SceneWeaver: Text-Driven Scene Generation with Geometry-aware Gaussian Splatting

In this **Supplementary Material**, we will further detail the following aspects omitted in the **Main Paper**.

- **Section 1:** Detailed description of progressive scene generation framework.
- **Section 2:** Model zoo.
- **Section 3:** Additional qualitative results.

1. Detailed description of progressive scene generation framework

In this section, we describe in more detail the progressive scene generation framework used by our method. Our goal is to generate 3D scenes that match the given text prompts.

Initial View Generation. For the initial view, if the user chooses to generate a scene based on a natural language description y, which could be a sentence describing objects or abstract style and quality requirements. We employ a pre-trained text-to-image diffusion model Rombach et al. (2022) F_{t2i} to generate the initial view image $\boldsymbol{I}_0 \in \mathbb{R}^{3\times H\times W}$, where H and W denote the height and width of the image. Alternatively, users can generate a scene using a synthetic or real image. In this scenario, a pre-trained image-to-text generation model F_{i2t} Contributors (2023) is used to create a description as text prompt y for text-conditioned inpainting. We utilize a monocular depth estimator F_d Bhat et al. (2023) to obtain the depth map $\boldsymbol{D}_0^m \in \mathbb{R}^{H\times W}$ from $\boldsymbol{I}_0$. The object boundaries and details in the depth map are enhanced by the Multi-level Depth Refinement (MDR) mechanism to improve the quality of the depth map and obtain a refined depth map $\tilde{\boldsymbol{D}}_0^m$.

Initial Point Cloud Generation. We provide predefined cameras $\{\boldsymbol{C}_i\}_{i=0}^N$, where N denotes the number of cameras. Each camera $\boldsymbol{C}_i$ consists of the extrinsic parameters $\boldsymbol{E}_i \in \mathbb{R}^{3\times 4}$ and a shared intrinsic parameter $\boldsymbol{K} \in \mathbb{R}^{3\times 3}$. Based on the camera parameters $\boldsymbol{E}_0$ and $\boldsymbol{K}$, the RGB image $\boldsymbol{I}_0$, and the corresponding refined depth map $\tilde{\boldsymbol{D}}_0^m$, we lift 2D pixels into 3D space. The point cloud for the initial view is defined as $\boldsymbol{P}_0$:

$$\boldsymbol{P}_0 = f_{2\to 3}(\boldsymbol{I}_0, \tilde{\boldsymbol{D}}_0^m, \boldsymbol{E}_0, \boldsymbol{K}), \tag{1}$$

where $f_{2\to 3}$ is a series of geometric transformations made to lift 2D pixels into 3D space.

Progressively Point Cloud Generation. The generated point cloud for each camera pose should be fused to the existing one. Specifically, at the i^{th} $(i \neq 0)$ camera, the existing point cloud $\boldsymbol{P}_{i-1}$ is projected into 2D-pixel space. Due to the change in camera position, this projection results in a partial image $\hat{\boldsymbol{I}}_i$ and a mask $\hat{\boldsymbol{M}}_i$ indicating the area to be inpainted:

$$\hat{\boldsymbol{I}}_i, \hat{\boldsymbol{M}}_i = f_{3\to 2}(\boldsymbol{P}_{i-1}, \boldsymbol{E}_i, \boldsymbol{K}), \tag{2}$$

where $f_{3\to2}$ is a series of geometric transformations made to project the point cloud from the world coordinate system to the pixel coordinate system.

The image inpainting model $F_{inpaint}$ is used to obtain the full image $\boldsymbol{I}_i$ based on the partial image $\hat{\boldsymbol{I}}_i$, the mask $\hat{\boldsymbol{M}}_i$ and the text prompt y. The monocular depth estimator F_d is utilized to get the corresponding depth $\boldsymbol{D}_i^m$, which is further refined to obtain refined depth $\tilde{\boldsymbol{D}}_i^m$ using MDR. In the progressive generation of the point cloud, inpainted pixels from $\boldsymbol{I}_i$ are lifted to the point cloud at each camera pose. Since there is some difference between the neighboring depth maps, $\tilde{\boldsymbol{D}}_i^m$ needs to be processed by minimizing the distance between the overlapping regions of the two point clouds to get the aligned depth $\bar{\boldsymbol{D}}_i$. Following Chung et al. (2023), we estimate the optimal depth scale factor d_i to minimize the distance. The above process can be defined as follows:

$$\bar{\boldsymbol{D}}_i = d_i \tilde{\boldsymbol{D}}_i^m \tag{3}$$

Then the inpainted 2D pixels need to be transformed from pixel coordinates to world coordinates to get the updated point cloud $\boldsymbol{P}_i$:

$$\boldsymbol{P}_i = f_{update}(f_{2\to3}(\boldsymbol{I}_i, \bar{\boldsymbol{D}}_i, \boldsymbol{E}_i, \boldsymbol{K}), \boldsymbol{P}_{i-1}, \hat{\boldsymbol{M}}_i = 0), \tag{4}$$

where $f_{update}(\cdot)$ is the function that fuses the new point cloud into the existing point cloud $\boldsymbol{P}_{i-1}$. $\hat{\boldsymbol{M}}_i = 0$ means only the inpainted pixels needs to be transformed. Repeat the above steps N times to get the final point cloud $\boldsymbol{P}_N$. 3D Gaussians are initialized by $\boldsymbol{P}_N$ and optimized by following the optimization objective $\mathcal{L}_{rgb}$ proposed in Kerbl et al. (2023).

2. Model zoo

SceneScape Fridman et al. (2024) is a text-driven perpetual view generation method that synthesizes long-term videos of various scenes from a text prompt describing the scene and camera poses. It leverages a pre-trained text-to-image model and a pre-trained monocular depth prediction model to construct a unified mesh representation of the scene.

Text2Room Höllein et al. (2023) generates room-scale textured 3D meshes from text prompts. It utilizes a pre-trained text-to-image model to synthesize a series of images from different camera poses. Using a progressive alignment strategy, it seamlessly merges the content of each image into a textured 3D mesh, resulting in complete 3D scenes with multiple objects and clear geometric structures.

WonderJourney Yu et al. (2024) is a modular framework for perpetual 3D scene generation. The model starts with any textual description or image and utilizes LLM to generate textual descriptions of the scene at each stage. Scenes are generated using a text-driven point cloud generation pipeline and the generated scenes are validated using a large VLM. For a fair comparison with other methods, WonderJourney uses the same textual description at different stages of a scene and does not use VLM validation.

LucidDreamer Chung et al. (2023) is a domain-free scene generation pipeline that leverages existing large-scale diffusion generation models and monocular depth estimators. The model incrementally lifts 2D pixels into 3D space, merging them with the existing point cloud. The point cloud of the scene is used as the initial points for 3D Gaussians, and the scene is optimized by the optimization objective in Gaussian Splatting.

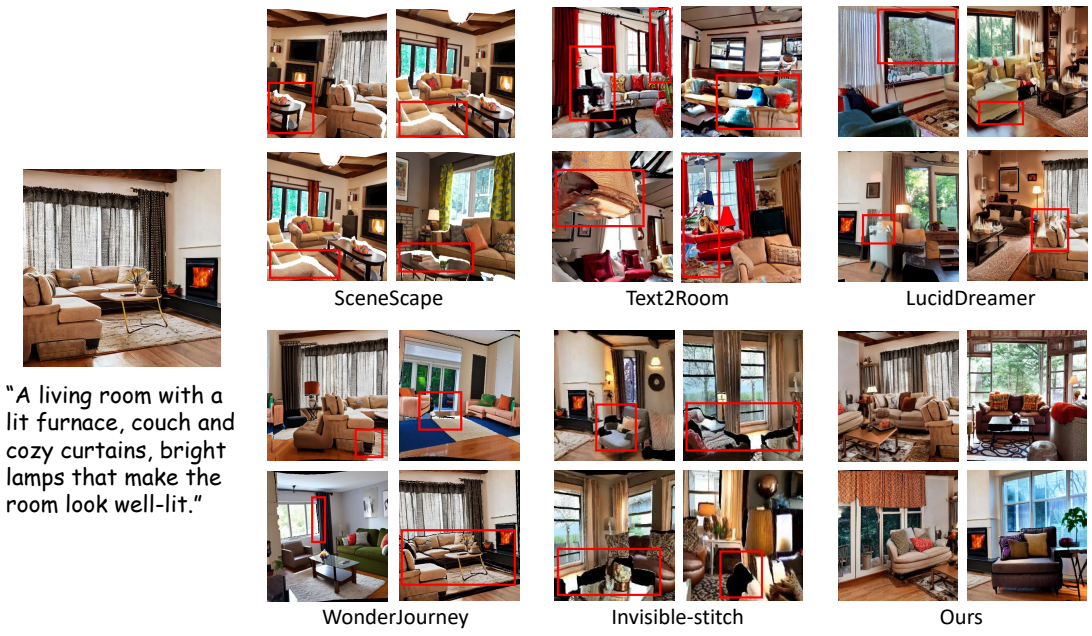

Figure 1: Qualitative comparison of our method and baselines.

Invisible-stitch Engstler et al. (2024) utilizes the proposed general-purpose depth inpainting model to complete the depth map at each viewpoint. The model incrementally lifts 2D pixels into 3D space, merging them with the existing point cloud. The obtained point cloud is used as the initial points for 3D Gaussians, thus generating a 3D scene by Gaussian splatting.

3. Additional qualitative results

Figure.1 and Figure.2 show more qualitative results. (i) SceneScape (Fridman et al., 2024), WonderJourney (Yu et al., 2024), and Invisible-stitch (Engstler et al., 2024) all create relatively complete scene structures and appear to be coherently connected at particular viewpoints. However, when the images are rendered from new viewpoints, clear breaks can be observed in the boxed regions of the rendered images, and geometric distortions are evident. (ii) Text2Room (Höllein et al., 2023) uses polygonal meshes to represent scenes. It proposes a threshold filtering scheme for mesh fusion, which leads to the possibility that not all mesh stretching regions can be detected. This results in a large number of distorted and oversmoothed regions observable in the rendered images, which is particularly noticeable in outdoor scenes. (iii) LucidDreamer (Chung et al., 2023) is currently the most visually appealing progressive scene generation framework, but we can observe artifacts and geometric distortions in the boxed parts of the rendered images. (iv) Compared to baselines, our approach contains the necessary scene structure and yields high-quality rendered results with realistic details, significantly reducing artifacts and geometric distortions.

Figure 2: Qualitative comparison of our method and baselines.

References

Shariq Farooq Bhat, Reiner Birkl, Diana Wofk, Peter Wonka, and Matthias Müller. Zoedepth: Zero-shot transfer by combining relative and metric depth. *arXiv preprint arXiv:2302.12288*, 2023.

Jaeyoung Chung, Suyoung Lee, Hyeongjin Nam, Jaerin Lee, and Kyoung Mu Lee. Luciddreamer: Domain-free generation of 3d gaussian splatting scenes. *arXiv preprint arXiv:2311.13384*, 2023.

XTuner Contributors. Xtuner: A toolkit for efficiently fine-tuning llm. https://github.com/InternLM/xtuner, 2023.

Paul Engstler, Andrea Vedaldi, Iro Laina, and Christian Rupprecht. Invisible stitch: Generating smooth 3d scenes with depth inpainting. *arXiv preprint arXiv:2404.19758*, 2024.

Rafail Fridman, Amit Abecasis, Yoni Kasten, and Tali Dekel. Scenescape: Text-driven consistent scene generation. *Advances in Neural Information Processing Systems*, 36, 2024.

Lukas Höllein, Ang Cao, Andrew Owens, Justin Johnson, and Matthias Nießner. Text2room: Extracting textured 3d meshes from 2d text-to-image models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7909–7920, 2023.

Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, and George Drettakis. 3d gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics*, 42(4): 1–14, 2023.

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

Hong-Xing Yu, Haoyi Duan, Junhwa Hur, Kyle Sargent, Michael Rubinstein, William T Freeman, Forrester Cole, Deqing Sun, Noah Snavely, Jiajun Wu, et al. Wonderjourney: Going from anywhere to everywhere. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6658–6667, 2024.