SiteBuilder: Sitemap-Grounded 3D Completion for Atrium Scene

Sizhe Song
The Hong Kong University of Science and Technology
Hong Kong, China
ssongad@cse.ust.hk

S.-H. Gary Chan
The Hong Kong University of Science and Technology
Hong Kong, China
gchan@cse.ust.hk

Abstract

Many indoor applications, such as mobile navigation and interior redesign, require the 3D model of the scene to structurally reflect the reality. We study the problem of 3D completion for *atrium scene*, i.e., a large and visually consistent multi-storey indoor space, given its sitemap (composed of the layout of each floor) and possibly an incomplete 3D model of the site. Existing approaches either consider no structural constraint or assume a cellular layout of independent rooms and regions, and hence cannot be extended to the atrium case. We propose SiteBuilder, a novel sitemap-grounded framework to complete the 3D atrium scene. SiteBuilder first transforms the sitemap into a bootstrap mesh to extract the geometry and appearance information, based on which along with the key frames rendered from the incomplete 3D model (if any), it then completes the scene using conditional video diffusion. We conduct extensive experiments on many complex atrium sites, and show that SiteBuilder, through integrated and well-guided generation, produces realistic complete 3D scenes that are globally consistent and structurally correct.

CCS Concepts

• **Computing methodologies** → *Computer graphics*.

Keywords

Scene Completion, Video Generation, Sitemap

ACM Reference Format:
Sizhe Song and S.-H. Gary Chan. 2026. SiteBuilder: Sitemap-Grounded 3D Completion for Atrium Scene. In *Proceedings of the 34th ACM International Conference on Multimedia (MM '26), November 10–14, 2026, Rio de Janeiro, Brazil.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3767308.3835226

1 Introduction

We consider 3D modeling of a physical *atrium scene*, i.e. a large seamless indoor site connected by floors. Examples of atrium include shopping mall, museum, and convention center. The 3D atrium scene, represented usually as point cloud, 3D gaussian splatting or NeRF, is fundamental for many indoor applications such as navigation, simulation and interior redesign.

Given possibly an incomplete (or partial) 3D model of the site, we study the novel problem of building a 3D atrium scene that is complete (covering the entire site), realistic (structurally faithful), and globally and visually consistent (coherent and integrated). Since the structure of an atrium has been captured by a sitemap, defined as the collection of the layout of each floor (the so-called "floorplan"), we adopt it in our work to guide the generation of the missing geometry and appearance.

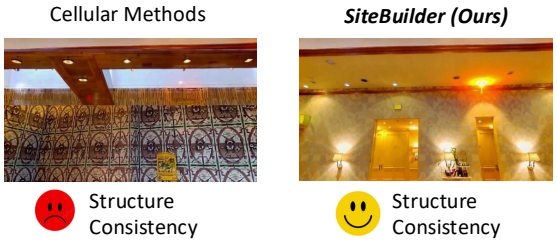


Figure 1: SiteBuilder generates atrium 3D scenes that are both structurally correct and visually consistent.

Many existing methods on 3D modeling assumes no structural constraint [6, 14, 16, 21, 35–37, 48, 49, 51, 65, 66], and hence the 3D scene generated does not reflect the reality. Some others mainly assume *cellular-layout* scenes [1, 5, 8–11, 13, 34, 44, 52, 58], where the site is essentially a simple concatenation of independent enclosed rooms or regions, and the 3D scene is completed by generating each small space individually. Note that atrium scenes are fundamentally different. They are typically larger in scale and more complex in structure, containing connected open spaces that span multiple regions or floors. This integrated spatial characteristic makes it difficult to be decomposed into independent local units. As a result, directly applying previous methods to atrium scenes likely result in global inconsistency, especially around large open areas and cross-floor connections, as demonstrated in Figure 1. Generating 3D atrium scene still remains an open and challenging problem.

In order to complete the 3D scene of an atrium, we need to answer two questions. First, how should we extract the geometry and appearance information out from a 2D sitemap? Second, how can we incorporate the extracted information into the generative process so that the completed scene reflects the physical building layout with consistency?

We propose SiteBuilder, a novel sitemap-grounded 3D completion framework for atrium scenes. Our key observation is that a realistic 3D scene is jointly determined by geometry and appearance: the model should not only place structures at the correct locations, but also assign visually consistent semantic content to them. SiteBuilder hence first transforms the sitemap into a bootstrap mesh. By turning the 2D representation into a 3D one, we are able to extract the geometry and appearance information of the

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.
MM '26, Rio de Janeiro, Brazil
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2213-4/2026/11
https://doi.org/10.1145/3767308.3835226

target scene as two pixel-wise depth and semantic sequences. SiteBuilder then feeds such two sequences into a Geometry-Appearance Encoder (GAE) to obtain the conditions for a video generative network. The incomplete input 3D model grounded on the sitemap, if presented, is used to render the key frames for video generation. Eventually, the depth condition constrains geometric layout, while the semantic condition guides appearance generation.

We summarize the contributions of SiteBuilder as follows:

- *The novel problem of completing a 3D atrium scene and its framework SiteBuilder:* We, for the first time, extend 3D scene completion problem from isolated cellular scenes to large and integrated atrium scenes, and propose a framework for it.
- *A bootstrap mesh to extract geometry and appearance information from a sitemap:* We propose a bootstrap mesh method that handles large-scale and diverse building sitemaps, transforming a collection of 2D floorplans into a coarse 3D structural reference from which depth and semantic sequences are extracted.
- *Geometry-Appearance Encoder (GAE) to condition and guide generation:* We propose a GAE that uses the previously extracted depth and semantic sequences as condition for the video generative network.

We conduct extensive qualitative and quantitative experiments on a diverse and complex set of atrium scenes, demonstrating that SiteBuilder produces strong completion results and achieves clear advantages over previous approaches. The rest of this paper is organized as follows. After reviewing related literature in Section 2, we present the methodology and technical details of SiteBuilder in Section 3. We report illustrative experimental results in Section 4, and conclude in Section 5.

2 Related Work

2.1 3D Scene Completion

Recent work on 3D scene completion and generative reconstruction has demonstrated that plausible geometry and appearance may be generated from text, sparse views, or incomplete observations using radiance-field and Gaussian representations [24, 32, 41], text-conditioned scene reconstruction pipelines [6, 14, 16, 36, 48, 49, 66], and learned priors for indoor scene synthesis [21, 35, 37, 51, 65]. These methods have substantially improved scene quality and scalability, ranging from immersive environment generation [2, 60, 64] to feed-forward or large-scale 3D generative models [19, 40, 56]. However, most of the methods focus on visual plausibility and local coherence, and have not considered how to enforce consistency with the physical building structure. As a result, although they can generate visually consistent rooms or local regions, they cannot match the structure of, and be extended to, a physical atrium, which is characterized by large open multi-storey space, long-range visibility, and cross-floor connections.

2.2 Structure-Guided Modeling

A parallel line of research introduces explicit structural signals such as floorplans, layouts, room proxies, or programmatic scene descriptions. Early floorplan-to-scene and layout-conditioned methods translate 2D layouts into 3D scenes or use semantic layouts to constrain generation [1, 5, 9, 52, 58]. Other methods improve controllability at the room level through layout constraints, proxy rooms, structured home generation, scene assembly, etc. [8, 10, 11, 13, 34, 44]. More recent work explores layout editing and planning with language models or program libraries [12, 17, 54, 59], as well as floorplan generation, tokenization, and reconstruction [7, 33, 38, 45, 67]. While these methods respect architectural structure, most of them are designed for cellular indoor layouts, room-scale decomposition, or object arrangement problems. By contrast, our work targets large real multi-storey atrium scenes, where the sitemap is composed of multiple floorplans and the completed 3D scene must remain globally consistent across vertically connected and visually continuous open spaces.

2.3 Generative Networks for Scene Completion

Foundational image diffusion models [42, 43] have enabled text-to-3D pipelines that optimize 3D representations from strong 2D priors [26, 39, 63]. A large body of follow-up work has studied multi-view-consistent image generation for 3D reconstruction, including single-image-to-3D and view-consistent diffusion models [27–29, 31, 46, 47, 50, 55], geometry-aware or correspondence-aware novel-view synthesis methods [18, 20, 22, 23, 25, 30, 57, 61], and video-based generative models [3, 15, 53, 68]. These models provide powerful priors to produce coherent geometry and appearance, making them attractive backbones for scene completion. However, these methods do not explicitly enforce the structure of a physical atrium scene. In contrast, SiteBuilder is sitemap-grounded that we ensure the real building structure is faithfully reflected in our final completed scene model.

3 Methodology

This section is organized as follows. In Section 3.1, we define the atrium completion objective, inputs, and notation. In Section 3.2, we present the end-to-end pipeline of SiteBuilder from sitemap and optional partial scene to completed 3D output. In Section 3.3, we detail sitemap-to-mesh conversion and floor-height estimation for structural grounding. In Section 3.4, we explain how depth and semantic controls are encoded to guide structure-faithful and appearance-consistent generation.

3.1 Problem Formulation

Given a sitemap $\mathcal{M}$ and an optional incomplete 3D scene model $\mathcal{G}_0$, our goal is to predict a complete 3D scene $\hat{\mathcal{G}}$ that is both structurally faithful to the building layout and visually coherent. In this paper, we represent both input and output 3D scenes with 3D Gaussian Splatting (3DGS). The overall mapping can be written as

$$\hat{\mathcal{G}} = \Phi\left(\mathcal{M}, \mathcal{G}_0\right), \tag{1}$$

where $\mathcal{G}_0 = \varnothing$ when no partial model is available. The sitemap is defined as a set of floorplans

$$\mathcal{M} = \left\{\mathcal{F}^1, \mathcal{F}^2, \cdots, \mathcal{F}^L\right\}, \tag{2}$$

where L is the number of floors. Each floorplan $\mathcal{F}^{(l)}$ contains a set of semantic 2D elements

$$\mathcal{F}^{(l)} = \left\{\left(\mathbf{p}_i^{(l)}, y_i^{(l)}\right)\right\}_{i=1}^{N_l}, \tag{3}$$

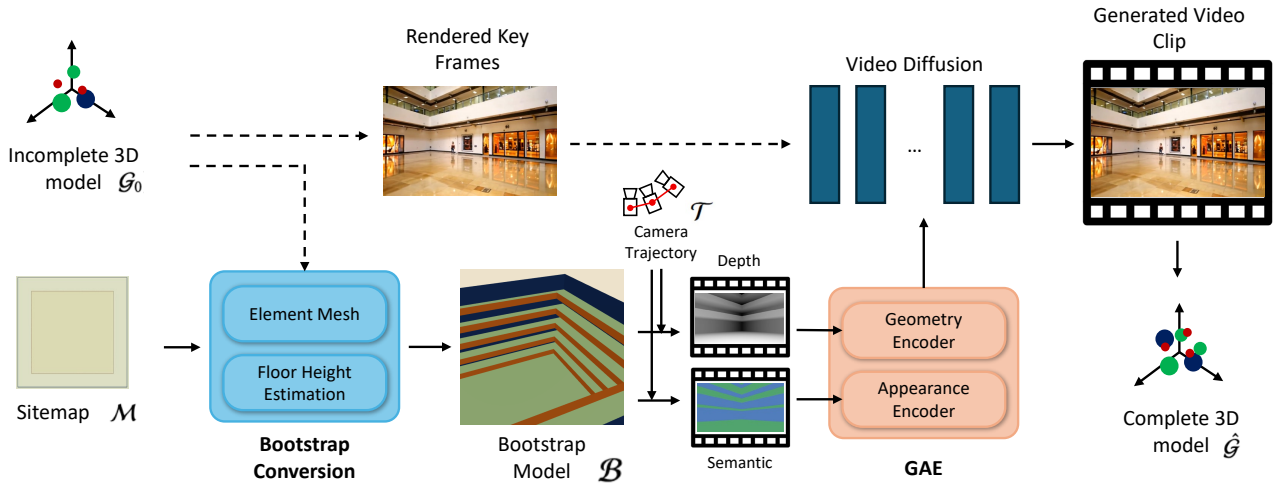
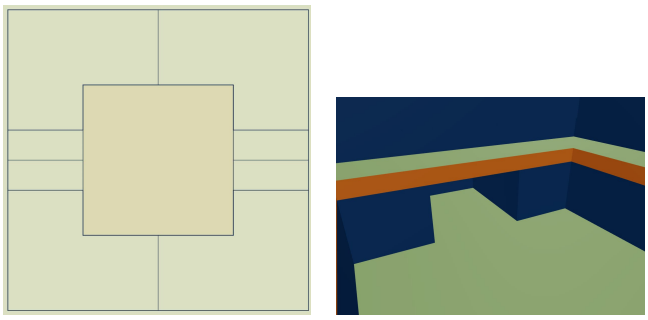


Figure 2: The SiteBuilder framework. Given a sitemap and an optional incomplete 3DGS model grounded on it, we first build a bootstrap mesh, then render depth/semantic sequences and optional key frames (dashed line) along sampled camera trajectories. The Geometry-Appearance Encoder (GAE) converts the depth and semantic sequences into geometry and appearance conditions that guide video diffusion, and generated frames are fused with camera poses to reconstruct the completed 3D scene.



(a) An example sitemap **(b)** The bootstrap mesh

Figure 3: An example of sitemap-to-bootstrap conversion in SiteBuilder. The sitemap is a stack of multiple floorplans of the target atrium scene. Different colors in the bootstrap mesh refers to different elements (e.g. walls, floors, etc.) in the sitemap.

with $\mathbf{p}_i^{(l)} \in \mathbb{R}^2$ denoting the 2D coordinates and $y_i^{(l)} \in \mathcal{Y}$ denoting the element type. In our setting, $\mathcal{Y}$ includes walls, handrails, stairs, floors, and ceilings. Each type of elements could also have some specific attributes if needed, which is too tedious to be listed here.

When provided, the incomplete 3DGS model $\mathcal{G}_0$ is grounded to the same sitemap coordinate frame so that its horizontal coordinates are the same as $\mathcal{M}$, with an additional vertical axis. This shared reference frame allows us to directly fuse 2D structural priors from the sitemap with 3D observations from the partial scene during completion.

3.2 Framework

Figure 2 presents the overall framework of SiteBuilder. SiteBuilder transforms the sitemap $\mathcal{M}$ into a bootstrap mesh $\mathcal{B}$ that lifts 2D structure into a coarse 3D representation of the building. A camera trajectory $\mathcal{T} = \{\pi_t\}_{t=1}^{T}$ is then sampled to cover unseen regions while revisiting seen regions for global consistency. Along this trajectory, we render frame-wise and pixel-wise depth and semantic maps from $\mathcal{B}$ as $\{\mathbf{D}_t, \mathbf{S}_t\}_{t=1}^{T}$, and when an incomplete model $\mathcal{G}_0$ is available, we also render key frames $\{\mathbf{K}_t\}_{t=1}^{T}$. The rendered structural cues are encoded by a Geometry-Appearance Encoder (GAE) to form geometry and appearance conditions, while key frames provide additional image conditions to preserve observed content. These conditions guide a video diffusion model to generate $\{\hat{\mathbf{I}}_t\}_{t=1}^{T}$ along $\mathcal{T}$, and the completed scene $\hat{\mathcal{G}}$ is reconstructed from generated frames and their corresponding camera poses.

3.3 Bootstrap Mesh

The bootstrap mesh method converts the multi-floor 2D sitemap into a coarse yet structurally meaningful 3D mesh. Given floorplans $\mathcal{M}$, SiteBuilder constructs floor-level meshes and stacks them along the vertical axis to obtain the global bootstrap mesh $\mathcal{B}$. The representation is intentionally lightweight so that it can be rendered efficiently for later condition generation. Figure 3 shows an example conversion from an input sitemap to its bootstrap mesh.

A hyperparameter involved is the floor height h. When no input scene model is provided, we use a default empirical value $\bar{h}$ for all floors. When $\mathcal{G}_0$ is available, SiteBuilder estimates h from grounded 3D observations and then uses this estimate during floor stacking. In each floor, elements are converted into naive planar primitives: walls are vertical planes of height h, floors and ceilings are horizontal planes (possibly with holes), handrails are vertical planes

of height $h/2$, and stairs are sloped planes connecting adjacent floors. Algorithm 1 summarizes bootstrap construction. For each floor index l, we compute its base elevation, parse all elements, and convert each element into a naive mesh primitive via Algorithm 2.

Algorithm 3 explains floor-height estimation, where we optimize a surface-distance objective. Let $C = \{\mathbf{c}_n\}_{n=1}^{N}$ be Gaussian centers from $\mathcal{G}_0$. For each candidate height h, we build a provisional bootstrap mesh $\mathcal{B}(h)$ and define

$$E(h) = \frac{1}{N}\sum_{n=1}^{N} d(\mathbf{c}_n, \mathcal{B}(h)),$$
$$d(\mathbf{c}_n, \mathcal{B}(h)) = \min_{\mathbf{x}\in \mathrm{surf}(\mathcal{B}(h))} \|\mathbf{c}_n - \mathbf{x}\|_2. \tag{4}$$

We then solve

$$h' = \arg\min_{h\in[h_{\min}, h_{\max}]} E(h). \tag{5}$$

This floor-height estimation objective uses a point-to-surface distance and is very similar in spirit to a directed Chamfer distance. In practice, we use a coarse-to-fine strategy. In each round we sample candidate heights in the current window, select the best one by average distance, and shrink the window around it.

In our current implementation, we only instantiate a simple element set mentioned above to build the bootstrap mesh. However, SiteBuilder is designed as a general framework such that users can define additional building elements and their conversion rules by modifying Algorithm 2. Therefore, in principle, SiteBuilder can support diverse and more complex atrium scenes according to specific user needs.

3.4 Geometry-Appearance Encoder

As shown in Figure 4, the Geometry-Appearance Encoder uses parallel depth and semantic branches to encode rendered controls into complementary conditioning signals. The two branches are architecturally identical and token-aligned with the DiT blocks in the video diffusion backbone, and they share the same video encoder as the diffusion model to keep latent representations consistent. Following this design, we first sample a camera trajectory $\mathcal{T} = \{\pi_t\}_{t=1}^{T}$ to cover unseen regions while revisiting seen regions for global consistency. For each specific pose π_t, we render a depth map $\mathbf{D}_t$ and a semantic map $\mathbf{S}_t$ from the bootstrap mesh $\mathcal{B}$. And when $\mathcal{G}_0$ is available, we also render a key frame $\mathbf{K}_t$ at the same pose. We then use the depth sequence $\{\mathbf{D}_t\}_{t=1}^{T}$ and the semantic sequence $\{\mathbf{S}_t\}_{t=1}^{T}$ as dual controls. Depth provides geometric layout, while each semantic map is a dense label in image space, obtained by projecting bootstrap-mesh semantics onto the current view. We use a fixed label encoding across time so that the same class keeps the same code in all frames. The Geometry-Appearance Encoder

Algorithm 1 Sitemap to Bootstrap Mesh

Require: Sitemap $\mathcal{M} = \{\mathcal{F}^{(l)}\}_{l=1}^{L}$, optional incomplete model $\mathcal{G}_0$, default floor height $\bar{h}$
Ensure: Bootstrap mesh $\mathcal{B}$

if $\mathcal{G}_0 = \varnothing$ **then**
  $h \leftarrow \bar{h}$
else
  $h \leftarrow$ EstimateFloorHeight$(\mathcal{M}, \mathcal{G}_0)$ ▷ Alg. 3
end if
$\mathcal{B} \leftarrow \varnothing$
for $l = 1$ to L **do**
  $z_l \leftarrow (l-1)h$
  $\mathcal{B}^{(l)} \leftarrow \varnothing$
  for each $(\mathbf{p}_i^{(l)}, y_i^{(l)}) \in \mathcal{F}^{(l)}$ **do**
    $\mathcal{Q}_i^{(l)} \leftarrow$ NaiveElementMesh$(\mathbf{p}_i^{(l)}, y_i^{(l)}, z_l, h)$ ▷ Alg. 2
    $\mathcal{B}^{(l)} \leftarrow \mathcal{B}^{(l)} \cup \mathcal{Q}_i^{(l)}$
  end for
  $\mathcal{B} \leftarrow \mathcal{B} \cup \mathcal{B}^{(l)}$
end for
return $\mathcal{B}$

Algorithm 2 Naive Element Mesh

Require: Element geometry $\mathbf{p}$, element type y, floor base height z, floor height h
Ensure: Naive planar primitive(s) $\mathcal{Q}$

if y is wall **then**
  Build a vertical plane from $\mathbf{p}$ with height h
else if y is floor **then**
  Build a horizontal plane at height z, allowing interior holes
else if y is ceiling **then**
  Build a horizontal plane at height $z + h$, allowing interior holes
else if y is handrail **then**
  Build a vertical plane from $\mathbf{p}$ with height $h/2$
else if y is stair **then**
  Build a sloped plane between two adjacent floors
else
  $\mathcal{Q} \leftarrow \varnothing$
end if
return $\mathcal{Q}$

Algorithm 3 Coarse-to-Fine Floor Height Estimation

Require: Sitemap $\mathcal{M}$, incomplete model $\mathcal{G}_0$, search window $[h_{\min}, h_{\max}]$, rounds R, samples per round K, shrink ratio $\rho \in (0, 1)$
Ensure: Estimated floor height h'

Extract Gaussian centers $C = \{\mathbf{c}_n\}_{n=1}^{N}$ from $\mathcal{G}_0$
for $r = 1$ to R **do**
  Sample K candidate heights $\mathcal{H}_r = \{h_k\}_{k=1}^{K}$ uniformly from $[h_{\min}, h_{\max}]$
  for each $h_k \in \mathcal{H}_r$ **do**
    Build provisional mesh $\mathcal{B}(h_k)$ from $\mathcal{M}$ using floor height h_k
    $E(h_k) \leftarrow \frac{1}{N}\sum_{n=1}^{N} \min_{\mathbf{x}\in\mathrm{surf}(\mathcal{B}(h_k))} \|\mathbf{c}_n - \mathbf{x}\|_2$
  end for
  $h' \leftarrow \arg\min_{h_k\in\mathcal{H}_r} E(h_k)$
  $\Delta \leftarrow \frac{\rho}{2}(h_{\max} - h_{\min})$
  $h_{\min} \leftarrow h' - \Delta,\ h_{\max} \leftarrow h' + \Delta$
end for
return h'

Table 1: Comparison on structural correctness between SiteBuilder and six representative structure-guided cellular methods. Both depth error (view-dependent) and chamfer distance (view-agnostic) are reported. SiteBuilder achieves best performance on all metrics.

Method	Mean Depth AbsRel (%)	Max Depth AbsRel (%)	Mean Rel SurfCD (%)
Plan2Scene [52]	14.3	19.2	18.1
Ctrl-Room [44]	14.7	17.0	19.0
HouseCrafter [34]	13.5	18.5	15.4
SceneCraft [58]	13.1	19.1	16.3
Layout2Scene [5]	12.1	18.8	16.7
SpatialGen [9]	11.9	19.3	15.8
SiteBuilder (Ours)	**8.2**	**15.3**	**10.7**

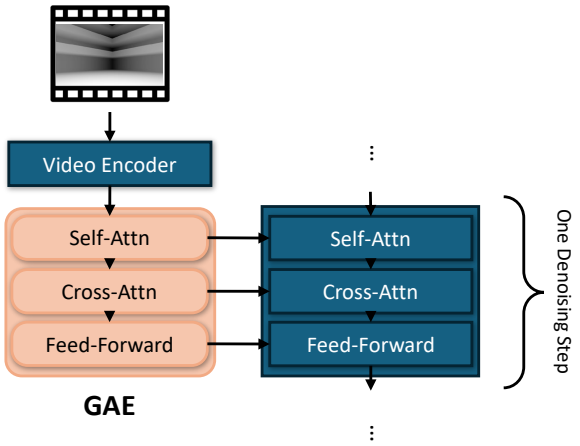


Figure 4: The depth branch and semantic branch of Geometry-Appearance Encoder (GAE) are the same and are aligned with the DiT in video diffusion network. The video encoder is shared between the diffusion network and GAE. Conditioning cues are injected to each layer in DiT in each denoising step.

(GAE) maps them to two conditioning streams

$$C^g = \mathcal{E}_g\left(\{\mathrm{D}_t\}_{t=1}^{T}\right), \qquad C^a = \mathcal{E}_a\left(\{\mathrm{S}_t\}_{t=1}^{T}\right), \tag{6}$$

where $\mathcal{E}^g$ and $\mathcal{E}^a$ are the geometry and appearance branch of GAE respectively, C^g and C^a are the computed geometry and appearance condition. Both branches use the same conditioning interface and are implemented as lightweight adapter updates on a frozen video generator, making the module compatible with different backbone choices.

Both controls are injected as reference-video latents into every DiT layer at each denoising step. Let C^0 denote optional keyframe conditions rendered from $\mathcal{G}_0$. With branch-wise conditioning strengths $\lambda_g, \lambda_a \in [0, 1]$, the conditioned video generator is

$$\hat{\mathbf{I}}_{1:T} = \mathcal{V}\left(\lambda_g C^g, \lambda_a C^a, C^0\right). \tag{7}$$

where $\mathcal{V}(\cdot)$ denotes the frozen video generative network.

During training, the depth branch can be initialized from an existing geometry controller, while the semantic branch is trained with aligned RGB-depth-semantic video tuples under the same recipe so that both branches remain architecturally consistent. The GAE module is optimized using the same video diffusion loss as the backbone video diffusion network. At inference, the two branches are fused additively to provide joint structural control. Depth mainly constrains geometry, semantics mainly constrains appearance.

4 Illustrative Experimental Results

In Section 4.1, we describe data preparation across real and synthetic atrium scenes. In Section 4.2, we present implementation details and training settings. In Section 4.3, we compare SiteBuilder with baseline methods quantitatively. In Section 4.4, we analyze the contribution of different Geometry-Appearance Encoder branches through ablation experiments. In Section 4.5, we provide qualitative visual comparisons, while Section 4.6 studies hyperparameter effects. In Section 4.7, we report user study results on structural faithfulness and visual consistency.

4.1 Data Preparation

We evaluate SiteBuilder on five atrium scenes with two real-world scenes and three synthetic ones. The sitemap is in geojson format [4], which commonly appears in any map and navigation applications and is easy to obtain. For the two real scenes, their sitemaps include the same element-level representation defined in our methodology (semantic 2D elements with geometric support and type labels, including walls, floors, ceilings, handrails, and stairs), and we manually align the ground-truth 3D models to the corresponding sitemaps so that horizontal position and scale are consistent in a shared coordinate frame. For the synthetic scene, we create the sitemap ourselves to mimic a realistic shopping-mall environment with multi-floor open spaces and cross-floor connectivity. Across all five scenes, we keep the unified sitemap format and the same preprocessing protocol to provide consistent conditioning inputs for bootstrap construction and geometry-appearance guided generation.

4.2 Implementation Details

We use the LTX-2 video diffusion checkpoints [15] as the generative network in SiteBuilder. The core idea of SiteBuilder is general and

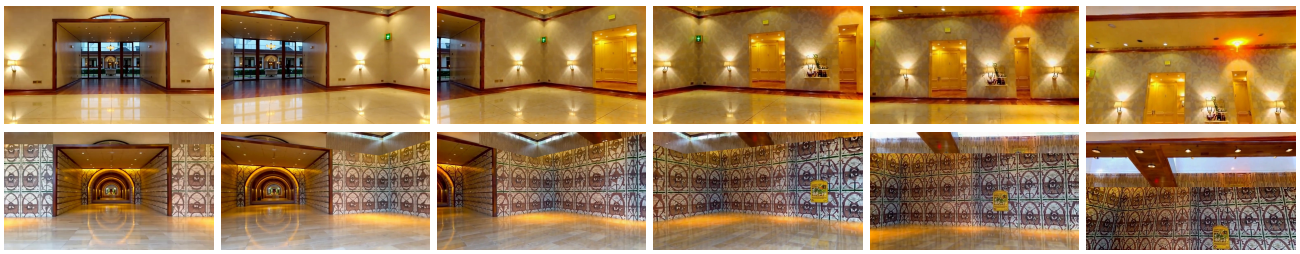


Figure 5: Qualitative comparison between SiteBuilder (above) and SpatialGen (below). The leftmost view shows clearly the incorrect structure and inconsistent appearance due to the individual generation in previous cellular methods.

compatible with other video diffusion backbones through the same conditioning interface. During training, we keep the video diffusion backbone frozen and only optimize the Geometry-Appearance Encoder (GAE) on real scene data using the AdamW optimizer and a cosine_with_restarts learning-rate scheduler. We set the geometry conditioning factor to $\lambda_g = 1.0$ and the appearance conditioning factor to $\lambda_a = 0.2$. For floor-height estimation, we set $R = 2$, $K = 10$, and shrink ratio $\rho = 0.5$. One trajectory is sampled for each floor which also covers the neighbor floors, the trajectory is then used to generate 160 video frames. For diffusion sampling, we use 50 denoising steps. The 3DGS framework here is gsplat [62]. We train the GAE on a NVIDIA A100 GPU for approximately 12 hours. The resolution of the final generated videos after upsampling is 1024×576. Video generation during the inference stage consumes 53GB VRAM and takes 4 minutes per 160 frames. 3DGS reconstruction consumes 35GB VRAM and takes about 4 hours.

4.3 Performance Comparison

Table 1 summarizes our quantitative comparison. As the first metric, we report the absolute relative error on depth (%), computed from the rendered depth map of the bootstrap mesh and the rendered depth map of the completed scene model under the same camera poses. We report both the mean and the max values. As the last metric, we report Mean Relative Surface Chamfer Distance (Mean Rel SurfCD, %). This metric is very similar to what we define in Eq. 4 except that the distance is further divided by the floor height. The intuition is that floor height is usually how humans perceive the internal scale of a building. Therefore, by this normalization, we can better estimate how people perceive the structural error of the completed scene.

We compare with six other representative structure-guided scene completion methods. Since all of them are designed for cellular-layout indoor scenes, we split each atrium scene into multiple subregions so that these approaches can function normally. For approaches that do not directly involve a point cloud or 3DGS, we further extract a point cloud from their intermediate results to report the metrics.

The results clearly show the superior performance of SiteBuilder across all metrics. SiteBuilder achieves the lowest Mean Depth AbsRel (8.2%), substantially better than the best baseline (11.9% from SpatialGen), and the lowest Max Depth AbsRel (15.3%), improving over the strongest baseline (17.0% from Ctrl-Room). On Mean Rel SurfCD, SiteBuilder reaches 10.7%, significantly lower than all competing methods (the best baseline is 15.4% from HouseCrafter), indicating better structural correctness. The results prove quantitatively that SiteBuilder is the best framework for physical atrium scene completion. We attribute these gains to our key design choices of bootstrap mesh and the GAE module. Together with trajectory-based generation, SiteBuilder achieves global consistency over large open regions that are difficult for cellular decomposition methods.

Table 2: Ablation study by removing either GAE branch. The depth branch proves to be essential for structural correctness in the completed scene model.

Method	Mean Rel SurfCD
SiteBuilder w/o Depth	54.1
SiteBuilder w/o Semantic	15.8
SiteBuilder (Ours)	10.7

4.4 Ablation Study

To understand how much the bootstrap mesh and the Geometry-Appearance Encoder (GAE) contribute to structural correctness in atrium scenes, we conduct an ablation study by comparing full SiteBuilder with variants that remove one branch in GAE. Specifically, we evaluate removing the depth branch and removing the semantic branch, and compare both settings against the full model. The ablation settings and corresponding quantitative results are summarized in Table 2.

Table 2 shows that both branches in GAE are important, while the depth branch is the primary contributor to structural correctness. Removing the depth branch causes a dramatic degradation of Mean Rel SurfCD from 10.7 to 54.1, indicating that geometric control is indispensable for maintaining correct global layout in atrium scenes. Removing the semantic branch also hurts performance, increasing Mean Rel SurfCD from 10.7 to 15.8, which verifies that semantic cues provide complementary constraints that stabilize appearance-structure alignment. The full model achieves the best result because depth and semantic conditions are jointly injected, matching our design goal of unified geometry-appearance guidance.

4.5 Qualitative Evaluation

We visualize completed atrium scenes for qualitative evaluation. Figure 5 shows the qualitative comparison between SiteBuilder and SpatialGen, and Figure 6 further compares our renderings with

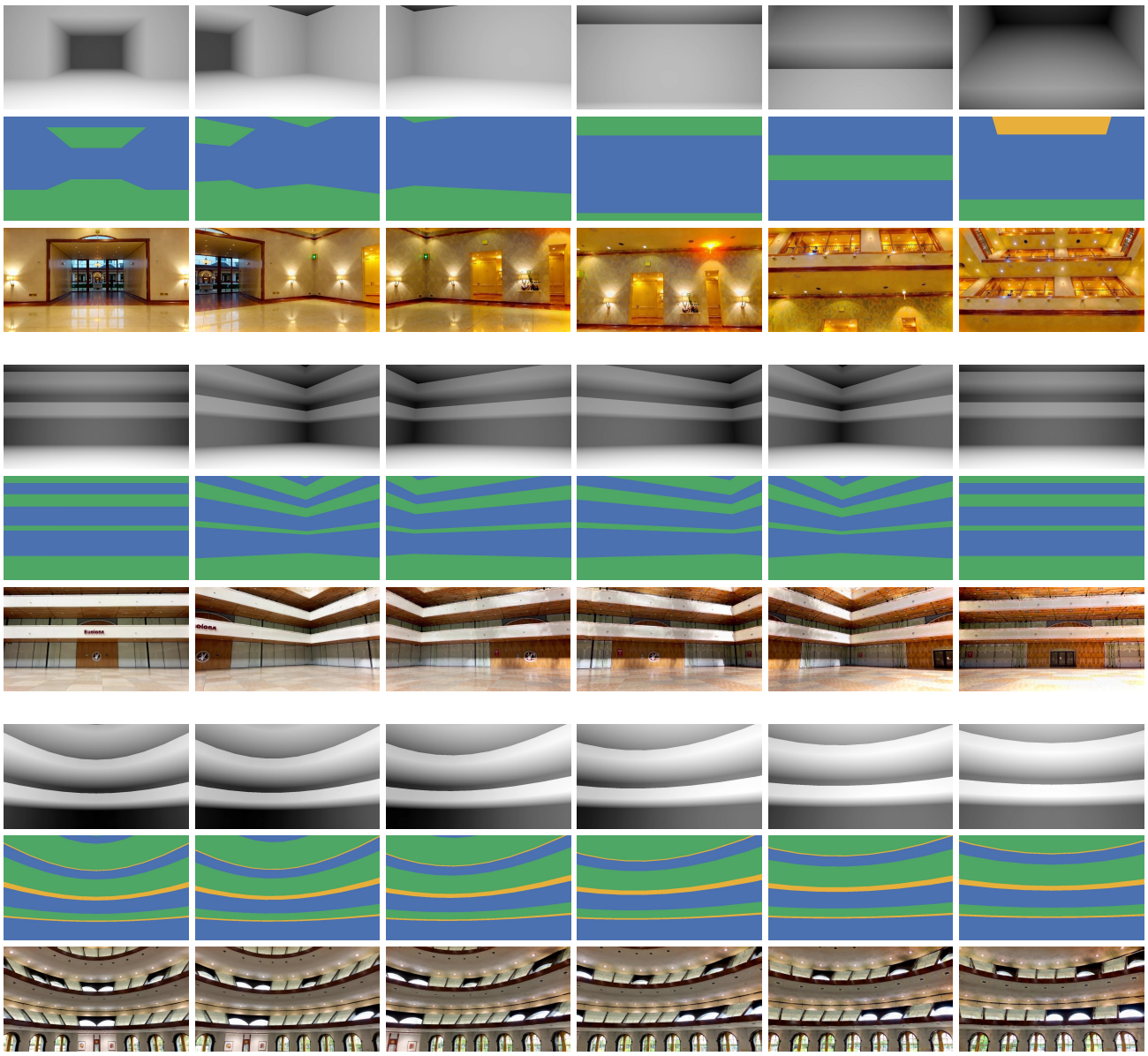


Figure 6: Qualitative analysis of SiteBuilder with depth and semantic references. Each scene has, from top to bottom, depth, semantic, and rendering from complete scene. The generated scenes well align with the depth and semantic cues extracted from the bootstrap mesh. SiteBuilder achieves both structural correctness and global visual consistency.

the corresponding depth and semantic references for structural alignment.

As shown in Figure 5, SpatialGen, as a representative method designed for cellular-layout buildings, does not handle atrium scenes well enough. Because it applies an independent generation process to each subregion, it fails to maintain global consistency. This is especially visible in the leftmost view, where there is a clear inconsistency between the generated first floor and second floor structures. In contrast, SiteBuilder adopts integrated generation over the entire scene by first constructing a whole-scene bootstrap mesh. This global structural conditioning leads to substantially better cross-region and cross-floor consistency in the completed results.

Figure 6 further shows that our generated scenes align well with the extracted depth and semantic conditioning sequences. The results maintain high quality in both geometry and appearance across different scenes with different building structures.

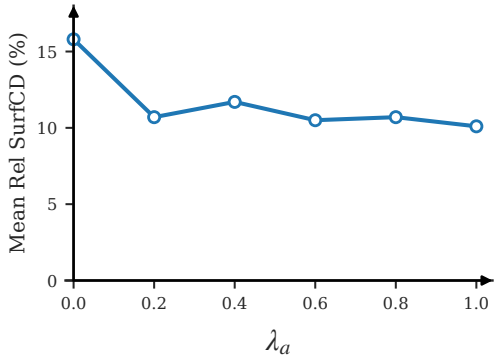


Figure 7: Hyperparameter analysis of appearance-conditioning factor λ_a. No significant improvement is observed when $\lambda_a > 0.2$.

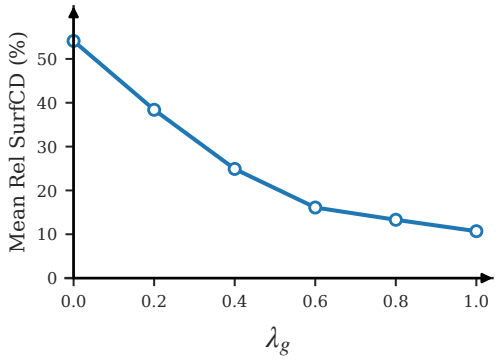


Figure 8: Hyperparameter analysis of geometry-conditioning factor λ_g. A clear decreasing trend is observed.

4.6 Hyperparameter Analysis

We would like to find out how our hyperparameters λ_g and λ_a affect the structural correctness. We plot the mean relative surface chamfer distance with respect to these two factors in Figure 7 and Figure 8. The plots show different behaviors for the two conditioning factors. As λ_g increases, the structural error consistently decreases, which echoes our core finding that depth is the irreplaceable control for faithfully reflecting real building structure. In contrast, increasing λ_a does not stably improve structural correctness. We further observe that overly strong semantic conditioning reduces the diversity of completed scenes. We attribute this phenomenon to the relatively small semantic vocabulary in our setup. When semantic control is too strong, it can over-constrain the video generator and hurt generation quality. Based on this trade-off, we set $\lambda_a = 0.2$ in our final framework as a balance between structural faithfulness and appearance quality.

Table 3: User study results for structural faithfulness and visual consistency. SiteBuilder obtains the majority of the positive votes in terms of both structural correctness and visual consistency. Although a few participants remain neutral, none shows an explicit preference on SpatialGen.

Vote	Structural	Visual
SiteBuilder (Ours)	17	18
Neutral	3	2
SpatialGen	0	0

4.7 User Study

We also conduct a user study to evaluate the structural correctness and visual consistency of completed atrium scenes. We recruit 20 participants, including 10 university students and 10 non-professional participants. In each trial, a participant is given two rendered images from the same camera pose, one generated by SiteBuilder and one generated by SpatialGen, together with the corresponding sitemap and bootstrap mesh as structural references. The participant is then asked to answer two questions: (1) which result has a more faithful structure, and (2) which result appears more visually consistent. For each question, participants are allowed to stay neutral, so each question has three options: SiteBuilder, Neutral, and SpatialGen.

As shown in Table 3, the user study results show a clear preference for SiteBuilder over SpatialGen in both criteria. For structural faithfulness, 17 out of 20 participants select SiteBuilder, while the remaining 3 choose Neutral and none prefer SpatialGen. For visual consistency, 18 out of 20 participants favor SiteBuilder, with 2 Neutral votes and again no votes for SpatialGen. This user study, together with previous qualitative and quantitative results, indicate that our sitemap-grounded conditioning improves both structure-level realism and global visual coherence from a human-perception perspective.

5 Conclusion

We presented SiteBuilder, a sitemap-grounded framework for 3D atrium scene completion in large multi-storey indoor spaces. By converting a sitemap into a bootstrap mesh and injecting depth/semantic controls through the Geometry-Appearance Encoder (GAE), SiteBuilder jointly enforces structural correctness and visual consistency during generation.

Experiments show that SiteBuilder achieves the best structural performance (8.2% Mean Depth AbsRel and 10.7% Mean Rel SurfCD), with ablations confirming depth as the key control and user studies showing clear preference over SpatialGen. A current limitation is the required pre-alignment between incomplete 3D input and sitemap coordinates; future work includes robust sitemap-guided registration and broader semantic/data coverage for better generalization.

Acknowledgments

This work was supported, in part, by Research Grants Council Collaborative Research Fund (under grant number C1045-23G), and RGC-General Research Fund (under grant number 16201625).

References

[1] Sherwin Bahmani, Jeong Joon Park, Despoina Paschalidou, Xingguang Yan, Gordon Wetzstein, Leonidas Guibas, and Andrea Tagliasacchi. 2023. Cc3d: Layout-conditioned generation of compositional 3d scenes. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 7171–7181.

[2] Miguel Angel Bautista, Pengsheng Guo, Samira Abnar, Walter Talbott, Alexander Toshev, Zhuoyuan Chen, Laurent Dinh, Shuangfei Zhai, Hanlin Goh, Daniel Ulbricht, et al. 2022. Gaudi: A neural architect for immersive 3d scene generation. *Advances in Neural Information Processing Systems* 35 (2022), 25102–25116.

[3] Andreas Blattmann, Tim Dockhorn, Sumith Kulal, Daniel Mendelevitch, Maciej Kilian, Dominik Lorenz, Yam Levi, Zion English, Vikram Voleti, Adam Letts, et al. 2023. Stable video diffusion: Scaling latent video diffusion models to large datasets. *arXiv preprint arXiv:2311.15127* (2023).

[4] Howard Butler, Martin Daly, Allan Doyle, Sean Gillies, Stefan Hagen, and Tim Schaub. 2016. *The geojson format*. Technical Report.

[5] Minglin Chen, Longguang Wang, Sheng Ao, Ye Zhang, Kai Xu, and Yulan Guo. 2025. Layout2Scene: 3D semantic layout guided scene generation via geometry and appearance diffusion priors. *arXiv preprint arXiv:2501.02519* (2025).

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

[7] Manuel Ladron de Guevara, Jinmo Rhee, Ardavan Bidgoli, Vaidas Razgaitis, and Michael Bergin. 2025. Tokenizing Buildings: A Transformer for Layout Synthesis. *arXiv preprint arXiv:2512.04832* (2025).

[8] Chuan Fang, Yuan Dong, Kunming Luo, Xiaotao Hu, Rakesh Shrestha, and Ping Tan. 2025. Ctrl-room: Controllable text-to-3d room meshes generation with layout constraints. In *2025 International Conference on 3D Vision (3DV)*. IEEE, 692–701.

[9] Chuan Fang, Heng Li, Yixun Liang, Jia Zheng, Yongsen Mao, Yuan Liu, Rui Tang, Zihan Zhou, and Ping Tan. 2025. Spatialgen: Layout-guided 3d indoor scene generation. *arXiv preprint arXiv:2509.14981* 3 (2025).

[10] Shaoheng Fang, Chaohui Yu, Fan Wang, and Qixing Huang. 2025. MVRoom: Controllable 3D Indoor Scene Generation with Multi-View Diffusion Models. *arXiv preprint arXiv:2512.04248* (2025).

[11] Weitao Feng, Hang Zhou, Jing Liao, Li Cheng, and Wenbo Zhou. 2025. CasaGPT: cuboid arrangement and scene assembly for interior design. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 29173–29182.

[12] Weixi Feng, Wanrong Zhu, Tsu-jui Fu, Varun Jampani, Arjun Akula, Xuehai He, Sugato Basu, Xin Eric Wang, and William Yang Wang. 2023. Layoutgpt: Compositional visual planning and generation with large language models. *Advances in Neural Information Processing Systems* 36 (2023), 18225–18250.

[13] Rao Fu, Zehao Wen, Zichen Liu, and Srinath Sridhar. 2024. Anyhome: Open-vocabulary generation of structured and textured 3d homes. In *European Conference on Computer Vision*. Springer, 52–70.

[14] Zeqi Gu, Yin Cui, Zhaoshuo Li, Fangyin Wei, Yunhao Ge, Jinwei Gu, Ming-Yu Liu, Abe Davis, and Yifan Ding. 2025. Artiscene: Language-driven artistic 3d scene generation through image intermediary. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 2891–2901.

[15] Yoav HaCohen, Benny Brazowski, Nisan Chiprut, Yaki Bitterman, Andrew Kvochko, Avishai Berkowitz, Daniel Shalem, Daphna Lifschitz, Dudu Moshe, Eitan Porat, et al. 2026. LTX-2: Efficient Joint Audio-Visual Foundation Model. *arXiv preprint arXiv:2601.03233* (2026).

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

[17] Joy Hsu, Emily Jin, Jiajun Wu, and Niloy J Mitra. 2025. From Programs to Poses: Factored Real-World Scene Generation via Learned Program Libraries. *arXiv preprint arXiv:2510.10292* (2025).

[18] Hanzhe Hu, Zhizhuo Zhou, Varun Jampani, and Shubham Tulsiani. 2024. Mvdfusion: Single-view 3d via depth-consistent multi-view generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9698–9707.

[19] Jiaxin Huang, Yuanbo Yang, Bangbang Yang, Lin Ma, Yuewen Ma, and Yiyi Liao. 2026. Gen3R: 3D Scene Generation Meets Feed-Forward Reconstruction. *arXiv preprint arXiv:2601.04090* (2026).

[20] Zehuan Huang, Hao Wen, Junting Dong, Yaohui Wang, Yangguang Li, Xinyuan Chen, Yan-Pei Cao, Ding Liang, Yu Qiao, Bo Dai, et al. 2024. Epidiff: Enhancing multi-view synthesis via localized epipolar-constrained diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9784–9794.

[21] Xiaoliang Ju, Zhaoyang Huang, Yijin Li, Guofeng Zhang, Yu Qiao, and Hongsheng Li. 2024. Diffindscene: Diffusion-based high-quality 3d indoor scene generation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4526–4535.

[22] Yash Kant, Aliaksandr Siarohin, Michael Vasilkovsky, Riza Alp Guler, Jian Ren, Sergey Tulyakov, and Igor Gilitschenski. 2023. invs: Repurposing diffusion inpainters for novel view synthesis. In *SIGGRAPH Asia 2023 Conference Papers*. 1–12.

[23] Yash Kant, Aliaksandr Siarohin, Ziyi Wu, Michael Vasilkovsky, Guocheng Qian, Jian Ren, Riza Alp Guler, Bernard Ghanem, Sergey Tulyakov, and Igor Gilitschenski. 2024. Spad: Spatially aware multi-view diffusers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10026–10038.

[24] Bernhard Kerbl, Georgios Kopanas, Thomas Leimkühler, George Drettakis, et al. 2023. 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.* 42, 4 (2023), 139–1.

[25] Xin Kong, Shikun Liu, Xiaoyang Lyu, Marwan Taher, Xiaojuan Qi, and Andrew J Davison. 2024. Eschernet: A generative model for scalable view synthesis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 9503–9513.

[26] Chen-Hsuan Lin, Jun Gao, Luming Tang, Towaki Takikawa, Xiaohui Zeng, Xun Huang, Karsten Kreis, Sanja Fidler, Ming-Yu Liu, and Tsung-Yi Lin. 2023. Magic3d: High-resolution text-to-3d content creation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 300–309.

[27] Minghua Liu, Ruoxi Shi, Linghao Chen, Zhuoyang Zhang, Chao Xu, Xinyue Wei, Hansheng Chen, Chong Zeng, Jiayuan Gu, and Hao Su. 2024. One-2-3-45++: Fast single image to 3d objects with consistent multi-view generation and 3d diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 10072–10083.

[28] Minghua Liu, Chao Xu, Haian Jin, Linghao Chen, Mukund Varma T, Zexiang Xu, and Hao Su. 2023. One-2-3-45: Any single image to 3d mesh in 45 seconds without per-shape optimization. *Advances in Neural Information Processing Systems* 36 (2023), 22226–22246.

[29] Ruoshi Liu, Rundi Wu, Basile Van Hoorick, Pavel Tokmakov, Sergey Zakharov, and Carl Vondrick. 2023. Zero-1-to-3: Zero-shot one image to 3d object. In *Proceedings of the IEEE/CVF international conference on computer vision*. 9298–9309.

[30] Yuan Liu, Cheng Lin, Zijiao Zeng, Xiaoxiao Long, Lingjie Liu, Taku Komura, and Wenping Wang. 2023. Syncdreamer: Generating multiview-consistent images from a single-view image. *arXiv preprint arXiv:2309.03453* (2023).

[31] Xiaoxiao Long, Yuan-Chen Guo, Cheng Lin, Yuan Liu, Zhiyang Dou, Lingjie Liu, Yuexin Ma, Song-Hai Zhang, Marc Habermann, Christian Theobalt, et al. 2024. Wonder3d: Single image to 3d using cross-domain diffusion. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9970–9980.

[32] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. 2021. Nerf: Representing scenes as neural radiance fields for view synthesis. *Commun. ACM* 65, 1 (2021), 99–106.

[33] Ahmed Mostafa, Omar Amir, Ali M Mohamed, and Marwa O Enany. 2025. Fine-Tuning Stable Diffusion for Generating 2D Floor PlansUsing Prompt Templates. *Machine Graphics and Vision* 34 (2025).

[34] Hieu T Nguyen, Yiwen Chen, Vikram Voleti, Varun Jampani, and Huaizu Jiang. 2024. Housecrafter: Lifting floorplans to 3d scenes with 2d diffusion model. *arXiv preprint arXiv:2406.20077* (2024).

[35] Yinyu Nie, Angela Dai, Xiaoguang Han, and Matthias Nießner. 2023. Learning 3d scene priors with 2d supervision. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 792–802.

[36] Hao Ouyang, Kathryn Heal, Stephen Lombardi, and Tiancheng Sun. 2023. Text2immersion: Generative immersive scene with 3d gaussians. *arXiv preprint arXiv:2312.09242* (2023).

[37] Despoina Paschalidou, Amlan Kar, Maria Shugrina, Karsten Kreis, Andreas Geiger, and Sanja Fidler. 2021. Atiss: Autoregressive transformers for indoor scene synthesis. *Advances in neural information processing systems* 34 (2021), 12013–12026.

[38] Hao Phung and Hadar Averbuch-Elor. 2026. Raster2Seq: Polygon Sequence Generation for Floorplan Reconstruction. *arXiv preprint arXiv:2602.09016* (2026).

[39] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. 2022. Dreamfusion: Text-to-3d using 2d diffusion. *arXiv preprint arXiv:2209.14988* (2022).

[40] Xuanchi Ren, Jiahui Huang, Xiaohui Zeng, Ken Museth, Sanja Fidler, and Francis Williams. 2024. Xcube: Large-scale 3d generative modeling using sparse voxel hierarchies. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4209–4219.

[41] Chris Rockwell, David F Fouhey, and Justin Johnson. 2021. Pixelsynth: Generating a 3d-consistent experience from a single image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 14104–14113.

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

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

[44] Jonas Schult, Sam Tsai, Lukas Höllein, Bichen Wu, Jialiang Wang, Chih-Yao Ma, Kunpeng Li, Xiaofang Wang, Felix Wimbauer, Zijian He, et al. 2024. Controlroom3d: Room generation using semantic proxy rooms. In *Proceedings of the*

IEEE/CVF Conference on Computer Vision and Pattern Recognition. 6201–6210.

[45] Mohammad Amin Shabani, Sepidehsadat Hosseini, and Yasutaka Furukawa. 2023. Housediffusion: Vector floorplan generation via a diffusion model with discrete and continuous denoising. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 5466–5475.

[46] Ruoxi Shi, Hansheng Chen, Zhuoyang Zhang, Minghua Liu, Chao Xu, Xinyue Wei, Linghao Chen, Chong Zeng, and Hao Su. 2023. Zero123++: a single image to consistent multi-view diffusion base model. *arXiv preprint arXiv:2310.15110* (2023).

[47] Yichun Shi, Peng Wang, Jianglong Ye, Mai Long, Kejie Li, and Xiao Yang. 2023. Mvdream: Multi-view diffusion for 3d generation. *arXiv preprint arXiv:2308.16512* (2023).

[48] Jaidev Shriram, Alex Trevithick, Lingjie Liu, and Ravi Ramamoorthi. 2024. Realmdreamer: Text-driven 3d scene generation with inpainting and depth diffusion. *arXiv preprint arXiv:2404.07199* (2024).

[49] Liangchen Song, Liangliang Cao, Hongyu Xu, Kai Kang, Feng Tang, Junsong Yuan, and Yang Zhao. 2023. Roomdreamer: Text-driven 3d indoor scene synthesis with coherent geometry and texture. *arXiv preprint arXiv:2305.11337* (2023).

[50] Stanislaw Szymanowicz, Christian Rupprecht, and Andrea Vedaldi. 2023. Viewset diffusion:(0-) image-conditioned 3d generative models from 2d data. In *Proceedings of the IEEE/CVF international conference on computer vision*. 8863–8873.

[51] Jiapeng Tang, Yinyu Nie, Lev Markhasin, Angela Dai, Justus Thies, and Matthias Nießner. 2024. Diffuscene: Denoising diffusion models for generative indoor scene synthesis. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 20507–20518.

[52] Madhawa Vidanapathirana, Qirui Wu, Yasutaka Furukawa, Angel X Chang, and Manolis Savva. 2021. Plan2scene: Converting floorplans to 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 10733–10742.

[53] Vikram Voleti, Chun-Han Yao, Mark Boss, Adam Letts, David Pankratz, Dmitry Tochilkin, Christian Laforte, Robin Rombach, and Varun Jampani. 2024. Sv3d: Novel multi-view synthesis and 3d generation from a single image using latent video diffusion. In *European Conference on Computer Vision*. Springer, 439–457.

[54] Can Wang, Hongliang Zhong, Menglei Chai, Mingming He, Dongdong Chen, and Jing Liao. 2025. Chat2Layout: Interactive 3D furniture layout with a multimodal LLM. *IEEE transactions on visualization and computer graphics* (2025).

[55] Haohan Weng, Tianyu Yang, Jianan Wang, Yu Li, Tong Zhang, CL Chen, and Lei Zhang. 2023. Consistent123: Improve consistency for one image to 3d object synthesis. *arXiv preprint arXiv:2310.08092* (2023).

[56] Zhennan Wu, Yang Li, Han Yan, Taizhang Shang, Weixuan Sun, Senbo Wang, Ruikai Cui, Weizhe Liu, Hiroyuki Sato, Hongdong Li, et al. 2024. Blockfusion: Expandable 3d scene generation using latent tri-plane extrapolation. *ACM Transactions on Graphics (ToG)* 43, 4 (2024), 1–17.

[57] Jiayu Yang, Ziang Cheng, Yunfei Duan, Pan Ji, and Hongdong Li. 2024. Consistnet: Enforcing 3d consistency for multi-view images diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7079–7088.

[58] Xiuyu Yang, Yunze Man, Jun-Kun Chen, and Yu-Xiong Wang. 2024. SceneCraft: Layout-guided 3D scene generation. *Advances in Neural Information Processing Systems* 37 (2024), 82060–82084.

[59] Yixuan Yang, Junru Lu, Zixiang Zhao, Zhen Luo, James JQ Yu, Victor Sanchez, and Feng Zheng. 2024. Llplace: The 3d indoor scene layout generation and editing via large language model. *arXiv preprint arXiv:2406.03866* (2024).

[60] Yue Yang, Fan-Yun Sun, Luca Weihs, Eli VanderBilt, Alvaro Herrasti, Winson Han, Jiajun Wu, Nick Haber, Ranjay Krishna, Lingjie Liu, et al. 2024. Holodeck: Language guided generation of 3d embodied ai environments. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16227–16237.

[61] Jianglong Ye, Peng Wang, Kejie Li, Yichun Shi, and Heng Wang. 2024. Consistent-1-to-3: Consistent image to 3d view synthesis via geometry-aware diffusion models. In *2024 International Conference on 3D Vision (3DV)*. IEEE, 664–674.

[62] Vickie Ye, Ruilong Li, Justin Kerr, Matias Turkulainen, Brent Yi, Zhuoyang Pan, Otto Seiskari, Jianbo Ye, Jeffrey Hu, Matthew Tancik, and Angjoo Kanazawa. 2025. gsplat: An open-source library for Gaussian splatting. *Journal of Machine Learning Research* 26, 34 (2025), 1–17.

[63] Taoran Yi, Jiemin Fang, Junjie Wang, Guanjun Wu, Lingxi Xie, Xiaopeng Zhang, Wenyu Liu, Qi Tian, and Xinggang Wang. 2024. Gaussiandreamer: Fast generation from text to 3d gaussians by bridging 2d and 3d diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 6796–6807.

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

[65] Guangyao Zhai, Evin Pınar Örnek, Dave Zhenyu Chen, Ruotong Liao, Yan Di, Nassir Navab, Federico Tombari, and Benjamin Busam. 2024. Echoscene: Indoor scene generation via information echo over scene graph diffusion. In *European Conference on Computer Vision*. Springer, 167–184.

[66] Jingbo Zhang, Xiaoyu Li, Ziyu Wan, Can Wang, and Jing Liao. 2024. Text2nerf: Text-driven 3d scene generation with neural radiance fields. *IEEE Transactions on Visualization and Computer Graphics* 30, 12 (2024), 7749–7762.

[67] Guangcong Zheng, Xianpan Zhou, Xuewei Li, Zhongang Qi, Ying Shan, and Xi Li. 2023. Layoutdiffusion: Controllable diffusion model for layout-to-image generation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 22490–22499.

[68] Jensen Zhou, Hang Gao, Vikram Voleti, Aaryaman Vasishta, Chun-Han Yao, Mark Boss, Philip Torr, Christian Rupprecht, and Varun Jampani. 2025. Stable virtual camera: Generative view synthesis with diffusion models. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 12405–12414.