict 360° depth [2, 32]. We perform depth prediction and subsequently apply our mesh fusion step.
- *Blockade* [38]: We apply *Blockade* [38], which uses a text-to-image diffusion model to produce more expressive RGB panoramas. We then extract the mesh similarly.

Evaluation Metrics. The generated 3D geometry is evaluated both quantitatively and qualitatively. We calculate CLIP Score (CS) [58] and Inception Score (IS) [69] on RGB renderings of the respective scenes. Additionally, we conduct a user study and ask n=61 users to score Perceptual Quality (*PQ*) and 3D Structure Completeness (*3DS*) of the whole scene on a scale of $1-5$.

Method	2D Metrics		User Study	
	CS ↑	IS ↑	PQ ↑	3DS ↑
PureClipNeRF [39]	24.06	1.26	2.34	2.38
Outpainting [59, 54]	23.10	1.60	2.90	2.58
Text2Light [11]+Ours	25.99	2.21	2.82	2.97
Blockade [38]+Ours	26.29	2.13	3.35	3.36
Ours w/o alignment	26.73	1.78	3.12	2.96
Ours w/o stretch removal	27.72	1.86	3.28	3.75
Ours w/o completion	27.97	2.18	3.72	3.87
Ours	**28.02**	**2.31**	**4.01**	**4.19**

Table 1. **Quantitative comparison.** We report 2D metrics and user study results, including: Clip Score (*CS*), Inception Score (*IS*), Perceptual Quality (*PQ*), and 3D Structure Completeness (*3DS*). Our method creates scenes with the highest quality.

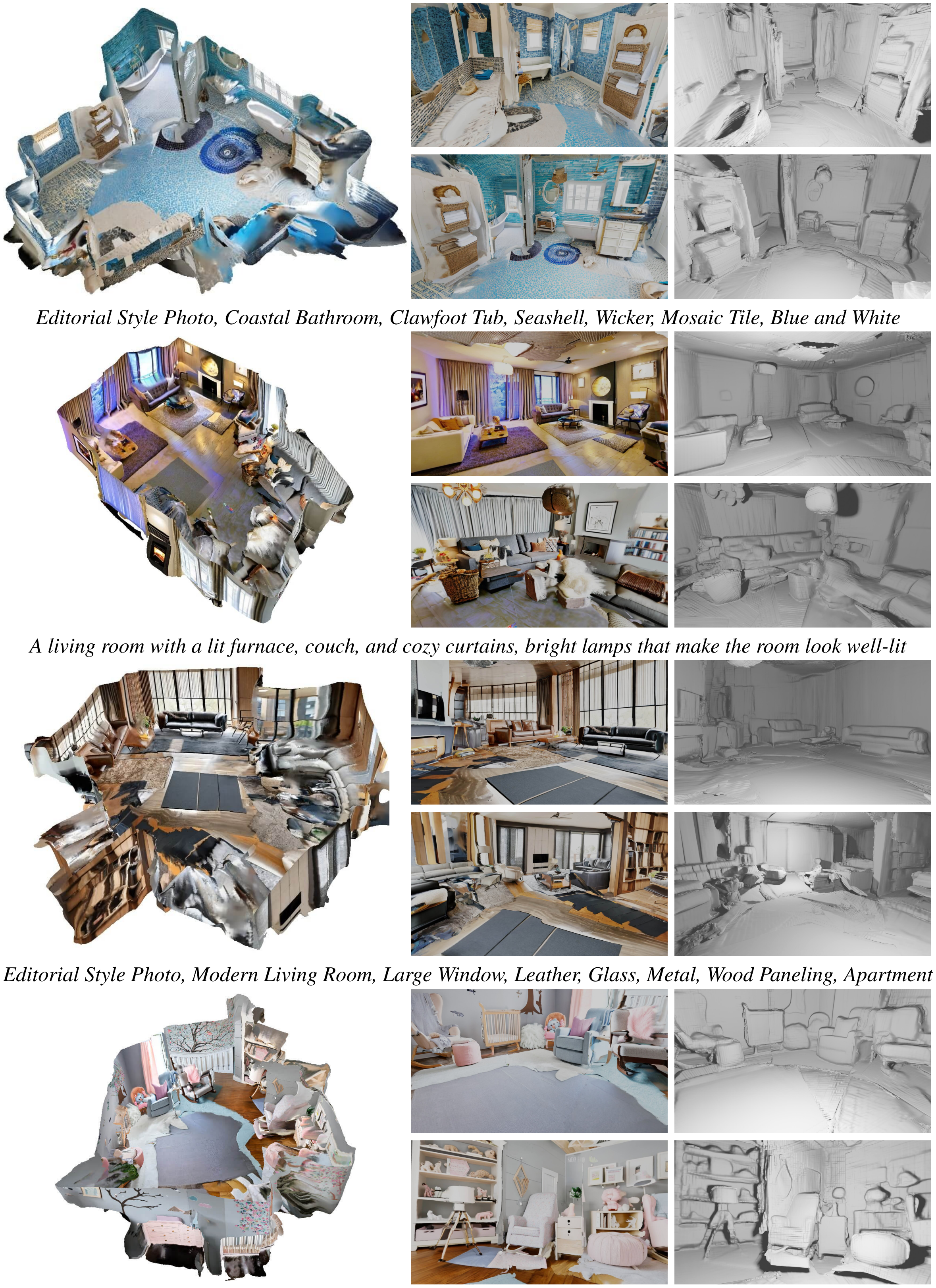
4.1. Qualitative Results

We show top-down views into the scene and RGB renderings from within for our method and baselines in Figure 6. We show additional results of our method in Figure 5. *PureClipNeRF* [39] creates the key objects of the given text prompt, but does not create a complete 3D structure with floor, walls and ceilings. *Outpainting* [59, 54] creates high-detail textures, but projection from a single viewpoint creates holes due to occlusion and hinders the creation of complete 3D geometry. *Text2Light* [11] and *Blockade* [38] both create a high-detail 360° view of a complete scene, but occlusions that cannot be resolved from a single panoramic viewpoint lead to holes in the extracted 3D geometry.

In contrast, our approach creates high-detail textures and geometry, that are fused into a complete 3D scene mesh without holes. The resulting scenes contain flat floors, walls and ceilings, as well as 3D object geometry distributed throughout the scene. When specifying text prompts with a huge variety, the resulting scene contains a diverse set of objects. Please see the supplemental material for more scenes, animated results, intermediate outputs of our baselines (such as the panoramic images) as well as top-down views of meshes, that contain the reconstructed ceilings.

4.2. Quantitative Results

We show quantitative results averaged over multiple scenes in Table 1. We render 60 images from novel viewpoints for each scene to calculate the 2D metrics. We present users with multiple top-down views and renderings for each scene and let them rate each method individually (no side-by-side comparison). Stretched-out geometry and holes in the 3D geometry lead to lower scores for the baselines in all image-based metrics. Our approach achieves the highest scores, because the renderings are complete from arbitrary novel poses, satisfy the given text-prompt and



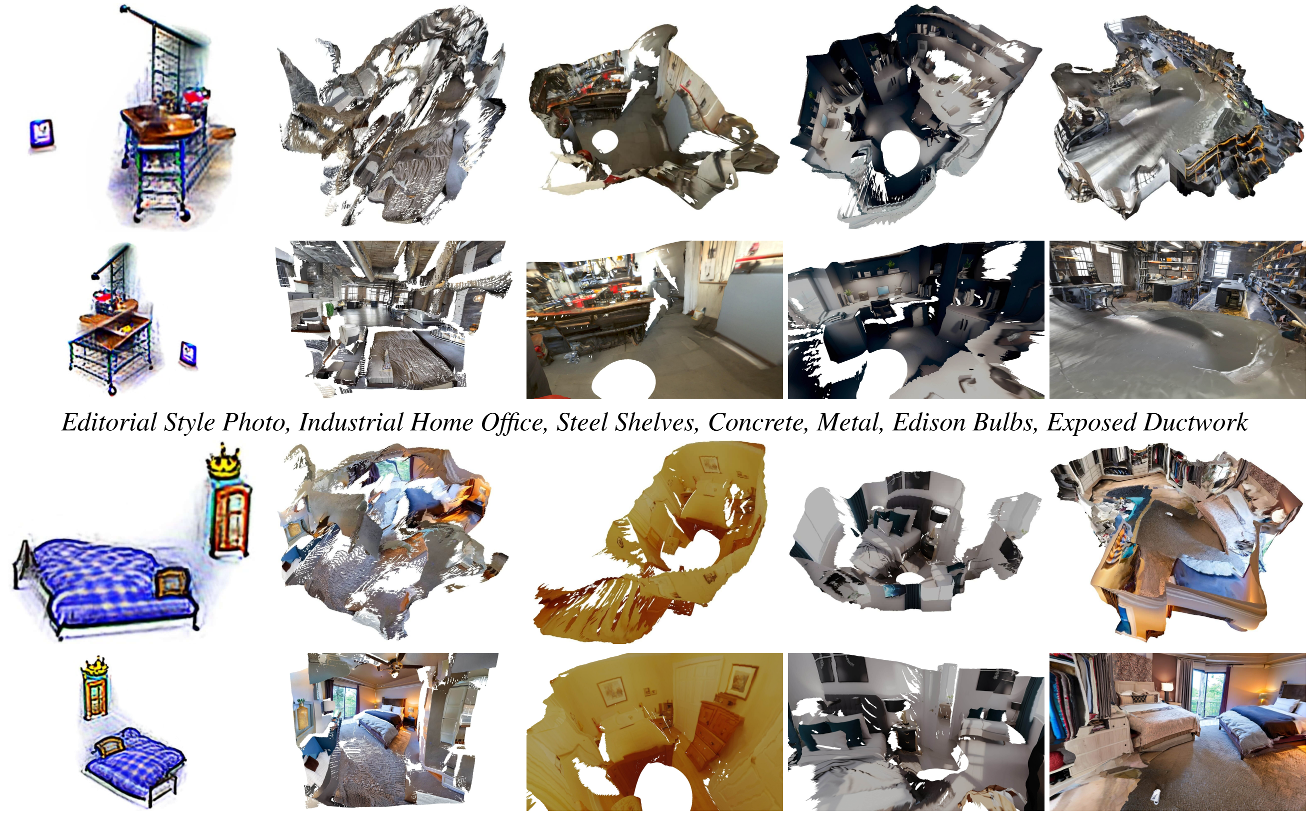
Editorial Style Photo, Coastal Bathroom, Clawfoot Tub, Seashell, Wicker, Mosaic Tile, Blue and White

A living room with a lit furnace, couch, and cozy curtains, bright lamps that make the room look well-lit

Editorial Style Photo, Modern Living Room, Large Window, Leather, Glass, Metal, Wood Paneling, Apartment

Editorial Style Photo, Modern Nursery, Table Lamp, Rocking Chair, Tree Wall Decal, Wood, Cotton, Faux Fur

Figure 5. **3D scene generation results of our method.** We show color and shaded geometry renderings from generated scenes with corresponding text prompts. Our method synthesizes realistic meshes satisfying text descriptions. We remove the ceiling in the top-down view for better visualization of the scene layout.



Editorial Style Photo, Industrial Home Office, Steel Shelves, Concrete, Metal, Edison Bulbs, Exposed Ductwork

a bedroom with a king-size bed and a large wardrobe

PureClipNeRF [39] Outpainting [59, 54] Text2Light [11]+Ours Blockade [38]+Ours Ours

Figure 6. **Qualitative comparison of our method and baselines.** *PureClipNeRF* [39] cannot produce immersive scenes with floors and walls. *Outpainting* [59, 54] does not produce 3D consistent scenes. *Text2Light* [11] and *Blockade* [38] both have holes due to occlusions. In contrast, our method creates complete meshes without holes and high details. We remove the ceiling in the top-down view for better visualization of the scene layout.

contain high-resolution image features. Users prefer our method, which highlights the quality of our accurate and complete geometry, as well as the RGB texture.

4.3. Ablations

The key ingredients of our method are depth alignment (Section 3.2), mesh fusion (Section 3.3) and the two-stage viewpoint selection (Section 3.4). We demonstrate the importance of each component in Figure 7 and Table 1.

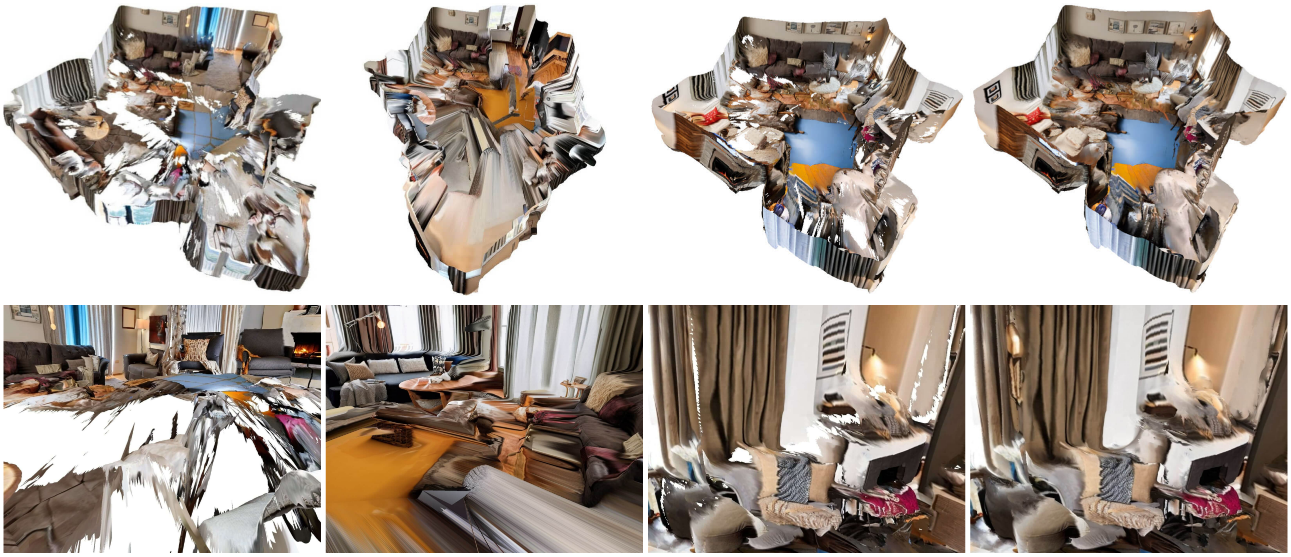
Depth alignment creates seamless scenes. Monocular depth predictions from subsequent frames can be inconsistent in scale. This leads to disconnected components in the mesh that are backprojected from multiple viewpoints (see Figure 7a). Our depth alignment strategy allows fusing multiple frames into a seamless mesh, eventually creating a complete scene with flat floors, walls, ceilings and no holes.

Stretch removal creates undistorted scene geometry. During mesh fusion, we update the scene geometry with the contents of the next frame. Due to noisy depth prediction, the objects become stretched out, if they are observed from small grazing angles. Thus, we propose two filters (edge length and surface normal thresholds) that alleviate this issue. Instead of baking in stretched-out geometry (see Figure 7b), we disregard the corresponding faces and let the object be completed from a more suitable, later viewpoint.

Two-stage generation creates complete scenes. Our approach chooses camera poses in two stages to create a complete scene without holes. After generating the scene from predefined trajectories, the scene still contains some holes (see Figure 7c). Because the scene is built-up over time, it is impossible to choose camera poses *a-priori*, that view all unobserved regions. To this end, our completion stage samples poses *a-posteriori* to refine those regions. The resulting mesh is watertight and contains no holes (see Figure 7d).

4.4. Spatially Varying Scene Generation

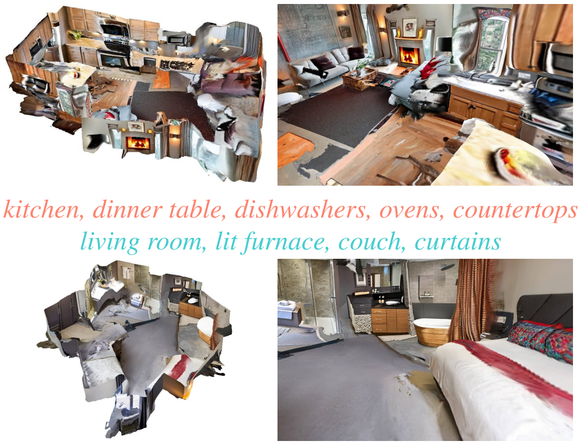
Our method can be applied to generate a scene as the combination of multiple text prompts. Specifically, we use separate text prompts for different poses, crafting a set of trajectories that spatially combines scene descriptions. This



living room, couches, curtains, lit furnace, lamps

(a) Ours w/o alignment (b) Ours w/o stretch removal (c) Ours w/o completion (d) Ours (full)

Figure 7. **Ablation study on the key components of our method.** Without depth alignment (see Section 3.2), different parts of the scene are disconnected and do not fuse into a seamless mesh. Without edge and surface normal thresholds (see Section 3.3), many faces are stretched out unnaturally. Without completion (see Section 3.4), the mesh has holes in remaining unobserved regions. Our full pipeline creates complete, high-resolution scenes. We remove the ceiling in the top-down view for better visualization of the scene layout.



kitchen, dinner table, dishwashers, ovens, countertops
living room, lit furnace, couch, curtains

bathroom, shower, bathtub
bedroom, king-size bed, wardrobes

Figure 8. **Spatially varying scene generation.** Our method can create rooms with multiple parts by prompt mixing. We use separate prompts for cameras viewing different parts of the scene. This is a controllable way to create rooms from multiple descriptions.

can be desired to avoid repeating elements in a complete scene (e.g., multiple couches could be spread out over the whole room when using the same prompt for every camera). Instead, users can specify different object positions through different camera poses and text prompts. It can also be used to design a house comprised of multiple rooms, each with a different type (e.g., a living room that leads to a kitchen).

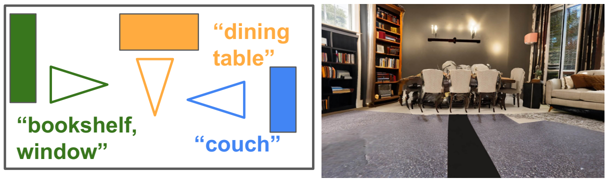


Figure 9. **Scene generation with layout guidance.** Our method can generate scenes from layout guidance. Left: we describe objects with different prompts for cameras facing at different directions. Right: the generated part of the room.

We show results that combine multiple text prompts in Figure 8 and Figure 9.

We note that the layout can only be partially controlled by the camera poses, since scene generation can create chunks with larger or smaller extent. We believe this demonstrates an exciting application of our method, that can be further explored in future work.

4.5. Limitations

Our approach allows to generate 3D room geometry from arbitrary text prompts that are highly detailed and contain consistent geometry. Nevertheless, our method can still fail under certain conditions (see supplemental material). First, our thresholding scheme (see Section 3.3) may not detect all stretched-out regions, which may lead to remaining distortions. Additionally, some holes may still not be completed fully after the second stage (see Section 3.4), which results in over-smoothed regions after applying poisson reconstruction. Our scene representation does not decompose material from lighting, which bakes in shadows or bright lamps, that are generated from the diffusion model.

5. Conclusion

We have shown a method to generate textured 3D meshes from only text input. We use text-to-image 2D generators to create a sequence of images. The core insight of our method is a tailored viewpoint selection, that allows to create a 3D mesh with seamless geometry and compelling textures. Specifically, we lift the images into a 3D scene, by employing our alignment strategy that iteratively fuses all images into the mesh. Our output meshes represent arbitrary indoor scenes that can be rendered with classical rasterization pipelines. We believe our approach demonstrates an exciting application of large-scale 3D asset creation, that only requires text as input.

Acknowledgements

This work was funded in part by Cisco Systems. It was also supported by the ERC Starting Grant Scan2CAD (804724) as well as the German Research Foundation (DFG) Research Unit "Learning and Simulation in Visual Computing." We also thank Angela Dai for the video voice over.

References

[1] Titas Anciukevicius, Zexiang Xu, Matthew Fisher, Paul Henderson, Hakan Bilen, Niloy J. Mitra, and Paul Guerrero. RenderDiffusion: Image diffusion for 3D reconstruction, inpainting and generation. *arXiv*, 2022. 2

[2] Manuel Rey Area, Mingze Yuan, and Christian Richardt. 360monodepth: High-resolution 360° monocular depth estimation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2022. 5, 15

[3] Omri Avrahami, Dani Lischinski, and Ohad Fried. Blended diffusion for text-driven editing of natural images. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18187–18197, 2021. 2

[4] Gwangbin Bae, Ignas Budvytis, and Roberto Cipolla. Irondepth: Iterative refinement of single-view depth using surface normal and its uncertainty. In *British Machine Vision Conference (BMVC)*, 2022. 3, 5

[5] Miguel Angel Bautista, Pengsheng Guo, Samira Abnar, Walter Talbott, Alexander Toshev, Zhuoyuan Chen, Laurent Dinh, Shuangfei Zhai, Hanlin Goh, Daniel Ulbricht, Afshin Dehghan, and Josh Susskind. Gaudi: A neural architect for immersive 3d scene generation. In *NeurIPS*, 2022. 2, 16

[6] Tim Brooks, Aleksander Holynski, and Alexei A. Efros. Instructpix2pix: Learning to follow image editing instructions. In *CVPR*, 2023. 2

[7] Shengqu Cai, Eric Ryan Chan, Songyou Peng, Mohamad Shahbazi, Anton Obukhov, Luc Van Gool, and Gordon Wetzstein. Diffdreamer: Consistent single-view perpetual view generation with conditional diffusion models. In *arXiV*, 2022. 2

[8] Angel X. Chang, Thomas A. Funkhouser, Leonidas J. Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, L. Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *ArXiv*, abs/1512.03012, 2015. 1

[9] Kevin Chen, Christopher Bongsoo Choy, Manolis Savva, Angel X. Chang, Thomas A. Funkhouser, and Silvio Savarese. Text2shape: Generating shapes from natural language by learning joint embeddings. *ArXiv*, abs/1803.08495, 2018. 1, 2

[10] Yongwei Chen, Rui Chen, Jiabao Lei, Yabin Zhang, and Kui Jia. Tango: Text-driven photorealistic and robust 3d stylization via lighting decomposition. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022. 2

[11] Zhaoxi Chen, Guangcong Wang, and Ziwei Liu. Text2light: Zero-shot text-driven hdr panorama generation. *ACM Transactions on Graphics (TOG)*, 41(6):1–16, 2022. 5, 7, 15, 17

[12] Yen-Chi Cheng, Hsin-Ying Lee, Sergey Tulyakov, Alexander Schwing, and Liangyan Gui. Sdfusion: Multimodal 3d shape completion, reconstruction, and generation. *arXiv preprint arXiv:2212.04493*, 2022. 2

[13] 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. 5

[14] Karan Desai, Gaurav Kaul, Zubin Trivadi Aysola, and Justin Johnson. Redcaps: Web-curated image-text data created by the people, for the people. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021. 2

[15] Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems*, 34:8780–8794, 2021. 2

[16] Patrick Esser, Robin Rombach, and Björn Ommer. Taming transformers for high-resolution image synthesis. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12868–12878, 2020. 2

[17] Seth Forsgren* and Hayk Martiros*. Riffusion - Stable diffusion for real-time music generation, https://riffusion.com/about, accessed 2023-03-06, 2022. 2

[18] Rafail Fridman, Amit Abecasis, Yoni Kasten, and Tali Dekel. Scenescape: Text-driven consistent scene generation. *arXiv preprint arXiv:2302.01133*, 2023. 1, 2, 4, 14

[19] Jiatao Gu, Alex Trevithick, Kai-En Lin, Joshua M. Susskind, Christian Theobalt, Lingjie Liu, and Ravi Ramamoorthi. Nerfdiff: Single-image view synthesis with nerf-guided distillation from 3d-aware diffusion. *ArXiv*, abs/2302.10109, 2023. 2

[20] Shuyang Gu, Dong Chen, Jianmin Bao, Fang Wen, Bo Zhang, Dongdong Chen, Lu Yuan, and Baining Guo. Vector quantized diffusion model for text-to-image synthesis.

2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 10686–10696, 2021. 2

[21] Yuan-Chen Guo, Ying-Tian Liu, Ruizhi Shao, Christian Laforte, Vikram Voleti, Guan Luo, Chia-Hao Chen, Zi-Xin Zou, Chen Wang, Yan-Pei Cao, and Song-Hai Zhang. threestudio: A unified framework for 3d content generation. https://github.com/threestudio-project/threestudio, 2023. 15

[22] Takayuki Hara and Tatsuya Harada. Enhancement of novel view synthesis using omnidirectional image completion. *arXiv preprint arXiv:2203.09957*, 2022. 5, 15

[23] Benjamin Hepp, Matthias Nießner, and Otmar Hilliges. Plan3d: Viewpoint and trajectory optimization for aerial multi-view stereo reconstruction. *ACM Transactions on Graphics (TOG)*, 38(1):1–17, 2018. 4

[24] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020. 2

[25] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. In *NeurIPS 2021 Workshop on Deep Generative Models and Downstream Applications*, 2021. 2

[26] Jonathan Ho, Tim Salimans, Alexey A Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. In *Advances in Neural Information Processing Systems*. 2

[27] Ching-Yu Hsu, Cheng Sun, and Hwann-Tzong Chen. Moving in a 360 world: Synthesizing panoramic parallaxes from a single panorama. *arXiv preprint arXiv:2106.10859*, 2021. 5, 15

[28] Huajian Huang, Yingshu Chen, Tianjian Zhang, and Sai-Kit Yeung. 360roam: Real-time indoor roaming using geometry-aware 360° radiance fields. *arXiv preprint arXiv:2208.02705*, 2022. 5, 15

[29] Qingqing Huang, Daniel S. Park, Tao Wang, Timo I. Denk, Andy Ly, Nanxin Chen, Zhengdong Zhang, Zhishuai Zhang, Jia Yu, C. Frank, Jesse Engel, Quoc V. Le, William Chan, and Weixiang Han. Noise2music: Text-conditioned music generation with diffusion models. *ArXiv*, abs/2302.03917, 2023. 2

[30] Ajay Jain, Ben Mildenhall, Jonathan T Barron, Pieter Abbeel, and Ben Poole. Zero-shot text-guided object generation with dream fields. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 857–866. IEEE Computer Society, 2022. 1, 2, 5, 15

[31] Zutao Jiang, Guangsong Lu, Xiaodan Liang, Jihua Zhu, Wei Zhang, Xiaojun Chang, and Hang Xu. 3d-togo: Towards text-guided cross-category 3d object generation. *arXiv preprint arXiv:2212.01103*, 2022. 2

[32] Lei Jin, Yanyu Xu, Jia Zheng, Junfei Zhang, Rui Tang, Shugong Xu, Jingyi Yu, and Shenghua Gao. Geometric structure based and regularized depth estimation from 360 indoor imagery. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 889–898, 2020. 5, 15

[33] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *International Conference on Learning Representations*, 2018. 2

[34] Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. Poisson surface reconstruction. In *Proceedings of the fourth Eurographics symposium on Geometry processing*, volume 7, page 0, 2006. 5, 13, 14

[35] Zhifeng Kong and Wei Ping. On fast sampling of diffusion probabilistic models. In *ICML Workshop on Invertible Neural Networks, Normalizing Flows, and Explicit Likelihood Models*. 2

[36] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. In *International Conference on Learning Representations*. 2

[37] Shreyas Kulkarni, Peng Yin, and Sebastian Scherer. 360fusionnerf: Panoramic neural radiance fields with joint guidance. *arXiv preprint arXiv:2209.14265*, 2022. 5, 15

[38] Blockade Labs. Blockade skybox, https://skybox.blockadelabs.com/, accessed 2023-03-04. 5, 7, 15, 17

[39] Han-Hung Lee and Angel X Chang. Understanding pure clip guidance for voxel grid nerf models. *arXiv preprint arXiv:2209.15172*, 2022. 1, 2, 5, 7, 15

[40] Gang Li, Heliang Zheng, Chaoyue Wang, Chang Li, Changwen Zheng, and Dacheng Tao. 3ddesigner: Towards photorealistic 3d object generation and editing with text-guided diffusion models. *arXiv preprint arXiv:2211.14108*, 2022. 2

[41] Zhengqi Li, Qianqian Wang, Noah Snavely, and Angjoo Kanazawa. Infinitenature-zero: Learning perpetual view generation of natural scenes from single images. In *ECCV*, 2022. 2, 3

[42] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. Magic3d: High-resolution text-to-3d content creation. *arXiv preprint arXiv:2211.10440*, 2022. 1, 2, 5, 15

[43] Andrew Liu, Richard Tucker, Varun Jampani, Ameesh Makadia, Noah Snavely, and Angjoo Kanazawa. Infinite nature: Perpetual view generation of natural scenes from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021. 2, 3

[44] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver++: Fast solver for guided sampling of diffusion probabilistic models. *arXiv preprint arXiv:2211.01095*, 2022. 5

[45] Luke Melas-Kyriazi, C. Rupprecht, Iro Laina, and Andrea Vedaldi. Realfusion: 360° reconstruction of any object from a single image. *ArXiv*, abs/2302.10663, 2023. 2

[46] Gal Metzer, Elad Richardson, Or Patashnik, Raja Giryes, and Daniel Cohen-Or. Latent-nerf for shape-guided generation of 3d shapes and textures. *arXiv preprint arXiv:2211.07600*, 2022. 2

[47] 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. 2

[48] 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 *European Conference on Computer Vision*, 2020. 2, 16

[49] 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. 2

[50] Gimin Nam, Mariem Khlifi, Andrew Rodriguez, Alberto Tono, Linqi Zhou, and Paul Guerrero. 3d-ldm: Neural implicit 3d shape generation with latent diffusion models. *arXiv preprint arXiv:2212.00842*, 2022. 2

[51] Alex Nichol, Prafulla Dhariwal, Aditya Ramesh, Pranav Shyam, Pamela Mishkin, Bob McGrew, Ilya Sutskever, and Mark Chen. Glide: Towards photorealistic image generation and editing with text-guided diffusion models. In *International Conference on Machine Learning*, 2021. 2

[52] Alex Nichol, Heewoo Jun, Prafulla Dhariwal, Pamela Mishkin, and Mark Chen. Point-e: A system for generating 3d point clouds from complex prompts. *arXiv preprint arXiv:2212.08751*, 2022. 2

[53] Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International Conference on Machine Learning*, pages 8162–8171. PMLR, 2021. 2

[54] OpenAI. Dalle: Introducing outpainting, https://openai.com/blog/dall-e-introducing-outpainting?utm_source=tldrnewsletter, accessed 2023-03-07. 5, 7, 15, 17

[55] Or Patashnik, Zongze Wu, Eli Shechtman, Daniel Cohen-Or, and Dani Lischinski. Styleclip: Text-driven manipulation of stylegan imagery. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2065–2074, 2021. 2

[56] Nick St. Pierre. Additive prompting, https://twitter.com/nickfloats/status/1628796348446253057, accessed 2023-03-07. 5

[57] Ben Poole, Ajay Jain, Jonathan T. Barron, and Ben Mildenhall. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv*, 2022. 1, 2, 5, 15

[58] 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. 2, 5

[59] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *ArXiv*, abs/2204.06125, 2022. 1, 2, 5, 7, 15, 17

[60] Nikhila Ravi, Jeremy Reizenstein, David Novotny, Taylor Gordon, Wan-Yen Lo, Justin Johnson, and Georgia Gkioxari. Accelerating 3d deep learning with pytorch3d. *arXiv:2007.08501*, 2020. 5

[61] Ali Razavi, Aaron Van den Oord, and Oriol Vinyals. Generating diverse high-fidelity images with vq-vae-2. *Advances in neural information processing systems*, 32, 2019. 2

[62] Xuanchi Ren and Xiaolong Wang. Look outside the room: Synthesizing a consistent long-term 3d scene video from a single image. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3553–3563, 2022. 2

[63] 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

[64] Mike Roberts, Debadeepta Dey, Anh Truong, Sudipta Sinha, Shital Shah, Ashish Kapoor, Pat Hanrahan, and Neel Joshi. Submodular trajectory optimization for aerial 3d scanning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5324–5333, 2017. 4

[65] C. Rockwell, David F. Fouhey, and Justin Johnson. Pixelsynth: Generating a 3d-consistent experience from a single image. *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14084–14093, 2021. 2

[66] Robin Rombach, A. Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10674–10685, 2021. 1, 2, 5, 13, 15

[67] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2015: 18th International Conference, Munich, Germany, October 5-9, 2015, Proceedings, Part III 18*, pages 234–241. Springer, 2015. 2

[68] Chitwan Saharia, William Chan, Saurabh Saxena, Lala Li, Jay Whang, Emily L. Denton, Seyed Kamyar Seyed Ghasemipour, Burcu Karagol Ayan, Seyedeh Sara Mahdavi, Raphael Gontijo Lopes, Tim Salimans, Jonathan Ho, David J. Fleet, and Mohammad Norouzi. Photorealistic text-to-image diffusion models with deep language understanding. *ArXiv*, abs/2205.11487, 2022. 1, 2

[69] Tim Salimans, Ian Goodfellow, Wojciech Zaremba, Vicki Cheung, Alec Radford, and Xi Chen. Improved techniques for training gans. *Advances in neural information processing systems*, 29, 2016. 5

[70] Aditya Sanghi, Hang Chu, J. Lambourne, Ye Wang, Chin-Yi Cheng, and Marco Fumero. Clip-forge: Towards zero-shot text-to-shape generation. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18582–18592, 2021. 2

[71] Flávio Miguel Schneider, Zhijing Jin, and Bernhard Schölkopf. Moûsai: Text-to-music generation with long-context latent diffusion. *ArXiv*, abs/2301.11757, 2023. 2

[72] Christoph Schuhmann, Romain Beaumont, Richard Vencu, Cade W Gordon, Ross Wightman, Mehdi Cherti, Theo Coombes, Aarush Katta, Clayton Mullis, Mitchell Wortsman, et al. Laion-5b: An open large-scale dataset for training next generation image-text models. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*. 2, 13

[73] Christoph Schuhmann, Richard Vencu, Romain Beaumont, Robert Kaczmarczyk, Clayton Mullis, Aarush Katta, Theo Coombes, Jenia Jitsev, and Aran Komatsuzaki. Laion-400m: Open dataset of clip-filtered 400 million image-text pairs. *ArXiv*, abs/2111.02114, 2021. 2

[74] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *ACL*, 2018. 2

[75] Meng-Li Shih, Shih-Yang Su, Johannes Kopf, and Jia-Bin Huang. 3d photography using context-aware layered depth inpainting. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8025–8035, 2020. 2

[76] Uriel Singer, Adam Polyak, Thomas Hayes, Xiaoyue Yin, Jie An, Songyang Zhang, Qiyuan Hu, Harry Yang, Oron Ashual, Oran Gafni, Devi Parikh, Sonal Gupta, and Yaniv Taigman. Make-a-video: Text-to-video generation without text-video data. *ArXiv*, abs/2209.14792, 2022. 2

[77] Uriel Singer, Shelly Sheynin, Adam Polyak, Oron Ashual, Iurii Makarov, Filippos Kokkinos, Naman Goyal, Andrea Vedaldi, Devi Parikh, Justin Johnson, and Yaniv Taigman. Text-to-4d dynamic scene generation. *ArXiv*, abs/2301.11280, 2023. 2

[78] Josef Sivic, Biliana K. Kaneva, Antonio Torralba, Shai Avidan, and William T. Freeman. Infinite images: Creating and exploring a large photorealistic virtual space. *Proceedings of the IEEE*, 98:1391–1407, 2008. 2

[79] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International Conference on Machine Learning*, pages 2256–2265. PMLR, 2015. 2

[80] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*. 2

[81] Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019. 2

[82] Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. *Advances in neural information processing systems*, 33:12438–12448, 2020. 2

[83] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*. 2

[84] Alexandru Telea. An image inpainting technique based on the fast marching method. *Journal of graphics tools*, 9(1):23–34, 2004. 5, 14, 16

[85] Arash Vahdat, Karsten Kreis, and Jan Kautz. Score-based generative modeling in latent space. In *Neural Information Processing Systems*, 2021. 2

[86] Ruben Villegas, Mohammad Babaeizadeh, Pieter-Jan Kindermans, Hernan Moraldo, Han Zhang, Mohammad Taghi Saffar, Santiago Castro, Julius Kunze, and Dumitru Erhan. Phenaki: Variable length video generation from open domain textual description. *arXiv preprint arXiv:2210.02399*, 2022. 2

[87] Can Wang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Clip-nerf: Text-and-image driven manipulation of neural radiance fields. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3825–3834, 2021. 2

[88] Can Wang, Ruixiang Jiang, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. Nerf-art: Text-driven neural radiance fields stylization. *arXiv preprint arXiv:2212.08070*, 2022. 2

[89] Haochen Wang, Xiaodan Du, Jiahao Li, Raymond A Yeh, and Greg Shakhnarovich. Score jacobian chaining: Lifting pretrained 2d diffusion models for 3d generation. *arXiv preprint arXiv:2212.00774*, 2022. 1, 2, 5, 15

[90] Daniel Watson, William Chan, Ricardo Martin-Brualla, Jonathan Ho, Andrea Tagliasacchi, and Mohammad Norouzi. Novel view synthesis with diffusion models. *arXiv preprint arXiv:2210.04628*, 2022. 2

[91] Olivia Wiles, Georgia Gkioxari, Richard Szeliski, and Justin Johnson. Synsin: End-to-end view synthesis from a single image. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7465–7475, 2019. 2

[92] Jay Zhangjie Wu, Yixiao Ge, Xintao Wang, Weixian Lei, Yuchao Gu, Wynne Hsu, Ying Shan, Xiaohu Qie, and Mike Zheng Shou. Tune-a-video: One-shot tuning of image diffusion models for text-to-video generation. *ArXiv*, abs/2212.11565, 2022. 2

[93] Dejia Xu, Yifan Jiang, Peihao Wang, Zhiwen Fan, Yi Wang, and Zhangyang Wang. Neurallift-360: Lifting an in-the-wild 2d photo to a 3d object with 360° views. *arXiv e-prints*, pages arXiv–2211, 2022. 2

[94] Jiale Xu, Jia Zheng, Yanyu Xu, Rui Tang, and Shenghua Gao. Layout-guided novel view synthesis from a single indoor panorama. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16438–16447, 2021. 5, 15

[95] Lvmin Zhang and Maneesh Agrawala. Adding conditional control to text-to-image diffusion models. *arXiv preprint arXiv:2302.05543*, 2023. 2

A. Supplemental Video

Please watch our attached video * for a comprehensive evaluation of the proposed method. We include rendered videos of multiple generated scenes from novel trajectories, that showcase the quality of both generated texture and geometry (and also show the generated ceilings). We also show an animation how the mesh is built up over time, that illustrates the usage of our two-stage pose sampling scheme (generation and completion). We compare against baselines and ablations of our method by showing rendered videos.

B. Societal Impact

Our method leverages text-to-image models to generate a sequence of images from text, specifically we use the Stable Diffusion model [66]. Thus it inherits possible drawbacks of these 2D models. First, our method could be exploited to generate harmful content, by forcing the text-to-image model to generate respective images. Furthermore, our method is biased towards the cultural or stereotypical data distribution, that was used to train the text-to-image model. Lastly, we note that text-to-image models are trained on large-scale text-image datasets [72]. Thus, the model learns to reproduce and combine the style of artists, whose works are contained in these datasets. This raises questions regarding the correct way to credit these artists or if it is ethical to benefit from their works in this way at all.

Our method can be used to generate meshes, that depict entire scenes, from only text as input. This significantly reduces the required expertise to model and design such 3D assets. Thus, we believe our work proposes a promising step towards the democratization of large-scale 3D content creation.

C. Limitations

Given a text prompt, our approach allows to generate 3D room geometry that is highly detailed and contains consistent 3D geometry. Nevertheless, our method can still fail under certain conditions (see Figure 10).

First, our completion stage (see Section 3.4) might not be able to inpaint all holes (Figure 10b). For example this can happen, if an object contains holes that are close to a wall. These angles are hard to see from additional cameras and thus might remain untouched. We still close these holes by applying Poisson surface reconstruction [34]. However, this can results in overly smoothed geometry.

Second, our mesh fusion stage (see Section 3.3) might not remove all stretched-out faces. Faces can appear stretched-out because of imperfect depth estimation and alignment. Over time this can yield unusual room shapes such as the curved wall in Figure 10c. We apply two filtering schemes to remove stretched-out faces before fusing them with the existing geometry. Both use thresholds $\delta_{sn}{=}0.1, \delta_{edge}{=}0.1$, that we fix during all our experiments. It can happen that some faces are not removed by the filtering schemes, but are still stretched-out unnaturally. However, we find that lowering the thresholds would also remove unstretched geometry. This would make creating a complete scene harder, because more holes need to be inpainted in the completion stage.

D. Details on User Study

We conduct a user study and ask $n{=}61$ users to score Perceptual Quality (*PQ*) and 3D Structure Completeness (*3DS*) of the whole scene on a scale of $1{-}5$. We show an example of how we asked the users to score these two metrics in Figure 11. We present users with multiple images from each scene, that show it from multiple angles. Then we ask them to rate the scene on a scale from $1{-}5$ by asking them about the 3D structure completeness and the overall perceptual quality. In total, we received 1098 datapoints from multiple scenes and report averaged results per method.

E. Additional Implementation Details

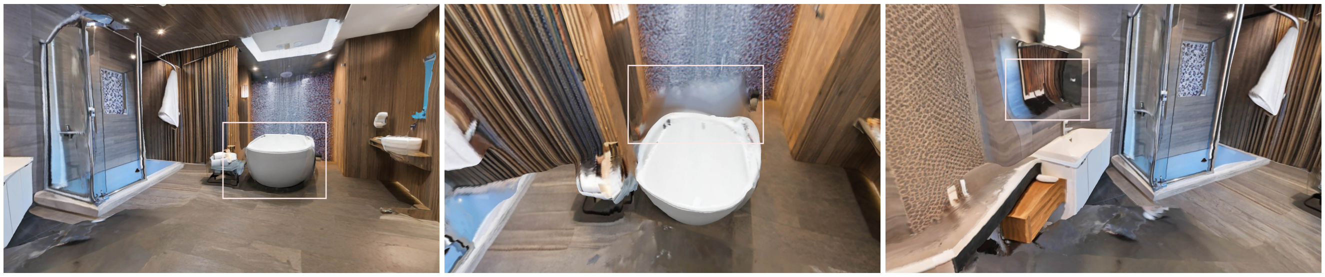
We give additional implementation details in the following subsections.

E.1. Importance of Predefined Trajectories

We create the complete scene layout and furniture in the first stage of our tailored two-stage viewpoint selection scheme (see Section 3.4). To this end, we sample multiple *predefined* trajectories from which we iteratively generate the scene. We fix the trajectories for our main results, as we found it already creates rooms with a variety of different layouts. Users can modify them according to our guidelines as demonstrated in Section 4.4 in the main paper. Each trajectory consists of a start pose and an end pose and we linearly interpolate between both. We found generation works best, if each trajectory starts off from a viewpoint with mostly unobserved regions. This gives the text-to-image model enough freedom to create novel content with reasonable global structure.

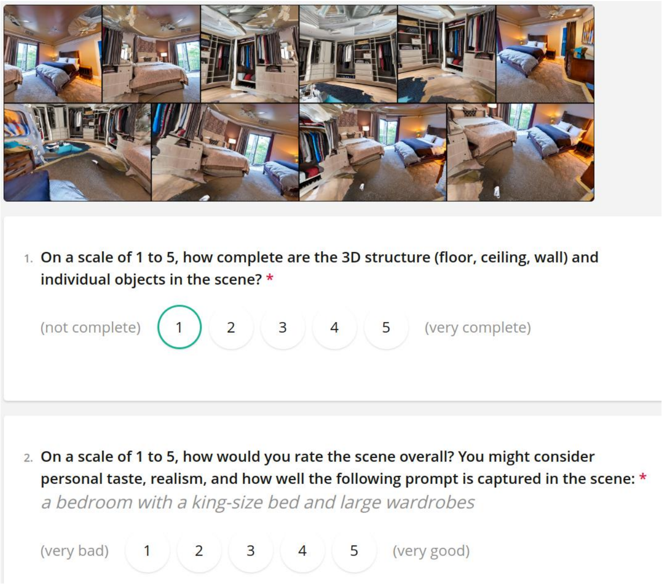
Thus, we construct each trajectory with the following principle. First, we select a start pose that views mostly unobserved content and generate the outline of the next scene chunk from it (Figure 12b). Then, we subsequently translate and rotate into the chunk to refine its 3D structure until the end of the trajectory (Figure 12c). This creates mesh patches with convincing 3D structure (Figure 12d). In contrast, if we design trajectories that do not follow this principle, results can degenerate. For example, if the viewpoint change is small, the text-to-image model creates novel con-

*https://youtu.be/fjRnFL91EZc



(a) created scene (b) overly smoothed geometry (c) stretched geometry

Figure 10. **Limitations of our method.** (a) Our approach creates scenes with compelling textures and complete structure like walls, floor and ceiling. (b) Our completion stage (see Section 3.4) might not be able to inpaint all holes, if no suitable camera pose could be sampled (e.g. small areas behind an object that are close to a wall). The hole is still closed through Poisson reconstruction [34], but the geometry may become smoothed. (c) Our fusion stage (see Section 3.3) might not remove all stretched-out faces, because we use fixed thresholds.



1. On a scale of 1 to 5, how complete are the 3D structure (floor, ceiling, wall) and individual objects in the scene? *

(not complete) [✓ 1] 2 3 4 5 (very complete)

2. On a scale of 1 to 5, how would you rate the scene overall? You might consider personal taste, realism, and how well the following prompt is captured in the scene: *
a bedroom with a king-size bed and large wardrobes

(very bad) 1 2 3 4 5 (very good)

Figure 11. **User study interface.** (Top) We present users with multiple images from each scene, that show it from multiple angles. (Bottom) We ask users to rate the scene on a scale from $1-5$ by asking them about the 3D structure completeness (question 1) and the overall perceptual quality (question 2).

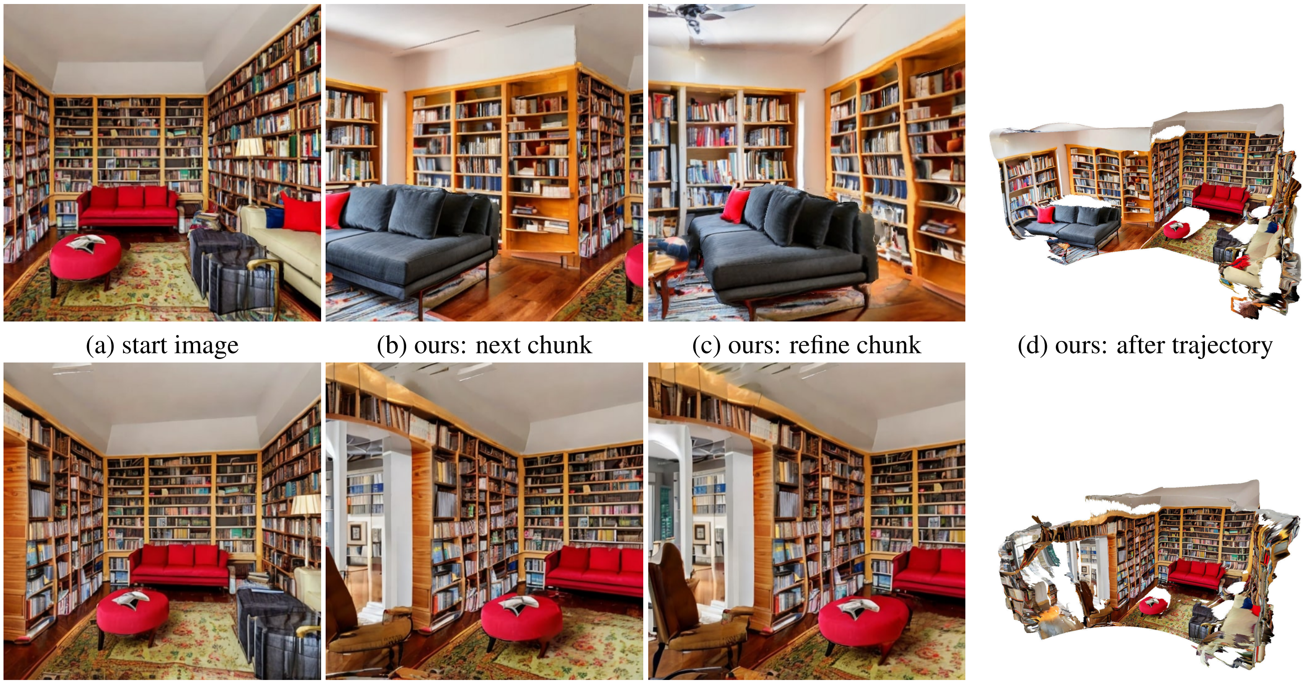
tent only for small portions of the image (Figure 12e-g). Thus, locally the generated content looks reasonable, but it accumulates into inconsistent global structure (Figure 12h).

E.2. Effect of Depth Smoothing in Alignment

For each camera pose in both stages, we follow an iterative scene generation scheme (see Section 3.1). After generating novel content, we predict its depth in our depth alignment stage (see Section 3.2). First, we predict the depth using a monocular depth inpainting network (Figure 13b). However, directly using this depth for mesh fusion results in unaligned mesh patches (Figure 13g). Thus, we improve the result by aligning rendered depth and inpainted depth in the least squares sense (Figure 13c). Finally, we smooth the aligned depth by applying a 5×5 gaussian blur kernel at the image edges between rendered and predicted depth (Figure 13d). This smoothens out remaining discontinuity artifacts between old and new content (Figure 13e and f). In practice, we found this can further reduce sharp borders between objects, leading to overall better alignment (Figure 13h).

E.3. Importance of Mask Dilation in Completion

We complete the scene in the second stage of our tailored two-stage viewpoint selection scheme, by filling in remaining holes in the mesh (see Section 3.4). To this end, we first select suitable camera poses that look at these holes (Figure 14a). We then follow the iterative scene generation scheme to fill in the holes in the mesh (see Section 3.1). The holes can have arbitrarily small or large sizes, depending on how the scene layout was generated in the first stage of our method (Figure 14b). Similarly to Fridman *et al.* [18], we found that directly inpainting such holes can lead to suboptimal results (Figure 14c). This is because the text-to-image model needs to inpaint small regions and the direct neighborhood of the holes can be distorted. To alleviate this issue, we inpaint small holes with a classical inpainting algorithm [84]. We classify small holes by applying a morphological erosion operation with a 3×3 kernel on the inpainting mask. Next, we increase the size of remaining holes, by repeating a morphological dilation operation with a 7×7 kernel on the eroded inpainting mask for five times (Figure 14d). Finally, we inpaint the image using the dilated mask (Figure 14e). This yields more convincing results because the text-to-image model can inpaint larger areas and create more meaningful global structure. To combine the new content with the existing mesh, we apply our triangulation scheme (see Section 3.3). Additionally, we remove all faces that fall into the dilated region and are close to the rendered screen-space depth (since they are replaced by the novel content).



(a) start image (b) ours: next chunk (c) ours: refine chunk (d) ours: after trajectory

(e) small change I (f) small change II (g) small change III (h) small change: after trajectory

Figure 12. **Importance of predefined trajectories.** We sample predefined trajectories in the first stage of our tailored two-stage viewpoint selection scheme (see Section 3.4). First, we create the outline of the next scene chunk (b). Then, we sample additional poses that translate and rotate into the new scene chunk to complete its 3D structure (c). This results in a 3D consistent next mesh patch, that we fuse with existing content (d). In contrast, results degenerate (h), if we sample sub-optimal poses (e.g. small viewpoint changes in e-g).

F. Additional Discussion on Related Methods and Baselines

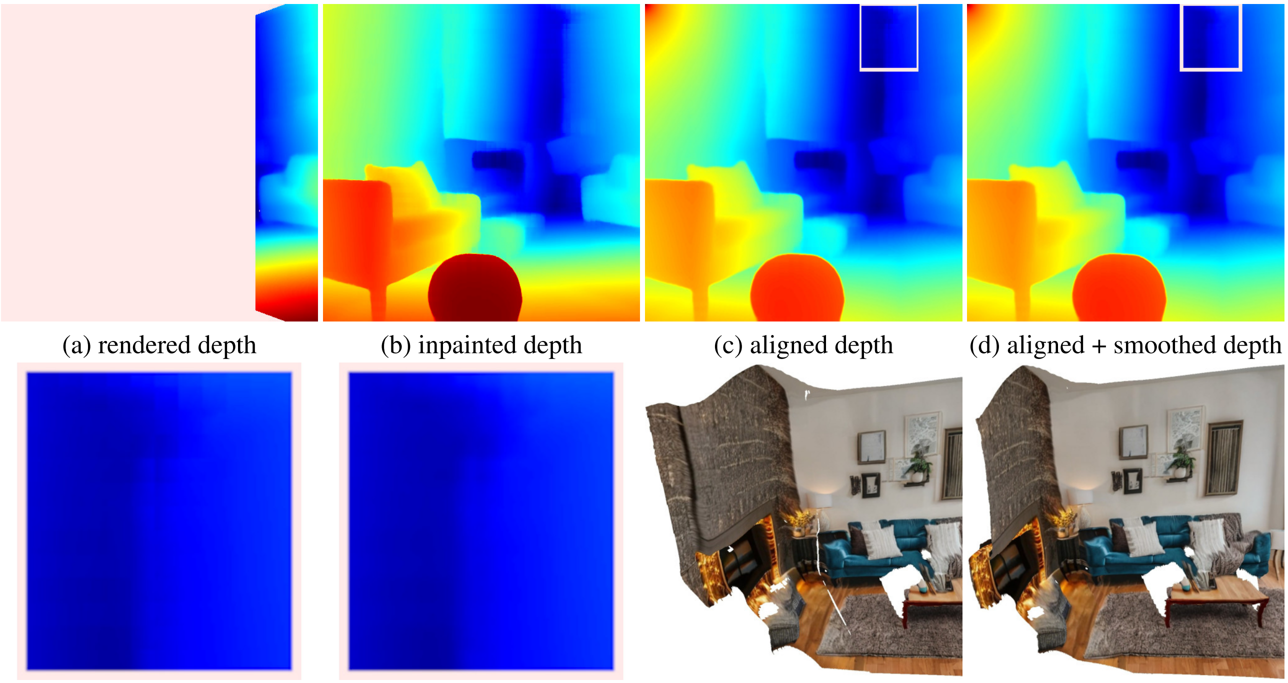
To the best of our knowledge, there are no direct baselines that generate textured 3D room geometry from text. We compare against four related methods, that do not require supervision from 3D datasets. In the following we give additional discussion on related methods and our selected baselines.

PureClipNeRF [39]: We compare against text-to-3D methods for generating objects [57, 42, 30, 39, 89] and choose Lee *et al*. [39] as open-source representative. A common pattern in these text-to-3D methods is to sample inward-facing poses on a hemisphere, from which the object is iteratively optimized. While the method of Lee *et al*. [39] does not use a diffusion model to create high-fidelity images, it still uses the same pose sampling pattern. This allows us to compare against these methods in general, by analyzing how well this pose sampling pattern can produce complete 3D scenes with structural elements like walls or floors. We also run DreamFusion [57] from the third-party implementation of Guo *et al*. [21], see Figure 15. Similar to PureClipNeRF, object-centric cameras yield incomplete rooms. Outward-facing cameras yield blurry 360° surroundings, showing floaters when rendered out-of-distribution.

Outpainting [59, 54]: We compare against image outpainting. We combine outpainting from a Stable Diffusion [66] model with depth estimation and triangulation to create a mesh from an enlarged viewpoint. Starting off from a single generated image, we can synthesize novel content around it to create a complete scene in a single image plane (Figure 16a). After creating the image, we then perform depth estimation and triangulation to lift the image into a 3D mesh.

Text2Light [11]: We generate RGB panoramas from text using Chen *et al*. [11]. We show example outputs in Figure 16b. One can create immersive experiences by rendering a panorama onto a sphere, allowing to view the scene from arbitrary 360° viewpoints. However, it is not possible to simulate a true 3D environment directly (e.g., translating or rotating around objects), because the panorama only captures a single viewpoint. Thus, related approaches estimate room layout [94], perform view synthesis [37, 27, 22, 28] or predict 360° depth [2, 32] from one or multiple panoramas. To compare to our method, we reconstruct the 3D mesh structure that can be obtained from a single panoramic image. To this end, we perform depth prediction and subsequently apply our mesh fusion step.

Blockade [38]: We compare against *Blockade* [38], which uses a text-to-image diffusion model to produce expressive


(a) rendered depth (b) inpainted depth (c) aligned depth (d) aligned + smoothed depth

(e) zoom-in of (c) (f) zoom-in of (d) (g) fused mesh from (b) (h) fused mesh from (d)

Figure 13. **Details on the depth alignment step.** For each novel pose, we predict the depth for the newly generated image content (see Section 3.2). First we inpaint the depth using a monocular depth prediction network (b). Then, we align inpainted depth (b) and rendered depth (a) in the least squares sense to obtain an aligned depth (c). Finally, we smooth the result to remove remaining sharp borders between old and new content (d). This results in smoother, less blocky depth (e and f). Our depth alignment is necessary to create transitions without holes between mesh patches (g and h).

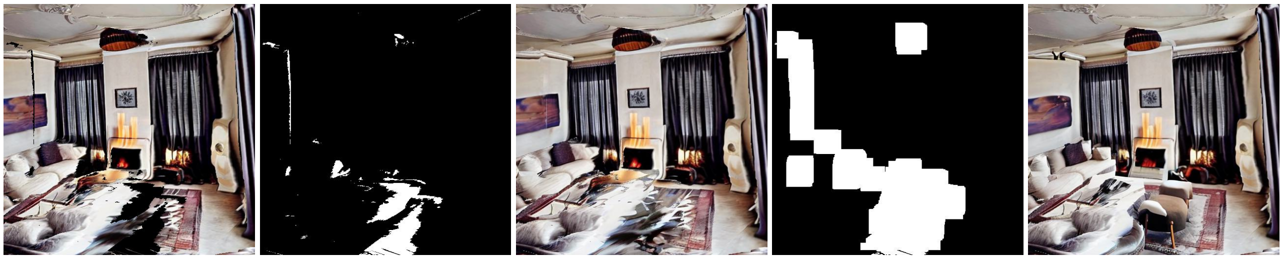

(a) rendered image (b) rendered mask (c) inpaint naïve (d) dilated mask (e) inpaint dilated

Figure 14. **Importance of mask dilation during completion.** In our second stage, we complete the scene mesh by filling in unobserved regions (see Section 3.4). First, we sample camera poses that view such unobserved regions (a). The unobserved regions can have arbitrary size (b). Directly inpainting only the masked regions from (b) gives distorted results, because the holes can be too small for reasonable inpainting results (c). Instead, we inpaint small holes with a classical inpainting method [84] and dilate remaining holes to a larger size (d). The resulting image after inpainting contains more reasonable structure (e).

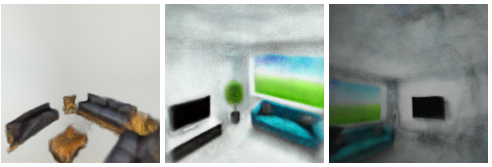
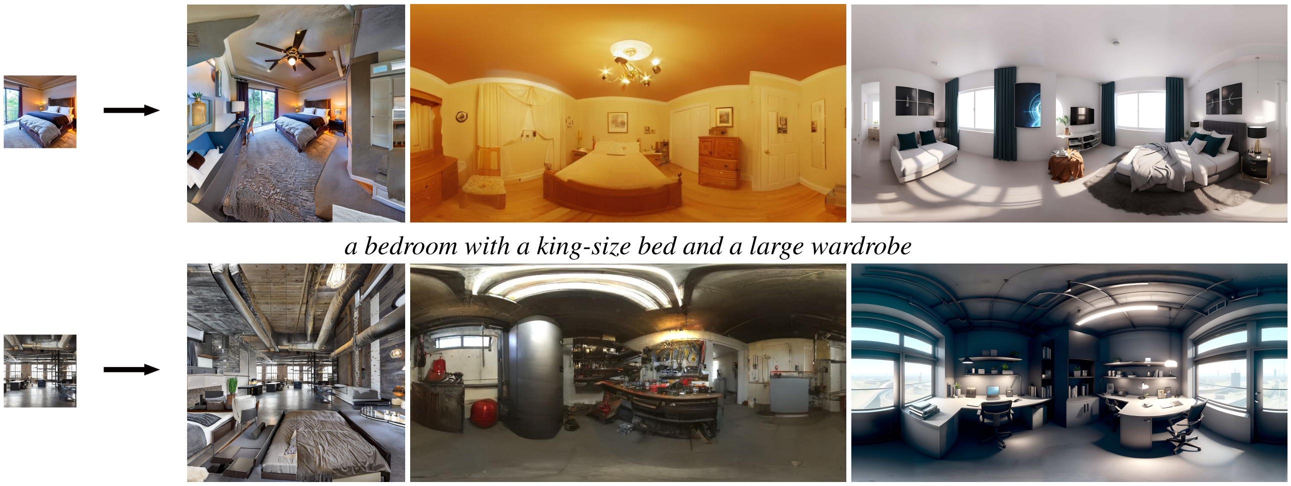


Figure 15. Left: DreamFusion-Inward. Mid/Right: DreamFusion-Outward from in- and out-of-distribution viewpoints.

RGB panoramas. We then extract the mesh similarly.

GAUDI [5]: Bautista and Guo *et al.* [5] present a method to generate large-scale 3D scenes encoded into a NeRF [48] representation. Their generative model can be conditioned to produce 3D indoor scenes from text as input. In general, each scene allows for a different distribution of camera poses. Walls and objects are placed at different positions in each scene, thus it depends on the scene to determine valid camera poses. They model this joint latent distribution of scenes and cameras. This allows to synthesize scenes that can be rendered from corresponding camera trajectories (e.g., a scene is rendered in a forward motion). How-



a bedroom with a king-size bed and a large wardrobe

Editorial Style Photo, Industrial Home Office, Steel Shelves, Concrete, Metal, Edison Bulbs, Exposed Ductwork

(a) Outpainting [59, 54] (b) Text2Light [11] (c) Blockade [38]

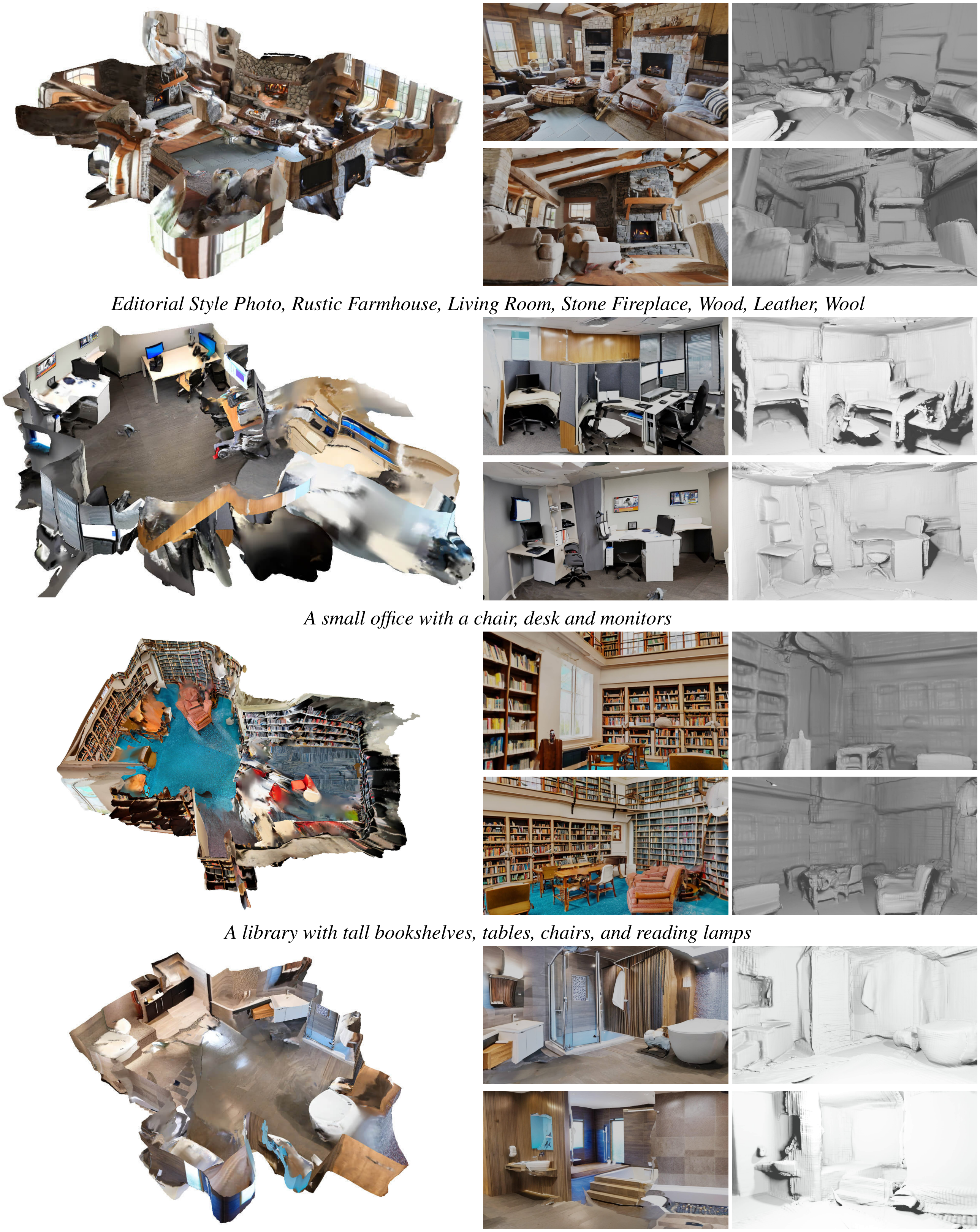
Figure 16. **Intermediate results from baselines.** We first produce these intermediate results, before unprojecting them into a 3D mesh. (a) Outpainting [59, 54] generates an enlarged scene from a single viewpoint. (b) Text2Light [11] creates a panoramic image of a scene. (c) Blockade [38] creates a panoramic image of a scene.

ever, it requires training supervision from 3D datasets that contain ground-truth camera trajectories. This restricts the method to the domain of a specific dataset of (synthetic, low-resolution) 3D scenes, which is limited in size and diversity.

In contrast, we choose another approach to represent the joint distribution of scenes and camera trajectories. Our two-stage tailored viewpoint selection (see Section 3.4) first creates the general scene layout and furniture from predefined trajectories. We choose these trajectories such that the camera poses do not intersect with generated geometry (see Section 3.4 for more details). Then we inpaint remaining holes by sampling additional poses. This allows us to generate complete scenes with varying layouts. Our resulting mesh can be rendered from arbitrary viewpoints, i.e., it is not bound to the specific trajectory used during generation. Furthermore, our method can directly lift the generated images of a 2D text-to-image model into 3D, without requiring supervised training from 3D datasets. This allows us to generate meshes, that can represent a much larger and more diverse set of indoor scenes with higher visual quality.

G. Additional Qualitative Results

We show additional qualitative results of our method in Figure 17.



Editorial Style Photo, Rustic Farmhouse, Living Room, Stone Fireplace, Wood, Leather, Wool

A small office with a chair, desk and monitors

A library with tall bookshelves, tables, chairs, and reading lamps

A large bathroom with shower, bathtub and a cozy wellness area

Figure 17. **3D scene generation results of our method.** We show color and shaded geometry renderings from generated scenes with corresponding text prompts. Our method synthesizes realistic meshes satisfying text descriptions. We remove the ceiling in the top-down view for better visualization of the scene layout.