FastScene: Text-Driven Fast 3D Indoor Scene Generation via Panoramic Gaussian Splatting

Yikun Ma¹, Dandan Zhan¹, Zhi Jin¹,² *

¹Sun Yat-sen University
²Guangdong Provincial Key Laboratory of Fire Science and Intelligent Emergency Technology
{mayk25, zhandd3}@mail2.sysu.edu.cn, jinzh26@mail.sysu.edu.cn

Abstract
Text-driven 3D indoor scene generation holds broad applications, ranging from gaming and smart homes to AR/VR applications. Fast and high-fidelity scene generation is paramount for ensuring user-friendly experiences. However, existing methods are characterized by lengthy generation processes or necessitate the intricate manual specification of motion parameters, which introduces inconvenience for users. Furthermore, these methods often rely on narrow-field viewpoint iterative generations, compromising global consistency and overall scene quality. To address these issues, we propose FastScene, a framework for fast and higher-quality 3D scene generation, while maintaining the scene consistency. Specifically, given a text prompt, we generate a panorama and estimate its depth, since the panorama encompasses information about the entire scene and exhibits explicit geometric constraints. To obtain high-quality novel views, we introduce the Coarse View Synthesis (CVS) and Progressive Novel View Inpainting (PNVI) strategies, ensuring both scene consistency and view quality. Subsequently, we utilize Multi-View Projection (MVP) to form perspective views, and apply 3D Gaussian Splatting (3DGS) for scene reconstruction. Comprehensive experiments demonstrate FastScene surpasses other methods in both generation speed and quality with better scene consistency. Notably, guided only by a text prompt, FastScene can generate a 3D scene within a mere 15 minutes, which is at least one hour faster than state-of-the-art methods, making it a paradigm for user-friendly scene generation.

1 Introduction
3D models have a wide range of applications in video production, gaming, AR/VR, and other fields. However, generating high-quality 3D models typically requires professional designers to utilize specialized software with a considerable amount of time, which is inconvenient for those seeking fast 3D model generation. The development of generative models makes Text-to-3D object generation [Poole et al., 2022], [Lin et al., 2023] possible and impressive. However, the generation of 3D scenes still presents significant challenges, requiring large-scale scene reconstruction, multi-view images, and the assurance of scene realism and consistency.

Recently, some works attempt to tackle the 3D scene generation challenges. Set-the-Scene [Cohen-Bar et al., 2023] applies global-local training from text prompts and 3D object proxies, while generating controllable scenes. However, the quality and resolution of the generated scenes are unsatisfactory due to the lack of corresponding geometry. SceneScape [Fridman et al., 2024] generates long-range views, producing diverse styles. However, its view quality decreases over time due to the inpainting and depth estimation error accumulation. Text2Room [Höllein et al., 2023] and Text2NeRF [Zhang et al., 2024] gradually generate perspective novel views. Nevertheless, their incremental local operations hardly ensure scene consistency and coherence. Ctrl-Room [Fang et al., 2023] fine-tunes ControlNet [Zhang et al., 2023] for editable panorama generation, and then performs mesh reconstruction. However, Ctrl-Room tends to flatten the 3D model with limited scene quality, since it hardly generates multi-view images.

As one of the 3D representation techniques, the radiance fields methods, exemplified by Neural Radiance Fields (NeRF) [Mildenhall et al., 2020], have made significant breakthroughs. Since most NeRF-based methods suffer from slow rendering speed [Mildenhall et al., 2020], [Barron et al., 2022], rendering process acceleration becomes an important issue. Recently, 3D Gaussian Splatting (3DGS) [Kerbl et al., 2023] has achieved success in the rendering speed with high-quality. However, the typical 3DGS only takes regular images as the input. It faces challenges when handling panoramas, which are difficult to handle with existing Structure-from-Motion (SFM) [Snively et al., 2006] methods.

To address the above issues, we propose a novel Text-to-3D scene framework, called FastScene, which aims at fast generating consistent and authentic scenes with high-quality. As shown in Figure 1, our approach primarily comprises three stages. 1) In the first stage, given a text prompt, we generate a panorama by utilizing the pre-trained Diffusion360 [Feng et al., 2023]. Panorama is selected due to its ability to capture the global information and exhibit explicit ge-
2.1 Text-Driven 3D scene Generation

Recently, there has been considerable focus on 3D scene generation. Set-the-Scene [Cohen-Bar et al., 2023] introduces an agent-based global-local framework to synthesize controllable 3D scenes, while enabling diverse scene editing options. However, it suffers from shortcomings in the quality and resolution of generation scenes without corresponding geometry. While SceneScape [Fridman et al., 2024] generates consistent views by introducing a pre-trained text-to-image model, FastScene, enabling fast and high-quality scene generation, while ensuring scene consistency. Additionally, given the text prompt, there is no need to pre-design complex camera parameters or motion trajectories, which makes FastScene a user-friendly scene generation paradigm.

1) We propose a novel Text-to-3D indoor scene framework FastScene, enabling fast and high-quality scene generation, while ensuring scene consistency. Furthermore, given the text prompt, there is no need to pre-design complex camera parameters or motion trajectories, which makes FastScene a user-friendly scene generation paradigm.

2) We propose a novel panoramic view synthesis method PNVI, which adopts CVS to generate novel views with holes, and performs precision-controllable progressive inpainting to generate refined views. Additionally, to improve the inpainting quality, we synthesize a large-scale distribution-based spherical mask dataset.

3) To the best of our knowledge, we are the first to solve panoramic 3DGS from a single panorama, and the proposed FastScene is highly adaptable to existing panoramic data for reconstruction.

The rest of this paper is organized as follows. Section 2 briefly reviews the related works of this paper. Section 3 introduces the design details of the proposed FastScene. Section 4 provides experimental results for comparisons and ablation study. Conclusions are summarized in Section 5.
[Rombach et al., 2022], and possesses the capability to generate scenes in various styles. However, the view quality of SceneScape is reliant on geometric priors and diminishes over time due to the inpainting and depth error accumulation. More recently, both Text2Room [Höllein et al., 2023] and Text2NeRF [Zhang et al., 2024] rely on incremental frameworks to generate new perspectives on a per-image basis. However, their incremental local operations can hardly guarantee scene consistency and coherence. Later on, Ctrl-Room [Fang et al., 2023] proposes to encode text input and convert scene code into a 3D bounding box for editing. Subsequently, it generates panoramas by fine-tuning ControlNet [Zhang et al., 2023], and reconstructs mesh through Posision reconstruction [Kazhdan et al., 2006] and MVS-texture [Waechter et al., 2014]. However, Ctrl-Room struggles to generate high-quality 3D models, and tends to flatten the 3D model due to the limited number of generated views.

2.2 Text-Driven Panorama Generation

Unlike 2D images, panoramas cover the 360° × 180° field of view, which provides more 3D scene information. Text2Light [Chen et al., 2022] synthesizes panorama images from text input via a multi-stage auto-regressive generative model. However, it ignores the boundary continuity of the panorama, resulting in an open-loop content. MVDiffusion [Tang et al., 2023] generates high-resolution panoramas by fine-tuning a pre-trained text-to-image diffusion model. However, artifacts usually appear on the “sky” and “floor” views, which decreases the realism of generated scenes. StitchDiffusion [Wang et al., 2024b] crops the left and right sides of the panorama to maintain the scene continuity. However, the cracks at the seams are still noticeable. Diffusion360 [Feng et al., 2023] proposes a circular blending strategy to maintain the geometry continuity, which generates high-resolution boundary-continuous panoramas.

2.3 Novel View Synthesis

Novel view synthesis is a popular area of significant interest. Early methods rely on multi-view images and attempt to incorporate the knowledge from epipolar geometry to perform smooth interpolation between the different views [Chen and Williams, 1993], [Debevec et al., 1996]. Some methods synthesize novel views by deep networks from a few images [Sajjadi et al., 2022], [Mirzaei et al., 2023]. In contrast, [Gu et al., 2023], [Shen et al., 2023] allow for generating novel views from a single image.

A significant breakthrough in novel view synthesis is NeRF [Mildenhall et al., 2020] and its derivative works [Barron et al., 2021], [Barron et al., 2022], [Chen et al., 2023]. The rendering speed of most radiance-based methods is slow, accelerating rendering becomes an important but challenging problem, with representative works such as Instant-NGP [Müller et al., 2022] and 3DGS [Kerbl et al., 2023]. Some NeRF-based works [Wang et al., 2024a], [Chen et al., 2024] attempt to synthesize panoramic novel views. However, since SFM struggles to handle panoramas due to its unique structure [Snavely et al., 2006], it is difficult to utilize original 3DGS for panorama rendering.

3 Method

3.1 Overview

As shown in Figure 1, given a text prompt $P$, we first use Diffusion360 [Feng et al., 2023] to generate the corresponding panorama $S_0$, and then employ EGformer [Yun et al., 2023] to estimate the depth map $D_0$. Thereafter, given a new camera pose $P_n$, we perform CVS to obtain the corrupted panorama $S_n$ with holes. To fill these holes, we propose PNVI, which gradually inpaints perspective cubemap views $S_n(i = 0, 1, \ldots, 5)$ rather than directly inpaints the panorama. Subsequently, these clean cubemap images are reprojected equidistantly to obtain the clean panorama. We then replace non-hole pixels in the inpainted panorama with their original values to obtain the novel panorama $S_n$. Similarly, iterating PNVI multiple times results in numerous novel clean panoramic views. As COLMAP [Schonberger and Frahm, 2016] does not support panoramic inputs, we employ MVP to generate the corresponding perspective views, followed by 3DGS to implement the 3D scene reconstruction.

3.2 Text-Driven Panorama Generation and CVS

Compared to perspective views, a key geometric characteristic of panorama is the continuity of the boundaries. Additionally, the panorama encompasses information about the entire scene and exhibits explicit geometric constraints, which is beneficial for our subsequent processing. Thus, we utilize Diffusion360 [Feng et al., 2023] for text-to-panorama generation, which adopts the blending strategy to maintain the geometry continuity. After that, we estimate the depth map using EGformer [Yun et al., 2023] to capture the spatial information of the scene. Then, we propose CVS to obtain a new panoramic view under a given camera pose, as shown in Figure 2. According to the theory of equidistant projection on the spherical panorama, we can project a 2D image of size $1024 \times 512$ onto a sphere, where the latitude range is 180° and the longitude range is 360°. The calculation for the latitude angle $\theta_a$ and longitude angle $\phi_a$ are as follows:

\[
\theta_a = \frac{\pi y_a}{W}, \quad (1)
\]

\[
\phi_a = \frac{2\pi x_a}{H}, \quad (2)
\]

where $x_a$ and $y_a$ represent the image coordinates of coordinate system $a$, while $W$ and $H$ represent the width and height of the panorama, respectively. We then utilize the triangle transformation to obtain the spherical basis coordinates:

\[
a_x = \cos \theta_a \cos \phi_a, \quad (3)
\]

\[
a_y = \sin \phi_a, \quad (4)
\]

\[
a_z = -\cos \theta_a \sin \phi_a, \quad (5)
\]

afterward, we multiply the depth $d$ by the 3D coordinates $a_x, a_y, a_z$ to initial the spherical coordinates $C_a$:

\[
C_a = (d \cdot a_x, d \cdot a_y, d \cdot a_z). \quad (6)
\]

Given a new camera pose $P_n$, we take it as the origin of the new spherical coordinate system $n$, and subtract the original coordinates $C_a$ from the new origin $P_n$ to get the new spherical coordinates:

\[
C_n = (n_x, n_y, n_z) = \frac{C_a - P_n}{|C_a - P_n|}, \quad (7)
\]
Figure 2: Given a new camera pose $P_n$, the calculation for movement in spherical coordinates.

Then, we reproject the coordinates $C_n$ to the new coordinate system $n$:

$$
\theta_n = \arctan \frac{n_y}{\sqrt{n_x^2 + n_z^2}}, \quad (8)
$$

$$
\phi_n = \arctan \frac{-n_z}{n_x}, \quad (9)
$$

$$
x_n = \frac{\phi_n}{2\pi} W, \quad (10)
$$

$$
y_n = \frac{\theta_n}{\pi} H, \quad (11)
$$

where $\theta_n$ and $\phi_n$ denote the latitude and longitude of the novel view, and $x_n$ and $y_n$ represent the image coordinates of coordinate system $n$.

We summarize equations (1) to (11) as a mapping $F$ from $(x_a, y_a)$ to $(x_n, y_n)$:

$$(x_n, y_n) = F(x_a, y_a). \quad (12)$$

Therefore, we only need to determine if the mapped pixels $(x_n, y_n)$ lie within the panorama. If they are inside, we keep the normal RGB values, otherwise we set them as holes with a value 255:

$$
S_n = \begin{cases} 
\text{normal}, & \text{if } (x_n \leq W, y_n \leq H, d_n > 0) \\
255, & \text{otherwise}
\end{cases} \quad (13)
$$

Correspondingly, we can obtain the mask image $M_n$ (with value 0 for normal regions and 1 for unseen) for inpainting:

$$
M_n = \begin{cases} 
0, & \text{if } (S_n = \text{normal}) \\
1, & \text{otherwise}
\end{cases} \quad (14)
$$

### 3.3 Progressive Novel View Inpainting

After CVS, we obtain multi-view panoramas with holes. To reconstruct the scene using 3DGS, we need to fill these holes. To this end, we propose the PNV1 to obtain clean novel views. Due to the lack of indoor panoramic datasets with mask information to retrain the inpainting network, we construct a new dataset, as detailed in Section 4.2. We endeavored to conduct direct panorama inpainting, yet observed that with an increasing distance of movement, a plethora of spurious shadows manifested along the peripheries of the panorama. Therefore, E2C is utilized to obtain six cubemap images from one panorama, and cubemap inpainting is conducted using the retrained AOT-GAN [Zeng et al., 2022]. After that, C2E is utilized to form the panorama. Finally, we replace non-hole pixels in the inpainted panorama with their original values to obtain the novel panorama $S_n$.

However, when directly moving the camera to large poses, the hole-to-image area ratios become extensive, raising difficulties for inpainting, irrespective of the model training quality. To address the aforementioned issue, we propose a progressive inpainting mode, as shown in Figure 3, which enables inpainting in large camera poses. Specifically, assuming we move the camera along the X-axis by a distance of 0.33 meters, the hole-to-image area ratio of the novel view image increases to 64.3%, which means more than half of the images are with holes, as reported in Table 1. Therefore, we decide to divide the long distance into small moves (e.g., 0.02 meters per move) to relieve the long distance inpainting difficulty. In this way, the hole-to-image ratio is only 15% at each move. By progressively moving from $P_0$ to $P_n$, we can obtain a clean view at the endpoint.

| Pose_X (m) | 0.03 | 0.27 | 0.15 | 0.09 | 0.03 | -0.02 |
| Mask (%)   | 64.3 | 58.7 | 51.9 | 43.2 | 33.2 | 17.7 |
| Pose_Y (m) | 0.20 | 0.16 | 0.12 | 0.09 | 0.05 | 0.02 |
| Mask (%)   | 28.9 | 24.2 | 19.6 | 15.3 | 11.2 | 7.1 |
| Pose_Z (m) | 0.33 | 0.27 | 0.21 | 0.15 | 0.09 | 0.03 |
| Mask (%)   | 62.2 | 56.7 | 50.0 | 41.9 | 31.5 | 16.7 |

Table 1: The camera movement from the current pose along different axes and their corresponding hole-to-image area ratios. Mask (%). Sign ‘-’ indicates moving towards the negative direction of axis.

### 3.4 Panoramic 3D Gaussian Splatting

The original inputs for 3DGS are multiple RGB perspective views. Following the COLMAP [Schonberger and Frahm, 2016] pipeline, sparse point clouds and camera parameters are obtained. Nevertheless, algorithms within COLMAP pertaining on perspective views exclusively exhibit inadequacies when confronted with panoramic perspectives, leading to a disorderly reconstruction outcome. As shown in Figure 4a, assuming the camera moves along $x, y, and z$ axes, the adoption of the original COLMAP fails to produce accurate point clouds and camera poses. This arises from the distinctive distortions and intricacies inherent in panoramas, making the application of conventional SFM arduous in the endeavor to align spatial information across diverse viewpoints.

Therefore, we introduce MVP to solve the aforementioned problem. Specifically, given the panorama $S$ with...
size $W \times H$ and the requirements for $n$ perspective images $(V_1, V_2, ..., V_n)$ with size $R \times R$. Firstly, we calculate the rotation matrix $R_i$ for each camera. For each perspective view $V_i$ ($1 \leq i \leq n$), we define a projection mapping function $P(S, V_i)$, which maps the pixels of the panorama to the perspective view. By projecting the panoramic pixels $(m, q), (0 \leq m < W, 0 \leq q < H)$, new projected coordinates $(j, k), (0 \leq j, k < R)$ for the perspective view can be obtained. Then, we send the multi-view perspective images to COLMAP to obtain the point clouds required by 3DGS. As shown in Figure 4b, for multi-view panoramic inputs, our method enables the generation of accurate point clouds and camera poses, thereby allowing for seamless processing using 3DGS. The loss function $L$ is defined as the weighted sum of $L_1$ and $L_{DSSIM}$ [Wang et al., 2004]:

$$L = (1 - \lambda)L_1 + \lambda L_{DSSIM}. \quad (15)$$

we follow [Kerbl et al., 2023] to set $\lambda = 0.2$.

4 Experiments

4.1 Implementations Details

We implement our method on PyTorch. We use the pre-trained Diffusion360 [Feng et al., 2023] and EGformer [Yun et al., 2023] for panorama generation and depth estimation, respectively. We retrain the AOT-GAN [Zeng et al., 2022] on our synthesized dataset for inpainting, described in Section 4.2. We choose CLIP Score [Hessel et al., 2021], Natural Image Quality Evaluator (NIQE) [Mittal et al., 2012b], and Blind/Referenceless Image Spatial Quality Evaluator (BRISQUE) [Mittal et al., 2012a] to evaluate the rendering quality in an unsupervised manner. It takes about 15 minutes to generate a complete scene on a single NVIDIA RTX A6000 GPU with 49G memory. Specifically, panorama generation takes 10 seconds, the PNVI process takes approximately 2 minutes, acquiring 3DGS training data requires around 3 minutes, and scene generation takes 10 minutes.

4.2 Panoramic Inpainting Dataset

Due to the absence of panoramic datasets with our mask distribution, it is essential to generate a corresponding dataset. Specifically, we select the synthetic dataset Structured3D [Zheng et al., 2020], which comprises 21k photorealistic panoramic scenes. We select 14k images with complete scenes that are more realistic. Subsequently, for each panorama, we generate 16 types of masks using equations (12) to (14), corresponding to eight movement directions on the coordinate axis, with two movement units of 0.02$m$ and 0.04$m$ for each direction. Then we perform E2C projection for each panorama and mask image. Finally, there are a total of 84k perspective RGB images and 1344k masks. After obtaining the dataset, we retrain AOT-GAN [Zeng et al., 2022], with all training and testing sizes set as $512 \times 512$.

4.3 Comparisons with Other Methods

To validate the effectiveness of our method, we conduct quantitative and qualitative comparisons with previous indoor scene generation methods, including Text2Room [Höllein et al., 2023], Set-the-Scene [Cohen-Bar et al., 2023], and SceneScape [Fridman et al., 2024]. We render 30 images of each scene for evaluation.

Quantitative Comparison. By giving an identical text prompt input, we test the generative performance of different methods. Since conventional image quality assessment metrics, such as PSNR and SSIM, are not applicable to our task, we adopt unsupervised evaluation metrics.

As reported in Table 2, Text2Room performs modestly due to the lack of global consistency. SceneScape suffers from decreased image quality caused by accumulated errors during long-distance movements. Set-the-Scene exhibits limited perceptual performance due to its lower resolution and texture quality. On the contrary, our method not only achieves superior performance in terms of CLIP Score, NIQE, and BRISQUE metrics, but also demonstrates the fastest generation speed. A fast generation process is important, since it is an obvious advantage for user-friendly tasks.

Qualitative Comparison. Furthermore, to comprehensively validate the performance of our FastScene, we present the qualitative comparison results with other scene generation methods. We provide the same text prompt, such as common indoor scenes: bedroom, living room, dining room, etc., and then obtain the generation results of different methods. As shown in Figure 5, Text2Room [Höllein et al., 2023] can generate faithful local views, but it fails to ensure consistency across the entire scene. SceneScape [Fridman et al., 2024] has the ability to generate long-range immersive views. However, as the distance increases, the accumulation of errors results in a detrimental loss of details. Set-the-Scene [Cohen-Bar et al., 2023] possesses the ability to generate editable scenes. However, its rendered images are blurry and texture quality is inadequate to meet perceptual needs. In comparison, our method generates high-quality scenes in a fast way, and ensures the scene consistency as well. More scene generation results can be found in our Supplementary Material.

In conclusion, both quantitative and qualitative comparison experiments confirm that our method can rapidly and effectively generate globally consistent scenes with high-quality.
Figure 5: Qualitative comparisons with other methods. For each methods, we show the rendering views for the 1st and 5th frames. Our method generates high-quality scenes from the same text prompts, while maintaining the scene consistency well.

<table>
<thead>
<tr>
<th>Methods</th>
<th>CLIP ↑</th>
<th>NIQE ↓</th>
<th>BRISQUE ↓</th>
<th>Time(min/scene)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text2Room</td>
<td>28.1</td>
<td>5.4</td>
<td>28.4</td>
<td>70</td>
</tr>
<tr>
<td>Set-the-Scene</td>
<td>23.8</td>
<td>9.3</td>
<td>51.6</td>
<td>155</td>
</tr>
<tr>
<td>SceneScape</td>
<td>24.7</td>
<td>4.4</td>
<td>32.3</td>
<td>110</td>
</tr>
<tr>
<td>Ours</td>
<td>29.0</td>
<td>3.9</td>
<td>20.6</td>
<td>15</td>
</tr>
</tbody>
</table>

Table 2: Quantitative comparison with other methods, with all results tested on the same hardware device.

4.4 Extension Experiments on Panoramic Datasets

To validate the adaptability of our PNVI and MVP on existing panoramas for 3DGS, we conduct extension experiments on the Matterport3D 1k, 2k [Chang et al., 2017], and Replica360 4K [Straub et al., 2019] datasets, containing panoramas at resolutions of 1K, 2K, and 4K, respectively. As shown in Figure 6, our method is capable of reconstructing 3D scenes from panoramas at different resolutions.

Figure 6: The reconstruction results of indoor panoramic datasets, validating that our method can effectively transfer to 360° datasets.

Furthermore, to demonstrate the effectiveness of our method, we compare the performance with panoramic novel views synthesis works on Replica360 4K: DS-NeRF [Deng et al., 2021], 360FusionNeRF [Kulkarni et al., 2023], and PERF [Wang et al., 2024a]. These NeRF-based methods inherently lack the ability to infer occluded content and have insufficient geometric constraints for panoramic structures. As a result, they suffer from varying degrees of blurriness and reduced quality, as shown in Figure 7 and Table 3. Among them, PERF exhibits relatively satisfactory results, but it lacks consideration of panoramic geometric information, and there is a certain degree of quality degradation. On the contrary, we design PNVI and MVP to fully consider the constraints of the panoramic structure, while employing 3DGS rather than NeRF architecture, resulting in higher rendering quality in both quantitative and qualitative performance.

The extension experiments further demonstrate that our method can be extended to existing panoramas and perform high-quality novel view synthesis.

<table>
<thead>
<tr>
<th>Methods</th>
<th>PSNR ↑</th>
<th>SSIM ↑</th>
<th>LPIPS ↓</th>
</tr>
</thead>
<tbody>
<tr>
<td>DS-NeRF</td>
<td>23.29</td>
<td>0.834</td>
<td>0.265</td>
</tr>
<tr>
<td>SinNeRF</td>
<td>22.70</td>
<td>0.826</td>
<td>0.251</td>
</tr>
<tr>
<td>DietNeRF</td>
<td>23.24</td>
<td>0.836</td>
<td>0.291</td>
</tr>
<tr>
<td>360FusionNeRF</td>
<td>21.54</td>
<td>0.833</td>
<td>0.245</td>
</tr>
<tr>
<td>PERF</td>
<td>23.49</td>
<td>0.838</td>
<td>0.244</td>
</tr>
<tr>
<td>Ours</td>
<td><strong>23.52</strong></td>
<td><strong>0.841</strong></td>
<td><strong>0.245</strong></td>
</tr>
</tbody>
</table>

Table 3: Quantitative comparisons on Replica360 dataset. Our FastScene achieves better quantitative evaluation results than other views rendering methods.

4.5 Ablation Studies

To validate the necessity of our inpainting mode and the effectiveness of the progressive inpainting strategy in PNVI, we design two corresponding ablation studies for different inpainting modes:
Directly inpainting panorama. We first retrain the AOT-GAN on our synthesized panoramic dataset, and then directly perform inpainting on panoramas. We find the performance is ideal for small-distance movements, as shown in Figure 8(a).

However, as the movement distance increases, noticeable distortion and edge-blurring artifacts appear, as shown in Figure 8(b). This is due to the accumulated errors in depth estimation and the projection errors in the inpainting process. Additionally, due to discrepancies between the truth depth values in the dataset and our estimations, the distribution of holes is not entirely consistent between the training and inference stages.

Inpainting for a large distance. To validate the effectiveness of our progressive inpainting strategy, we perform inpainting on novel views with large camera poses, rather than incrementally moving. According to Figure 8(c), it is evident that directly inpainting large poses results in serious artifacts, which affects subsequent processing. When there is a large hole-to-image ratio, it becomes challenging to ensure the generation quality, thus affecting the consistency of the overall scene. By progressively inpainting the cubemap images, our PNVI strategy can address distortion and edge-blurring issues, as shown in Figure 8(a).

Table 4 reports the quantitative comparisons of different inpainting modes, where our FastScene achieves the best performance in scene generation. In summary, the ablation studies further demonstrate the effectiveness of our method.

Table 4: Ablation studies for directly, large-distance, and cubemap inpainting. We retrain AOT-GAN on our synthetic dataset.

<table>
<thead>
<tr>
<th>Methods</th>
<th>CLIP ↑</th>
<th>NIQE ↓</th>
<th>BRISQUE ↓</th>
<th>TIME(min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Directly</td>
<td>27.3</td>
<td>6.8</td>
<td>45.1</td>
<td>14</td>
</tr>
<tr>
<td>Large-distance</td>
<td>25.7</td>
<td>7.4</td>
<td>42.3</td>
<td>11</td>
</tr>
<tr>
<td>Ours</td>
<td>29.0</td>
<td>3.9</td>
<td>20.6</td>
<td>15</td>
</tr>
</tbody>
</table>

5 Conclusion

We propose a fast Text-to-3D indoor scene generation framework FastScene, exhibiting satisfactory scene quality and consistency. For users, FastScene only requires a text prompt without designing motion parameters, and provide a complete high-quality 3D scene in only 15 minutes. The proposed PNVI with CVS can generate consistent novel panoramic views, while MVP projects them into perspective views, facilitating 3DGS reconstruction. Extensive experiments demonstrate the effectiveness of our method. FastScene provides a user-friendly scene generation paradigm, and we believe it has wide-ranging potential applications. In future work, we will focus on 3D scene editing and multimodal learning.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No.62071500), and Shenzhen Science and Technology Program (Grant No. JCYJ2023080711107015).

References


