

MeSS: City Mesh-Guided Outdoor Scene Generation with Cross-View Consistent Diffusion

Xuyang Chen^{1,2*}, Zhijun Zhai³, Kaixuan Zhou¹, Zengmao Wang³, Jianan He¹,
Dong Wang¹, Yanfeng Zhang¹, Mingwei Sun^{1,3}, Xuqin Wang^{1,2}, Rüdiger
Westermann², Tao Wu¹, Konrad Schindler⁴, and Liqiu Meng²

¹ Huawei Hilbert Research Center

² Technical University of Munich, Munich, Germany

xuyang.chen@tum.de

³ Wuhan University, Wuhan, China

⁴ ETH Zurich, Zurich, Switzerland

Abstract. Mesh models have become increasingly accessible for numerous cities; however, the lack of realistic textures restricts their application in virtual urban navigation and autonomous driving. Texturing them with generative models remains difficult: existing video-diffusion and outpainting-based scene generators drift in appearance over long camera trajectories and cannot guarantee pixel-level alignment with the given city geometry. This paper proposes MeSS (Mesh-based Scene Synthesis), which converts a textureless city mesh into a high-quality, style-consistent 3D Gaussian Splatting (3DGS) scene. Our key insight is a control-distribution match: we train image ControlNets directly on mesh-rendered depth, normal, and semantic priors, so the control signals stay in-distribution for any deployed city mesh and yield tight geometry–appearance alignment. Building on this, three cross-view consistency mechanisms counter appearance drift: Cascaded Outpainting CtrlNets generate geometrically consistent sparse key views; Appearance Guided Inpainting (AGInpaint) densifies the scene with intermediate views; and Global Consistency Alignment (GCAAlign) removes global inconsistencies such as exposure shifts. Concurrently with generation, the 3DGS scene is reconstructed by initializing Gaussian surfels on the mesh surface. On a city-scale benchmark from the UE5 City Sample Project, MeSS reduces cross-view LPIPS by 31% and KID by 36% relative to the best competing method on each metric, achieving an FID of 28.17, and it transfers zero-shot to a real LoD3 city mesh. Once synthesized, the scene can be rendered in diverse styles through relighting and style transfer.

Keywords: Scene Generation · 3D Gaussian Splatting

* Corresponding Author

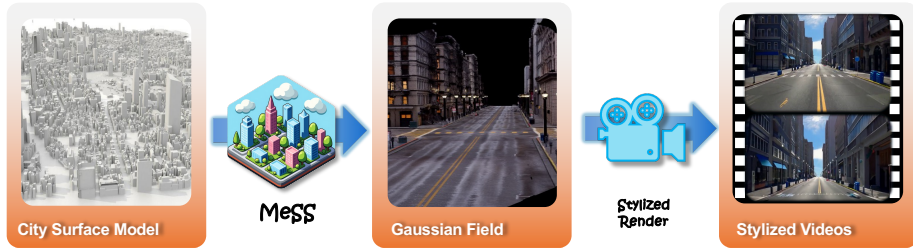


Fig. 1. Starting from textureless urban meshes, our MeSS synthesizes high-quality Gaussian Splatting scenes with realistic appearance. After synthesis, these Gaussian scenes can be further rendered into stylized videos.

1 Introduction

Detailed city-scale mesh models are becoming widely available. Examples include 3DCityDB [52], Hamburger LOD3 [24], Ingolstadt LOD3 [42], and 3DBAG [38], typically at Level of Detail 3 (LoD3) for buildings. However, these models often lack high-quality textures, which hinders downstream applications such as VR-based urban navigation, where street-level users are close enough to notice visual flaws.

Generative models offer a scalable and customizable way to enrich the visual appearance of such mesh models. Street-view capture with survey vehicles is prohibitively costly and prone to occlusions and illumination variability; generation is cost-efficient and lets users tailor scene aesthetics to their needs.

Existing generative scene methods, however, struggle to respect the geometry of a given city model. Recent works generate a city as NeRF [36] or 3DGS [22] from city semantic & height maps [6,28,9,49,50,51], but being supervised in bird’s-eye view, their visual quality degrades significantly at street level. Other approaches [48,25,57,18] synthesize driving scenes with video diffusion models [3]; focusing on foreground traffic participants and road infrastructure, they hallucinate the background—particularly buildings—without structural constraints, so the results often misalign with the actual city layout. StreetScapes [11] conditions on simplified building geometries and map layouts, and Cosmos-Transfer1 [1] injects sparse lidar-derived depth; yet their imprecise geometric inputs and feed-forward conditional pipelines still fail to guarantee precise geometric alignment.

Approaches that operate directly on meshes or images come closer to our setting, yet still fall short at city scale. Texturing mesh models with 2D diffusion models [8,35,7,40] is limited to single objects or small scenes with low polygon counts. Perpetual view generation [53,54,47,10] progressively outpaints RGB images and depth maps from a single image using 2D diffusion models [41], but gradually drifts away from the initial appearance and cannot generate consistent dense views.

In summary, one must solve two main challenges to generate city-scale scenes from mesh models: (i) Generate visual results aligning with given city surface models. (ii) Maintain a consistent appearance across long range without drifting.

Requirement (i) motivates a *control-distribution match*. Our image CtrlNets are trained directly on the depth, semantic, and normal priors rendered from city meshes. At deployment, the control signals from any city mesh—including real LoD3 models—therefore remain in-distribution and yield tight pixel-level alignment between geometry and appearance. Generic video-diffusion controllers, by contrast, are conditioned on depth estimated from real video and degrade on clean mesh-rendered control.

To meet requirement (ii), we organize generation as a two-staged *warp-and-outpaint* framework with dedicated cross-view consistency mechanisms. In Stage I, **sparse key frames** are generated by two *Cascaded Outpainting CtrlNets* that consecutively outpaint each frame from its predecessor, suppressing appearance drift over long ranges; the key frames are meanwhile spread as Gaussian surfels on the mesh surface to guarantee geometric alignment. In Stage II, intermediate views rendered from the Gaussian field—commonly exhibiting blurry artifacts or silhouettes—are repaired with *Appearance Guided Inpainting* (AGInpaint). Finally, *Global Consistency Alignment* (GCAAlign) harmonizes remaining global discrepancies such as exposure shifts.

In summary, we make the following technical contributions:

- We introduce **Cascaded Outpainting CtrlNets** conditioned on preceding frames to generate key frames that stay consistent over long camera trajectories.
- We devise an **Appearance Guided Inpainting** method to inpaint the occluded areas in the intermediate dense views, guided by surrounding known regions.
- On a city-scale benchmark rendered from the UE5 City Sample Project [15], MeSS achieves state-of-the-art performance on 200-frame sequences spanning 200m camera paths, reducing cross-view LPIPS by 31% and KID by 36% relative to the best competing method on each metric with the lowest FID of 28.17, and transfers zero-shot to a real LoD3 city mesh (Fig. 5).

2 Related Work

2.1 Scene Generation from 2D Maps

One line of work generates entire 3D cities from 2D semantic and height maps. InfiniCity [28] decomposes 3D scene generation into 2D map generation, map-to-voxel lifting, and voxel texturing via neural rendering. SceneDreamer [9] represents unbounded 3D landscapes via a BEV layout with height and semantic fields. CityDreamer [49] and CityDreamer4D [51] refine this paradigm by decomposing rendering into specialized modules for distinct scene components, with CityDreamer4D further incorporating temporal consistency. Since these methods [50,49,51] are supervised largely in bird’s-eye view, they still struggle with

layout stability and high-fidelity rendering at street level. MeSS instead conditions on an existing city mesh, so the generated scene conforms to the given city layout down to street-level detail.

2.2 Perpetual View Generation

Perpetual view generation [29] synthesizes views along arbitrary camera trajectories from a single image via a render-and-refine pipeline. This principle has been extended through GANs [27] and diffusion models [5,17], with WonderWorld [53] enabling interactive 3D scene generation and VistaDream [47] enforcing multi-view geometry constraints during denoising. These approaches [10,54] all profit from monocular depth estimation [39,2,21], yet still exhibit rapid appearance drift in complex urban scenarios where outpainting error accumulates across occluded regions. Our Cascaded Outpainting CtrlNets adopt the same warp-and-outpaint principle but anchor every view to mesh-rendered controls, which suppresses this error accumulation.

2.3 Scene Synthesis from Video Generation

Video generators can synthesize approximately consistent views by conditioning on a single frame [55,16,44] or interpolating multiple viewpoints [30,55,44]. StreetScapes [11] addresses a task similar to ours with a two-frame AnimatedDiff-based [20] generator conditioned on rendered depth, height, and semantics, and Cosmos-Transfer1 [1] conditions video generation on sparse lidar-derived depth. However, training such models is expensive in data and compute, and their imprecise geometric control leads to temporal drift in long sequences. We instead build on image diffusion models whose controls are trained directly on mesh-rendered priors, keeping the control signals in-distribution at deployment without any video-model training.

2.4 Test-time Guidance during Diffusion Sampling

RePaint [31] re-imposes known pixels at each reverse step to guide inpainting, but in Latent Diffusion Models [41] the $8\times$ downsampling erases narrow mask boundaries. Recent works [53,46,43,23] instead back-propagate pixel-space losses through the latent at each denoising step, yet gradient-based guidance for RGB inpainting remains unexplored. Our AGInpaint fills this gap, inpainting intermediate views under the guidance of surrounding known regions.

3 Methods

The MeSS pipeline (Fig. 2) synthesizes views along a virtual camera path and reconstructs a Gaussian scene from them in a sparse-to-dense scheme; every generation step is conditioned on control signals rendered from the mesh, realizing the control-distribution match of Sec. 1. Given a 3D city map (*i.e.*, a mesh model

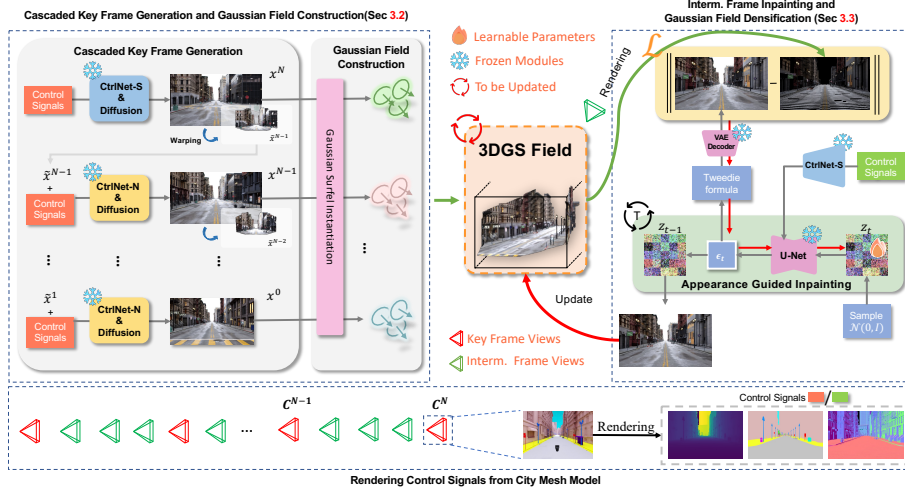


Fig. 2. Overview of the MeSS pipeline. **Bottom:** given a city mesh and a camera path, we render control signals (depth, semantics, normals) at each viewpoint. **Left (Stage I):** key views are generated autoregressively—starting from the last pose with *CtrlNet-S*, then outpainting backwards via *CtrlNet-N* with warped reference images—and instantiated as Gaussian surfels on the mesh surface to form a 3DGS field. **Right (Stage II):** intermediate frames rendered from the Gaussian field are refined by Appearance Guided Inpainting, densifying the field with each inpainted view.

with semantic and instance labels but without texture) and a camera path of M views, **Stage I** generates N sparse key frames by *warp-and-outpaint* with two cascaded CtrlNets and instantiates them as Gaussian surfels on the mesh surface (Sec. 3.2). **Stage II** renders the intermediate views between consecutive key frames from this Gaussian field and repairs their silhouette artifacts with *Appearance Guided Inpainting* (Sec. 3.3). Lastly, *Global Consistency Alignment* (Sec. 3.4) harmonizes the appearance of surfels learned from different views.

3.1 Preliminaries

Conditioned Image Diffusion Model. In a Latent Diffusion Model (LDM) [41], images are encoded by a pretrained autoencoder $(\mathcal{E}, \mathcal{D})$ and a network $\hat{\epsilon}_\theta$ is trained to denoise the latent; ControlNet [56] adds geometric control by training a copy conditioned on a signal $\mathbf{c}$ (e.g., semantics or depth). In the forward process a clean latent is corrupted as $\mathbf{z}_t = \alpha_t \mathbf{z}_0 + \sigma_t \epsilon$ with $\epsilon \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$. Given the noise estimate, Tweedie’s formula [13] recovers the posterior mean of the clean latent,

$$\hat{\mathbf{z}}_0 = \frac{\mathbf{z}_t - \sigma_t \hat{\epsilon}_\theta(\mathbf{z}_t, \mathbf{c}, t)}{\alpha_t}, \quad (1)$$

and $\hat{\mathbf{y}} = \mathcal{D}(\hat{\mathbf{z}}_0)$ gives a differentiable clean-image prediction at any step, which is essential for the gradient guidance in Eq. (5).

Gaussian Surfels from 2D Images. 3D Gaussian Splatting (3DGS) [22] renders scenes via differentiable rasterization of Gaussian kernels; following [53,19], Gaussian *surfels* flatten them into surface-aligned disks parameterized by position, orientation, tangential scales, opacity, and color. Given an RGB image with depth, one surfel is instantiated per pixel by back-projecting its position, copying the RGB color, aligning its orientation to the surface normal, and setting the scales proportional to $d/(\sqrt{2}f)$ (d depth, f focal length) for seamless coverage [53]; we give the full parameterization in the supplementary material. *Multiview Consistency Sampling.* VistaDream [47] introduces training-free *Multiview Consistency Sampling* (MCS): at each step t it takes per-view clean estimates $\{\hat{\boldsymbol{\mu}}_t^{(n)}\}$, fits a temporary 3DGS, re-renders consistent images $\{\bar{\boldsymbol{\mu}}_t^{(n)}\}$, and rectifies the denoising direction by blending

$$\tilde{\boldsymbol{\mu}}_t^{(n)} = w_t \gamma_t^{(n)} \bar{\boldsymbol{\mu}}_t^{(n)} + (1 - w_t) \hat{\boldsymbol{\mu}}_t^{(n)}, \quad (2)$$

where w_t trades consistency against detail and $\gamma_t^{(n)} = \text{std}(\hat{\boldsymbol{\mu}}_t^{(n)})/\text{std}(\bar{\boldsymbol{\mu}}_t^{(n)})$ prevents exposure drift.

Guided Diffusion Sampling. Sampling can be steered toward a partially observed target $\mathbf{y}_{\text{guide}}$ with visibility mask $\mathbf{M}_{\text{guide}}$ [12,53,46], by adjusting the noise estimate of the standard denoising step

$$\mathbf{z}_{t-1} = \text{Denoise}(\mathbf{z}_t, t, \hat{\boldsymbol{\epsilon}}_t) \quad (3)$$

with a guidance gradient:

$$\hat{\boldsymbol{\epsilon}}_t = \hat{\boldsymbol{\epsilon}}_\theta(\mathbf{z}_t, \mathbf{c}, t) - s_t \mathbf{g}_t, \quad (4)$$

$$\mathbf{g}_t = \nabla_{\mathbf{z}_t} \|(\hat{\mathbf{y}}_t - \mathbf{y}_{\text{guide}}) \odot \mathbf{M}_{\text{guide}}\|^2, \quad (5)$$

where s_t controls the strength and $\hat{\mathbf{y}}_t = \mathcal{D}(\hat{\mathbf{z}}_0)$ is decoded from the Tweedie estimate (Eq. (1)).

3.2 Stage I: Gaussian Field Construction from Key Views

Cascaded Key Frame Generation. Two multi-conditioned *CtrlNets* [56] are introduced to generate N key views in autoregressive fashion, moving backwards from the end to the start of the camera path⁵. Both networks build on the standard ControlNet architecture but are extended to accept multiple control signals simultaneously; we illustrate their architecture in the supplementary material. *CtrlNet-S* receives three geometric conditions—depth $\mathbf{d}$, semantic segmentation $\mathbf{s}$, and surface normals $\mathbf{n}$ —concatenated along the channel dimension and processed through convolutional layers before being fed into the ControlNet encoder. Since single-channel disparity maps may lose the ability to discriminate depth values in the far field, we encode depth as a colormap to preserve both near-field and far-field depth variations. *CtrlNet-N* extends this design with an additional

⁵ Views are indexed in driving direction.

reference image condition. The geometric maps follow the same processing path as in *CtrlNet-S*, while the reference image is separately encoded by the frozen VAE encoder. The resulting latent features are element-wise added to the geometric condition features before being fed into the ControlNet, enabling the network to outpaint regions consistent with the reference view.

As illustrated in Fig. 2 (left), starting from the last camera pose, *CtrlNet-S* takes $\{\mathbf{d}^N, \mathbf{s}^N, \mathbf{n}^N\}$ at $\mathcal{C}^N$ to generate the initial view image $\mathbf{x}^N$. Having generated $\mathbf{x}^N$, we warp the image content to its subsequent key view $\mathcal{C}^{N-1}$ via the known depth map. The reason for working backwards from the last frame is that, in a forward-facing camera, the warping will contract the pixel coordinates towards the image center. In this way, image generation artifacts due to limitations of the ControlNet, which mostly occur in the far field, will be diminished, whereas forward warping would amplify them and cause error build-up. To fill the peripheral regions where warped content is not available, *CtrlNet-N* outpaints the missing values, taking the warped image $\tilde{\mathbf{x}}^{N-1}$ as its reference condition alongside the geometric maps $\{\mathbf{d}^{N-1}, \mathbf{s}^{N-1}, \mathbf{n}^{N-1}\}$. The alternation between warping and outpainting is repeated until the first view of the sequence is reached. The next key frame is placed by thresholding the fraction of yet-unobserved (to-be-outpainted) area along the path, which corresponds to intervals of ~ 20 m on our trajectories.

However, the extended key frames generated with the described *warp-and-outpaint* scheme may exhibit noticeable seams (Fig. 3(b), left) due to different exposures. To remove them and achieve coherent appearance, we synchronize all key frames with a *Global Consistency Alignment*, which will be discussed in Sec. 3.4.

Gaussian Field Construction on the Mesh Surface. From the set of key frames $\{\mathbf{x}^n, n \in [1, N]\}$, a 3D Gaussian field is constructed. For the last frame $\mathbf{x}^N$ (*i.e.* the first generated image) we instantiate a Gaussian splat for every pixel. In subsequent key frames $\mathbf{x}^n$, further Gaussians are added to fill in regions with low opacity values in the Gaussian field, in other words we also “outpaint” the Gaussians.

Since precise metric depth and surface normals are directly available from the mesh, we bypass any depth estimation and instantiate one surfel per pixel directly from the mesh depth and normals, without costly Gaussian field optimization. Additional details are provided in the supplementary material.

3.3 Stage II: Intermediate Frame Inpainting and Gaussian Field Densification

Appearance Guided Inpainting. We render the constructed scene at intermediate novel views $\{\mathcal{C}^l\}$ between two key frames $\mathbf{x}^n$ and $\mathbf{x}^{n+1}$, $n \in [1, N)$. Rendered from the coarse scene model, these novel views exhibit silhouettes—regions of low rendered opacity in the Gaussian field (see the supplementary material for an illustration). Novel views not included during the 3DGS reconstruction commonly suffer from holes and blurry textures [22]. We instantiate the guided diffusion

sampling framework (Eqs. (4) and (5)) for RGB inpainting, yielding a training-free method termed *Appearance Guided Inpainting* (AGInpaint). Concretely, as shown in Fig. 2 (right), the target signal $\mathbf{y}_{\text{guide}}$ is set to the rendered RGB image, and $\mathbf{M}_{\text{guide}}$ is obtained by thresholding the rendered opacity from the Gaussian field, thereby masking the silhouette regions that require inpainting. At each denoising step, the clean image $\hat{\mathbf{y}}_t = \mathcal{D}(\hat{\mathbf{z}}_0)$ is obtained via Tweedie’s formula (Eq. (1)), and the guidance gradient $\mathbf{g}_t$ (Eq. (5)) steers the noise estimate so that the prediction matches the known pixels outside the mask. This rectification is repeated N_g times per step, incrementally updating $\hat{\epsilon}_t$ with a learning rate lr , corresponding to gradient descent toward a local optimum that preserves sharp details and consistent textures. The guidance is effective across the full sampling trajectory: at early timesteps it aligns the overall color palette with the reference, while at later timesteps the progressive refinement corrects fine-grained details.

The complete procedure, together with the adaptive rule that regulates the step size s_t by scaling the gradient with the ratio of the mean absolute magnitudes of the current noise estimate and the gradient, is given in the supplementary material. All newly inpainted pixels are spread onto the mesh surface in the aforementioned way.

3.4 Global Consistency Alignment

The Gaussian field produced by the above two stages still exhibits appearance artifacts such as brightness drift and outpainting mistakes. To resolve them, we design *Global Consistency Alignment* (GCAAlign), which applies the MCS framework (Eq. (2)) to our sparse-to-dense pipeline.

Concretely, we re-render the key views $\mathbf{x}'^{(1:N)}$ from the current Gaussian field at viewpoints $\mathcal{C}^{(1:N)}$ and add T_1 steps of forward-diffusion noise ($T_1=9$ of our 50-step inference schedule) to obtain $\mathbf{x}_T'^{(1:N)}$. Tweedie’s formula (Eq. (1)) yields noise-free estimates $\tilde{\mathbf{x}}_0'^{(1:N)}$, which are blended with the rendered views following Eq. (2): the rendered images $\mathbf{x}'^{(n)}$ take the role of the 3DGS-consistent term $\bar{\mu}_t^{(n)}$, and these noise-free estimates $\tilde{\mathbf{x}}_0'^{(n)}$ take the role of the diffusion prediction $\hat{\mu}_t^{(n)}$, with the exposure-balancing factor $\gamma_t^{(n)}$ and trade-off weight w_t defined identically.

The blended estimates serve to refine the Gaussian field; the updated scene is then re-rendered, and the process repeats across the denoising trajectory, progressively enforcing multi-view consistency while preserving fine details. Finally, the same procedure is applied to the sub-sequences of intermediate frames $\mathcal{C}^{(n:(n+1)k)}$ between consecutive key frames.

4 Experiments

4.1 Data Preparation and Implementation

City Scene Data. We use the City Sample Project [15] in Unreal Engine 5 [14], which features two distinct cities with varied, realistic outdoor scenes. Camera



Fig. 3. (a) Resample (left) vs. AGInpaint (right): AGInpaint inpaints slim regions more faithfully. (b) Without (left) and with (right) GCAAlign: GCAAlign harmonizes the seams caused by exposure differences.

sequences are rendered as RGB plus depth, normal, and semantic maps using modified tools from MatrixCity [26] and EasySynth [58], at 960×544 with a 45° field of view, simulating a forward-facing vehicle camera sampled at 1 m intervals. In total, 16k images with paired control signals are obtained for training.

Implementation Details. Both CtrlNets [56] are trained on frozen Stable Diffusion 1.5 (SD1.5) [41] with a learning rate of 10^{-5} and a batch size of 128 for 10k iterations. The reference image for *CtrlNet-N* is randomly picked from 10–20 meters ahead of the current view. For long-range generation, the sequence is divided into blocks of 200 frames processed autoregressively, each block starting from the last generated frame of the previous one.

To stabilize the one-step clean-signal estimation required by the guidance gradient (Eq. (1)), we load LCM-LoRA weights [33] into SD1.5, whose self-consistency property [32] provides a reliable noise-free prediction at any timestep. Guided rectification runs on the latent with an SDEdit-style [34] 50% noise initialization for $N_g = 64$ iterations with a learning rate $lr = 0.00375$ at each of 5 LCM denoising steps. Per-stage runtime and memory are reported in the supplementary material.

4.2 Comparison with Other Methods

We compare with methods from 3D city generation, video-based scene generation, and perpetual view generation. Generation quality is measured by FID and KID against ground-truth imagery, while cross-view consistency is evaluated via LPIPS on extended sequences generated from a real starting image. Because MeSS is, to our knowledge, the only outdoor mesh-based 3DGS generator, WonderWorld[†] and VistaDream[†] serve as directly matched baselines run end-to-end under identical geometric inputs, whereas StreetScapes and CityDreamer4D serve as paradigm reference points.

Comparison with Perpetual View Generation. Both WonderWorld [53] and VistaDream [47] rely on monocular depth estimation methods [4,21] to give depth value for placing Gaussians into 3D space, while our method is based on mesh geometry with precise metric depth. To form a fair comparison, we inject metric depth into their pipelines and align their inpainting diffusion model with ours; the altered versions are denoted WonderWorld[†] and VistaDream[†]. For each cam-

Table 1. *Quantitative evaluations* on generated sequences.

	LPIPS↓	FID↓	KID↓
CityDreamer4D [51]	-	88.48	0.049
WonderWorld [†] [53]	0.516	75.81	0.076
VistaDream [†] [47]	0.508	72.44	0.073
StreetScapes [11]	0.519	29.93	0.025
MeSS (Ours)	0.348	28.17	0.016

Table 2. *Ablation studies.* Each row disables one pipeline component.

	LPIPS↓	FID↓	KID↓
w/o <i>CtrlNet-N</i>	0.382	54.54	0.053
w/o <i>AGInpaint</i>	0.422	51.12	0.046
w/o <i>GCAAlign</i>	0.346	26.25	0.013
Full MeSS	0.348	28.17	0.016

era path, we start the scene from the same initial view and outpaint it by moving the camera backwards for 200 meters with both baselines and our pipeline, producing 3200 images in total.

As depicted in Tab. 1, our method achieves better metrics across the board. We visualize several intermediate frames in Fig. 4: for the same outpainting task, our results stay consistent and align well with the underlying geometry, whereas both baselines suffer from severe quality degradation caused by accumulated inpainting discrepancies. Notably, although VistaDream[†] is equipped with Multiview Consistency Sampling, it shows no apparent improvement over WonderWorld[†]; consistency sampling pays off only when coupled with consistent inpainting, as in our pipeline. We also include the naive WonderWorld in Fig. 4: lacking any prior about the building geometry, it cannot extend facades and drifts to content irrelevant to the initial view, underscoring the difficulty of this task.

Comparison with 3D City Generation. CityDreamer4D [51] is implemented on the Citytopia dataset [51], which is also constructed based on the City Sample Project in Unreal Engine 5 (UE5), similar to ours. In comparison, our method demonstrates superior performance, as evidenced by lower FID and KID (Tab. 1). Furthermore, as illustrated in Fig. 4, their generated scenes exhibit notable limitations in visual quality upon closer inspection, particularly at street level. Specifically, the results lack critical urban elements such as streetlamps, traffic signs, and other street-level instances, significantly diminishing their realism and practical applicability.

Comparison with Video-Based Generation. For StreetScapes [11], we directly use the results reported in their paper since their code is not publicly available. However, forming a fair comparison is challenging, as their model is trained on a significantly larger dataset compared to ours⁶. Given that our generated sequence is longer than theirs (200 frames vs. 64 frames), we evaluate their results within the range of 32 to 64 frames. In terms of quality, our results are on par with theirs, while we outperform them in terms of temporal consistency. This advantage can be attributed to our method’s effective utilization of geometric priors, which plays a key role in achieving superior consistency.

⁶ Even for their experiment on the London patch, they utilize approximately 20 times more data than we do.

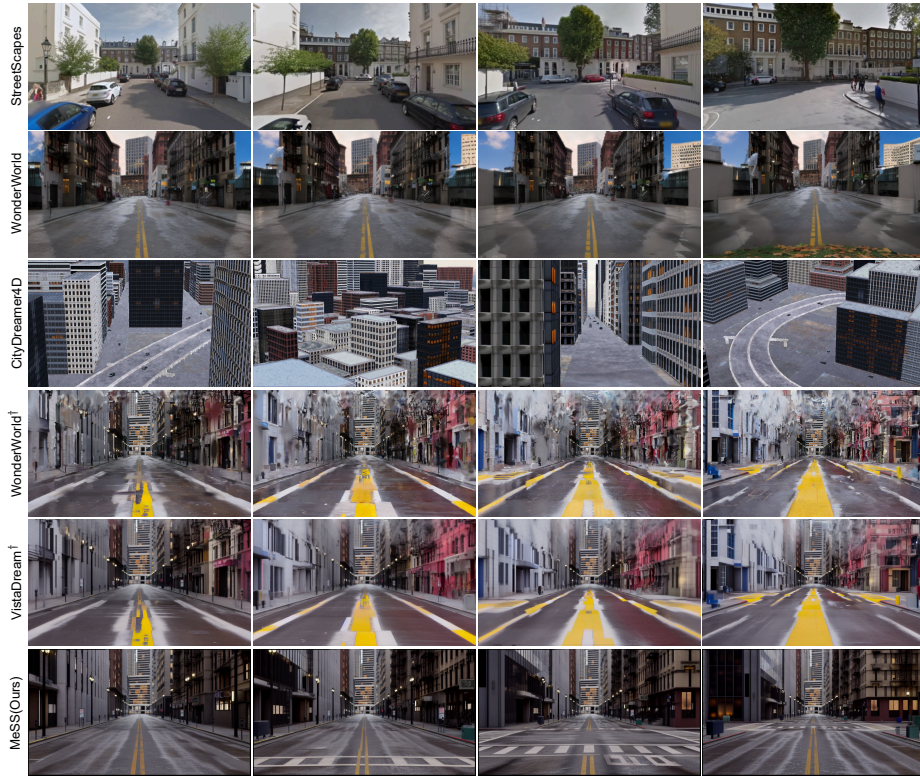


Fig. 4. Qualitative comparison with scene-generation methods. Please zoom in to inspect details.

4.3 Real-World LoD3 Generalization

To test transfer beyond the synthetic training domain, we apply MeSS *without any retraining* to TUM2Twin [45], a public real LoD3 mesh of central Munich. Conditioned only on mesh-rendered depth, normal, and semantic priors, MeSS generates facades that follow the LoD3 geometry (*e.g.*, doors and windows align with the rendered mesh) and stay consistent across views (Fig. 5). This supports the control-distribution argument of Sec. 1: mesh-rendered priors are in-distribution for our CtrlNets. The residual failure mode is UE5-style texture drift on real facades, which a light Stage-I fine-tune on real imagery would close. The bottom row of Fig. 5 shows a head-to-head with the video-diffusion ControlNet Cosmos-Transfer2.5 [37], which preserves only a coarse layout; we analyze this comparison and give further details in the supplementary material.

4.4 Ablation Study

Cascaded CtrlNets. For the outpainting of key views in Stage I, we replace *CtrlNet-N*, which is conditioned on the preceding frame, with *CtrlNet-S. AGIn-*

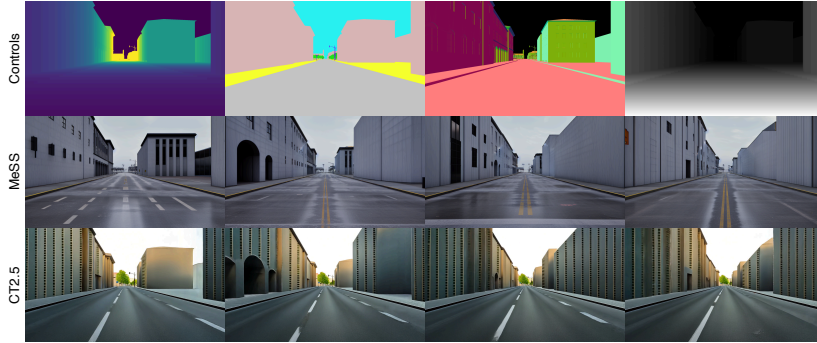


Fig. 5. Zero-shot generalization to the real LoD3 mesh TUM2Twin [45], no retraining. **Top:** mesh-rendered control signals—colormapped depth, semantics, normals, and the grayscale depth given to the video baseline. **Middle:** street-level views generated by MeSS along the camera path; facades follow the LoD3 geometry and remain consistent across views. **Bottom:** Cosmos-Transfer2.5 (CT2.5), a video-diffusion ControlNet conditioned on the same mesh geometry, preserves only a coarse layout with incorrect facade detail.

paint remains enabled. This setting differs from standard pipelines that combine warping and outpainting [53,54]. As depicted in Tab. 2, the fidelity of generated frames degrades dramatically, and consistency also deteriorates because appearance drift accumulates during scene extrapolation.

Appearance Guided Inpainting. We replace the *AGInpaint* component in our method with a simpler approach called *Resample* [11,31], which involves re-adding several rounds of noise at each denoising step to achieve a more homogeneous inpainted result. As demonstrated in Tab. 2, the absence of *AGInpaint* significantly impacts the generation quality. Through a visual comparison of the inpainted regions produced by *AGInpaint* and *Resample* (Fig. 3(a)), we highlight the superior efficacy of *AGInpaint*. Due to the downsampling of inpainting masks, *Resample* struggles to fill in slim regions with Latent Diffusion Models, causing noticeable streak patterns.

Global Consistency Alignment. As shown in Fig. 3(b), the generated scene exhibits noticeable lighting misalignment when *GCAAlign* is not applied, whereas integrating *GCAAlign* into our pipeline effectively harmonizes the seams. An interesting observation is that *GCAAlign* has a trade-off: while it improves visual coherence, it tends to introduce a slight blurring effect, leading to a loss of fine details. This is also reflected in the marginally better quantitative metrics when *GCAAlign* is disabled (Tab. 2). Nevertheless, we consider this trade-off acceptable in favor of achieving a more visually consistent appearance, and it can be disabled when per-frame fidelity is preferred.

5 Conclusion

We present MeSS, a pipeline for generating Gaussian fields from city mesh models, built on a *control-distribution match*: CtrlNets trained directly on mesh-rendered priors yield tight alignment between geometry and generated appearance. By combining *Cascaded Outpainting CtrlNets*, *AGInpaint*, and *GCAAlign*, our method produces geometry-aligned scenes with coherent appearance across extended view sequences at low training cost, outperforming state-of-the-art scene generators on a city-scale benchmark and transferring zero-shot to a real LoD3 mesh.

Limitations. Our method is trained on high-detail meshes from the UE5 City Sample Project, and we show it transfers zero-shot to a real LoD3 mesh (Fig. 5), preserving mesh-aligned layout while inheriting a UE5-style texture prior. Closing this residual texture-domain gap on a specific real city would benefit from a light fine-tune on real street imagery, which is far cheaper than the full CtrlNet training. In future work, integrating stronger base models (*e.g.*, FLUX.1, SD3) and incorporating geometric priors into video diffusion are promising directions.

References

1. Alhaija, H.A., Alvarez, J., Bala, M., Cai, T., Cao, T., Cha, L., Chen, J., Chen, M., Ferroni, F., Fidler, S., et al.: Cosmos-transfer1: Conditional world generation with adaptive multimodal control. arXiv preprint arXiv:2503.14492 (2025)
2. Bhat, S.F., Birkel, R., Wofk, D., Wonka, P., Müller, M.: Zoedepth: Zero-shot transfer by combining relative and metric depth. arXiv preprint arXiv:2302.12288 (2023)
3. Blattmann, A., Dockhorn, T., Kulal, S., Mendelevitch, D., Kilian, M., Lorenz, D., Levi, Y., English, Z., Voleti, V., Letts, A., et al.: Stable video diffusion: Scaling latent video diffusion models to large datasets. arXiv preprint arXiv:2311.15127 (2023)
4. Bochkovskii, A., Delaunoy, A., Germain, H., Santos, M., Zhou, Y., Richter, S.R., Koltun, V.: Depth pro: Sharp monocular metric depth in less than a second. arXiv preprint arXiv:2410.02073 (2024)
5. Cai, S., Chan, E.R., Peng, S., Shahbazi, M., Obukhov, A., Van Gool, L., Wetzstein, G.: Diffdreamer: Towards consistent unsupervised single-view scene extrapolation with conditional diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2139–2150 (2023)
6. Chai, L., Tucker, R., Li, Z., Isola, P., Snavely, N.: Persistent nature: A generative model of unbounded 3d worlds. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 20863–20874 (2023)
7. Chen, D.Z., Li, H., Lee, H.Y., Tulyakov, S., Nießner, M.: Scenetex: High-quality texture synthesis for indoor scenes via diffusion priors. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 21081–21091 (2024)
8. Chen, D.Z., Siddiqui, Y., Lee, H.Y., Tulyakov, S., Nießner, M.: Text2tex: Text-driven texture synthesis via diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 18558–18568 (2023)

9. Chen, Z., Wang, G., Liu, Z.: Scenedreamer: Unbounded 3d scene generation from 2d image collections. *IEEE transactions on pattern analysis and machine intelligence* (2023)
10. Chung, J., Lee, S., Nam, H., Lee, J., Lee, K.M.: Luciddreamer: Domain-free generation of 3d gaussian splatting scenes. *arXiv preprint arXiv:2311.13384* (2023)
11. Deng, B., Tucker, R., Li, Z., Guibas, L., Snavely, N., Wetzstein, G.: Streetscapes: Large-scale consistent street view generation using autoregressive video diffusion. In: *ACM SIGGRAPH 2024 Conference Papers*. pp. 1–11 (2024)
12. Dhariwal, P., Nichol, A.: Diffusion models beat gans on image synthesis. *Advances in neural information processing systems* **34**, 8780–8794 (2021)
13. Efron, B.: Tweedie’s formula and selection bias. *Journal of the American Statistical Association* **106**(496), 1602–1614 (2011)
14. Epic Games: Unreal engine 5. <https://www.unrealengine.com/en-US/unreal-engine-5> (2021), accessed: 2025-08-01
15. Epic Games: City sample project. <https://www.unrealengine.com/marketplace/en-US/product/city-sample> (2022), accessed: 2025-08-01
16. Fan, Z., Wen, K., Cong, W., Wang, K., Zhang, J., Ding, X., Xu, D., Ivanovic, B., Pavone, M., Pavlakos, G., et al.: Instantplat: Sparse-view sfm-free gaussian splatting in seconds. *arXiv preprint arXiv:2403.20309* (2024)
17. Fridman, R., Abecasis, A., Kasten, Y., Dekel, T.: Scenescape: Text-driven consistent scene generation. *Advances in Neural Information Processing Systems* **36**, 39897–39914 (2023)
18. Gao, S., Yang, J., Chen, L., Chitta, K., Qiu, Y., Geiger, A., Zhang, J., Li, H.: Vista: A generalizable driving world model with high fidelity and versatile controllability. *Advances in Neural Information Processing Systems* **37**, 91560–91596 (2025)
19. Guédon, A., Lepetit, V.: Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 5354–5363 (2024)
20. Guo, Y., Yang, C., Rao, A., Liang, Z., Wang, Y., Qiao, Y., Agrawala, M., Lin, D., Dai, B.: Animatediff: Animate your personalized text-to-image diffusion models without specific tuning. *arXiv preprint arXiv:2307.04725* (2023)
21. Ke, B., Obukhov, A., Huang, S., Metzger, N., Daut, R.C., Schindler, K.: Repurposing diffusion-based image generators for monocular depth estimation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 9492–9502 (2024)
22. Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.* **42**(4), 139–1 (2023)
23. Koo, G., Yoon, S., Lee, Y., Hong, J.W., Yoo, C.D.: Flowdrag: Mesh-guided drag-based image editing with deformation vector flow fields. In: *ICML* (2025)
24. Landesbetrieb Geoinformation und Vermessung Hamburg: 3D-Gebäudemodell LoD3.0-HH. <https://metaver.de/trefferanzeige?docuuid=B438AD57-223B-43A4-8E74-767CEC8A96D7> (2025), cityGML dataset, Freie und Hansestadt Hamburg, Datenlizenz Deutschland – Namensnennung 2.0. Last updated April 2025
25. Li, X., Zhang, Y., Ye, X.: Drivingdiffusion: Layout-guided multi-view driving scenarios video generation with latent diffusion model. In: *European Conference on Computer Vision*. pp. 469–485. Springer (2024)
26. Li, Y., Jiang, L., Xu, L., Xiangli, Y., Wang, Z., Lin, D., Dai, B.: Matrixcity: A large-scale city dataset for city-scale neural rendering and beyond. In: *Proceedings*

- of the IEEE/CVF International Conference on Computer Vision. pp. 3205–3215 (2023)
27. Li, Z., Wang, Q., Snavely, N., Kanazawa, A.: Infinitenature-zero: Learning perpetual view generation of natural scenes from single images. In: European Conference on Computer Vision. pp. 515–534. Springer (2022)
 28. Lin, C.H., Lee, H.Y., Menapace, W., Chai, M., Siarohin, A., Yang, M.H., Tulyakov, S.: Infinitcity: Infinite-scale city synthesis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 22808–22818 (2023)
 29. Liu, A., Tucker, R., Jampani, V., Makadia, A., Snavely, N., Kanazawa, A.: Infinite nature: Perpetual view generation of natural scenes from a single image. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14458–14467 (2021)
 30. Liu, F., Sun, W., Wang, H., Wang, Y., Sun, H., Ye, J., Zhang, J., Duan, Y.: Reconx: Reconstruct any scene from sparse views with video diffusion model. arXiv preprint arXiv:2408.16767 (2024)
 31. Lugmayr, A., Danelljan, M., Romero, A., Yu, F., Timofte, R., Van Gool, L.: Repaint: Inpainting using denoising diffusion probabilistic models. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11461–11471 (2022)
 32. Luo, S., Tan, Y., Huang, L., Li, J., Zhao, H.: Latent consistency models: Synthesizing high-resolution images with few-step inference. arXiv preprint arXiv:2310.04378 (2023)
 33. Luo, S., Tan, Y., Patil, S., Gu, D., von Platen, P., Pasber, A., Huang, L., Li, J., Zhao, H.: Lcm-lora: A universal stable-diffusion acceleration module. arXiv preprint arXiv:2311.05556 (2023)
 34. Meng, C., He, Y., Song, Y., Song, J., Wu, J., Zhu, J.Y., Ermon, S.: Sdedit: Guided image synthesis and editing with stochastic differential equations. arXiv preprint arXiv:2108.01073 (2021)
 35. Metzger, G., Richardson, E., Patashnik, O., Giryes, R., Cohen-Or, D.: Latent-nerf for shape-guided generation of 3d shapes and textures. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12663–12673 (2023)
 36. Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. Communications of the ACM **65**(1), 99–106 (2021)
 37. NVIDIA: World simulation with video foundation models for physical ai. arXiv preprint arXiv:2511.00062 (2025)
 38. Peters, R., Dukai, B., Vitalis, S., van Liempt, J., Stoter, J.: Automated 3d reconstruction of lod2 and lod1 models for all 10 million buildings of the netherlands (2022). <https://doi.org/10.14358/PERS.21-00032R2>
 39. Ranftl, R., Lasinger, K., Hafner, D., Schindler, K., Koltun, V.: Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. IEEE transactions on pattern analysis and machine intelligence **44**(3), 1623–1637 (2020)
 40. Richardson, E., Metzger, G., Alaluf, Y., Giryes, R., Cohen-Or, D.: Texture: Text-guided texturing of 3d shapes. In: ACM SIGGRAPH 2023 conference proceedings. pp. 1–11 (2023)
 41. Rombach, R., Blattmann, A., Lorenz, D., Esser, P., Ommer, B.: High-resolution image synthesis with latent diffusion models. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 10684–10695 (June 2022)

42. SAVeNoW, P.: Lod3 road space models. <https://github.com/savenow/lod3-road-space-models> (2020)
43. Shi, Y., Xue, C., Liew, J.H., Pan, J., Yan, H., Zhang, W., Tan, V.Y.F., Bai, S.: Dragdiffusion: Harnessing diffusion models for interactive point-based image editing. In: CVPR (2024)
44. Sun, W., Chen, S., Liu, F., Chen, Z., Duan, Y., Zhang, J., Wang, Y.: Dimensionx: Create any 3d and 4d scenes from a single image with controllable video diffusion. arXiv preprint arXiv:2411.04928 (2024)
45. TUM2Twin Project: TUM2Twin: Digital twin of TU Munich (lod2/lod3) (2025)
46. Viola, M., Qu, K., Metzger, N., Ke, B., Becker, A., Schindler, K., Obukhov, A.: Marigold-dc: Zero-shot monocular depth completion with guided diffusion. arXiv preprint arXiv:2412.13389 (2024)
47. Wang, H., Liu, Y., Liu, Z., Wang, W., Dong, Z., Yang, B.: Vistadream: Sampling multiview consistent images for single-view scene reconstruction. arXiv preprint arXiv:2410.16892 (2024)
48. Wang, X., Zhu, Z., Huang, G., Chen, X., Zhu, J., Lu, J.: Drivedreamer: Towards real-world-drive world models for autonomous driving. In: European Conference on Computer Vision. pp. 55–72. Springer (2024)
49. Xie, H., Chen, Z., Hong, F., Liu, Z.: Citydreamer: Compositional generative model of unbounded 3d cities. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9666–9675 (2024)
50. Xie, H., Chen, Z., Hong, F., Liu, Z.: Gaussiancity: Generative gaussian splatting for unbounded 3d city generation. arXiv preprint arXiv:2406.06526 (2024)
51. Xie, H., Chen, Z., Hong, F., Liu, Z.: Citydreamer4d: Compositional generative model of unbounded 4d cities (2025), <https://arxiv.org/abs/2501.08983>
52. Yao, Z., Chaturvedi, K., Kolbe, T.H.: 3DCityDBV5: Open Source 3D City Database for CityGML. Chair of Geoinformatics, Technical University of Munich (2025), version 5.0, <https://www.3dcitydb.org/>
53. Yu, H.X., Duan, H., Herrmann, C., Freeman, W.T., Wu, J.: Wonderworld: Interactive 3d scene generation from a single image. arXiv preprint arXiv:2406.09394 (2024)
54. Yu, H.X., Duan, H., Hur, J., Sargent, K., Rubinstein, M., Freeman, W.T., Cole, F., Sun, D., Snavely, N., Wu, J., et al.: Wonderjourney: Going from anywhere to everywhere. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6658–6667 (2024)
55. Yu, W., Xing, J., Yuan, L., Hu, W., Li, X., Huang, Z., Gao, X., Wong, T.T., Shan, Y., Tian, Y.: Viewcrafter: Taming video diffusion models for high-fidelity novel view synthesis. arXiv preprint arXiv:2409.02048 (2024)
56. Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 3836–3847 (2023)
57. Zhao, G., Wang, X., Zhu, Z., Chen, X., Huang, G., Bao, X., Wang, X.: Drivedreamer-2: Llm-enhanced world models for diverse driving video generation. arXiv preprint arXiv:2403.06845 (2024)
58. Zhao, Y., Zhou, Z., Wang, Y., Huang, J., Zhao, H.: Easysynth: A unified toolkit for generating multi-modal synthetic data. <https://github.com/ydrive/EasySynth> (2024), gitHub repository, accessed: 2025-08-01