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

A. Supplemental Video

Please watch our attached video [1] 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 [20]. 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 [21]. 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 1).

First, our completion stage (see Section 3.4) might not be able to inpaint all holes (Figure 1b). 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 [11]. 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 1c. We apply two filtering schemes to remove stretched-out faces before fusing them with the existing geometry. Both use thresholds δ_{sn}=0.1, δ_{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 2. 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

[1] https://youtu.be/fjRnFL91EZc

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

Figure 1. **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 [11], 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.

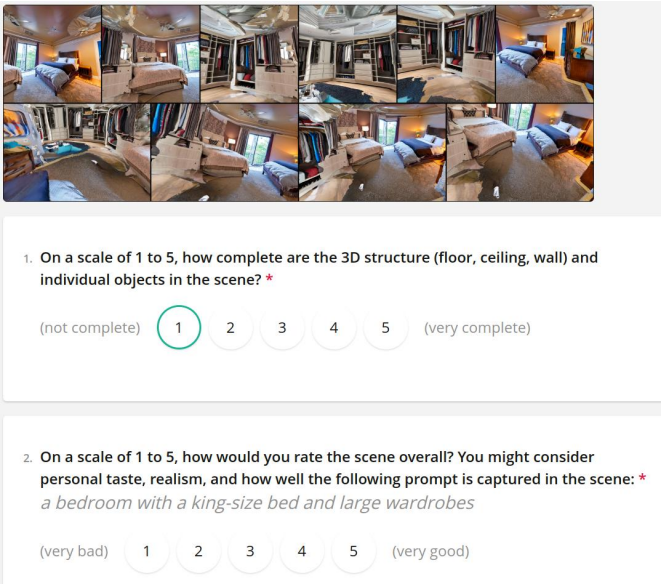


Figure 2. **User study interface.** (Top) We present users with multiple images from each scene, that show it from multiple angles. (Bottom) We ask users to rate the scene on a scale from $1-5$ by asking them about the 3D structure completeness (question 1) and the overall perceptual quality (question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

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 3b). Then, we subsequently translate and rotate into the chunk to refine its 3D structure until the end of the trajectory (Figure 3c). This creates mesh patches with convincing 3D structure (Figure 3d). 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 content only for small portions of the image (Figure 3e-g). Thus, locally the generated content looks reasonable, but it accumulates into inconsistent global structure (Figure 3h).

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 4b). However, directly using this depth for mesh fusion results in unaligned mesh patches (Figure 4g). Thus, we improve the result by aligning rendered depth and inpainted depth in the least squares sense (Figure 4c). Finally, we smooth the aligned depth by applying a 5×5 gaussian blur kernel at the image edges between rendered and predicted depth (Figure 4d). This smoothens out remaining discontinuity artifacts between old and new content (Figure 4e and f). In practice, we found this can further reduce sharp borders between objects, leading to overall better alignment (Figure 4h).

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 5a). 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 5b). Similarly to Fridman *et al.* [4], we found that directly inpainting such holes can lead to sub-optimal results (Figure 5c). This is because the text-to-image model needs to inpaint small regions and the direct

(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 3. **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).

neighborhood of the holes can be distorted. To alleviate this issue, we inpaint small holes with a classical inpainting algorithm [22]. 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 5d). Finally, we inpaint the image using the dilated mask (Figure 5e). 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).

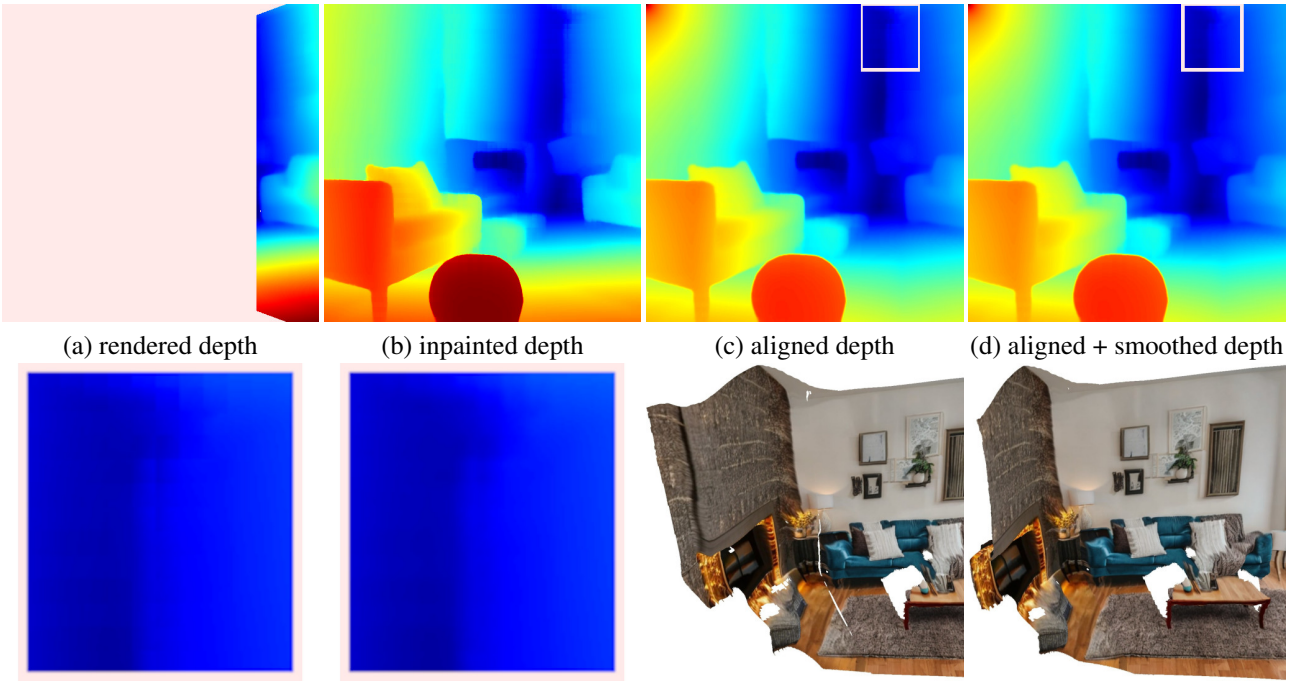
F. Additional Discussion on Related Methods and Baselines

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 [14]: We compare against text-to-3D methods for generating objects [18, 15, 9, 14, 23] and choose Lee *et al*. [14] 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*. [14] 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 [18] from the third-party implementation of Guo *et al*. [5], see Figure 6. Similar to PureClipNeRF, object-centric cameras yield incomplete rooms. Outward-facing cameras yield blurry 360° surroundings, showing floaters when rendered out-of-distribution.

Outpainting [19, 17]: We compare against image outpainting. We combine outpainting from a Stable Diffusion [20] 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 7a). After creating the image, we then perform depth estimation and triangulation to lift the image into a 3D mesh.

Text2Light [3]: We generate RGB panoramas from text



(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 4. **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 5. **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 [22] and dilate remaining holes to a larger size (d). The resulting image after inpainting contains more reasonable structure (e).

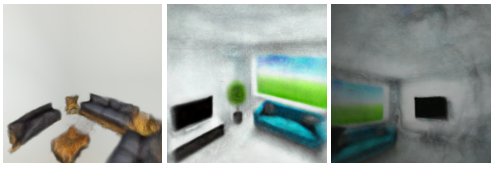


Figure 6. Left: DreamFusion-Inward. Mid/Right: DreamFusion-Outward from in- and out-of-distribution viewpoints.

using Chen *et al*. [3]. We show example outputs in Figure 7b. 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 [24], perform view synthesis [12, 7, 6, 8] or predict 360° depth [1, 10] 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 subse-

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 [19, 17] (b) Text2Light [3] (c) Blockade [13]

Figure 7. **Intermediate results from baselines.** We first produce these intermediate results, before unprojecting them into a 3D mesh. (a) Outpainting [19, 17] generates an enlarged scene from a single viewpoint. (b) Text2Light [3] creates a panoramic image of a scene. (c) Blockade [13] creates a panoramic image of a scene.

quently apply our mesh fusion step.

Blockade [13]: We compare against *Blockade* [13], which uses a text-to-image diffusion model to produce expressive RGB panoramas. We then extract the mesh similarly.

GAUDI [2]: Bautista and Guo *et al*. [2] present a method to generate large-scale 3D scenes encoded into a NeRF [16] 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). However, 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 8.

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 8. **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.

References

[1] 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. 4

[2] 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. 5

[3] 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. 3, 4, 5

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

[5] 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. 3

[6] Takayuki Hara and Tatsuya Harada. Enhancement of novel view synthesis using omnidirectional image completion. *arXiv preprint arXiv:2203.09957*, 2022. 4

[7] 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. 4

[8] 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. 4

[9] 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. 3

[10] 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. 4

[11] 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. 1, 2

[12] Shreyas Kulkarni, Peng Yin, and Sebastian Scherer. 360fusionnerf: Panoramic neural radiance fields with joint guidance. *arXiv preprint arXiv:2209.14265*, 2022. 4

[13] Blockade Labs. Blockade skybox, https://skybox.blockadelabs.com/, accessed 2023-03-04. 5

[14] Han-Hung Lee and Angel X Chang. Understanding pure clip guidance for voxel grid nerf models. *arXiv preprint arXiv:2209.15172*, 2022. 3

[15] 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. 3

[16] 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. 5

[17] OpenAI. Dalle: Introducing outpainting, https://openai.com/blog/dall-e-introducing-outpainting?utm_source=tldrnewsletter, accessed 2023-03-07. 3, 5

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

[19] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen. Hierarchical text-conditional image generation with clip latents. *ArXiv*, abs/2204.06125, 2022. 3, 5

[20] 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, 3

[21] 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*. 1

[22] Alexandru Telea. An image inpainting technique based on the fast marching method. *Journal of graphics tools*, 9(1):23–34, 2004. 3, 4

[23] 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. 3

[24] 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. 4