CitySculpt: 3D City Generation from Satellite Imagery with UV Diffusion

Anonymous Author(s)
Submission Id: 2285

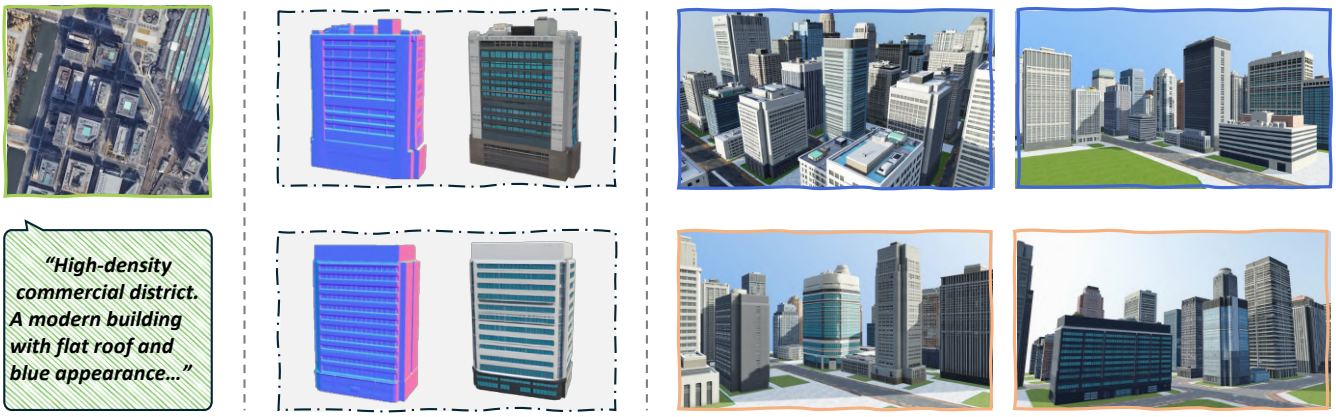

Figure 1: Overview of our CitySculpt framework. (a) Given satellite imagery as input, we first perform multi-scale scene understanding; (b) then, we design a diffusion-based network to generate high-quality 3D objects despite the limited information from satellite views; (c) finally, these objects are assembled into complete 3D cities with diverse styles.

Abstract

Generating 3D cities from satellite imagery opens up new avenues for gaming, urban planning, and cinematic production. However, the limited information from satellite views presents significant challenges, hindering existing methods from generating high-quality cities that meet application standards. To address these challenges, we propose CitySculpt, a UV diffusion-based framework for generating 3D cities with high-fidelity geometry and photorealistic textures. Specifically, we first generate the detailed 3D geometries by refining coarse structures using a UV normal diffusion network. Building on these refined geometries, we introduce a texture generation approach that produces photorealistic textures despite the limited satellite information. To ensure style consistency across multiple objects, we design a cross-attention mechanism that enables feature sharing among them. Additionally, we contribute the CitySculpt dataset, a collection of high-quality 3D urban assets with multi-view renderings and comprehensive annotations to advance research in 3D city generation. Experiments demonstrate that CitySculpt outperforms state-of-the-art approaches in both generating detailed individual buildings and creating cities with high visual quality and rich architectural details.

CCS Concepts

• **Computing methodologies** → **Computer vision problems**.

Keywords

Generative Model; Scene Generation; Diffusion; Generative Multimedia

ACM Reference Format:
Anonymous Author(s). 2025. CitySculpt: 3D City Generation from Satellite Imagery with UV Diffusion. In *Proceedings of Make sure to enter the correct conference title from your rights confirmation email (Conference acronym 'XX)*. ACM, New York, NY, USA, 13 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

City generation is gaining traction in gaming [10, 25], autonomous driving simulation [8, 28], and cinematic production [33, 37]. A particularly effective approach is generating 3D cities directly from satellite imagery, as it enables the efficient creation of large-scale urban environments without the labor-intensive process of layout design or manual modeling. However, this task remains challenging, as satellite perspectives provide limited information, making it difficult for prior methods [14–16, 37, 38] to generate high-quality geometries and textures that meet application standards.

In recent years, notable advancements [3, 5, 11, 14, 16, 37, 38, 40] in 3D city generation have been made. They typically follow a common process: satellite imagery is first processed to extract

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
Conference acronym 'XX, Dublin, Ireland
© 2025 Copyright held by the owner/author(s). Publication rights licensed to ACM.
ACM ISBN 978-1-4503-XXXX-X/2025/06
https://doi.org/XXXXXXX.XXXXXXX

semantic and height maps. The 3D geometry is then constructed by lifting each pixel from the 2D map to the corresponding height according to the height map. Afterward, they generate texture algorithms are applied to generate textures on the 3D geometry. However, these approaches produce buildings with flat surfaces, *lacking architectural details* such as windows or balconies. This significantly limits their visual realism and applicability.

While 3D asset generation [7, 12, 24, 31, 35, 36, 39, 41, 42] has garnered significant attention for its ability to create photorealistic objects such as avatars, toys, and furniture. It is therefore tempting to adopt these techniques for city generation. However, this integration faces several challenges. Satellite imagery provides only *limited information* from a top-down perspective, posing significant difficulties for generating high-quality buildings with architectural details. More importantly, buildings within the same urban area are often interconnected in style, yet current approaches cannot guarantee *stylistic consistency* across them. This is because these methods focus solely on generating individual buildings, making them unsuitable for city-scale generation.

To address the aforementioned issues, we propose CitySculpt, a diffusion-based framework designed for 3D city generation with *rich geometric details* and *photorealistic visual quality*. As shown in Figure 2, taking satellite imagery as input, we first extract scene characteristics at multiple scales and initialize the coarse 3D geometry. To enhance geometric details, we propose a refinement approach that improves the 3D geometry by generating high-quality UV normal maps. With the refined geometry, our texture diffusion model generates realistic textures despite the limited information available from satellite perspectives. To maintain *stylistic consistency* during the multi-object generation process, we introduce a cross-attention mechanism that enables feature sharing across buildings. In addition, we contribute the *CitySculpt dataset*, a collection of 5,000 high-quality 3D urban assets with multi-view renderings and semantic annotations to support urban-scale generation tasks.

The key contributions are as follows:

- We propose CitySculpt, a UV diffusion-based framework for generating cities with rich geometric details and photorealistic visual quality from satellite imagery.
- We propose a novel geometry refinement approach that optimizes coarse geometries by generating detailed UV normal maps.
- We develop a texture generation approach that produces photorealistic textures from limited satellite perspectives, while maintaining stylistic consistency across multiple buildings through our cross-attention mechanism.
- We present the CitySculpt dataset, a comprehensive collection of high-quality 3D urban assets with multi-view renderings and semantic annotations to facilitate research in city-scale 3D generation.

2 Related Works

2.1 3D City Generation.

Recent approaches have made significant progress in 3D city scene generation, which can be broadly categorized into two groups: layout-based methods and procedural methods.

3D layout-based methods first construct 3D layouts from satellite semantic maps and height fields, then generate textures on these structures. InfiniCity [16] pioneered this framework by constructing Octree-based voxels from satellite data, then employing neural rendering with a SPADE generator for texture synthesis. Building upon this foundation, CityGen [5] improved layout diversity by introducing a multi-scale diffusion model for semantic map generation. Subsequently, CityDreamer [37] proposed a compositional approach that separates building instances from background elements through 3D GAN training on Google Earth data to generate volumetric renderings. Most recently, GaussianCity [38] further enhanced this approach by leveraging Gaussian splatting representation to improve rendering efficiency and visual quality, extending city generation to unbounded scales. However, a common limitation of these methods is that buildings often **lack geometric detail**. This is due to their structures being directly lifted from depth maps, resulting in flat surfaces without architectural features and diminished visual realism.

Procedural-based methods generate urban environments by collecting high-quality 3D object assets and assembling them according to specific rules to create complete 3D cities. These approaches can produce higher quality urban scenes than 3D layout-based methods due to the superior quality of individual assets. CityCraft [6] exemplifies this approach by using large language models as city planners, analyzing satellite imagery to guide urban element allocation based on real-world planning principles. Building on this approach, CityX [44] and SceneX [46] implement a multi-agent framework that coordinates various procedural content generation modules through a management protocol, transforming user descriptions and multimodal inputs into executable programs for urban scene construction. Similarly, ProcGS [13] integrates procedural code with 3D Gaussian Splatting to efficiently generate buildings with repeated architectural elements while maintaining visual fidelity. However, these approaches are limited by their **reliance on existing asset collections**, restricting their ability to generate novel architectural styles.

2.2 3D Assets Generation.

3D asset generation has received significant attention for its ability to create photorealistic 3D objects such as avatars, furniture, toys, and more. Current approaches can be broadly categorized into two types: image-guided methods and UV map-based methods.

Early image-to-3D methods focused on single-view generalization[2, 4, 19, 22, 23, 45]. DreamFusion [24] adapted 2D diffusion priors to optimize 3D representations, producing high-quality results but requiring extensive computation time. CRM [35] adopted a feed-forward approach to efficiently generate high-fidelity 3D models from single images by integrating geometric priors into its convolutional architecture. However, these approaches face limitations in handling complex occlusions and diverse object geometries. To address single-view limitations, multi-view generation methods have emerged[17, 18, 29, 30, 32]. InstantMesh [39] combines multi-view diffusion models with sparse view reconstruction techniques, optimizing geometric consistency for efficient single-image to 3D model conversion. Hi3D [41] reformulates multi-view generation

as an orbital video generation problem, improving accuracy in generating view-consistent images with high-resolution texture details. Beyond texture generation, some works enhance geometry through normal maps, such as CraftsMan [12], which employs normal map refinement as post-processing, and Trellis [36], which incorporates normal rendering constraints during VAE training.

UV map-based methods offer the advantage of generating only a single texture image rather than multiple views, resulting in faster processing. Texturify [31] uses GANs to generate realistic textures directly on 3D surfaces, learning from real images without requiring 3D color supervision or shape-image correspondence. PointUV [42] introduces a coarse-to-fine approach combining point diffusion and UV diffusion, first generating a basic texture through point sampling before refining it with UV diffusion. UV3-TeD [7] introduce a UV-free alternative that represents textures as colored point clouds on mesh surfaces, employing denoising diffusion and geodesic heat propagation to eliminate common UV mapping issues like seams and distortions.

Despite these advances, these approaches typically require substantial information about target 3D assets, limiting their applicability in **minimal-input** scenarios. Additionally, when generating scene-level content with **multiple objects**, these methods struggle to maintain stylistic consistency across different elements.

3 Methods

Figure 2 shows an overview of our CitySculpt framework. Given a satellite image I, we first predict its semantic map, depth map, and density map. This is followed by multi-scale scene understanding and initialization of the coarse 3D city geometry (Section 3.1). After that, we unwrap the objects into UV space, where geometric details are refined using the proposed UV-space normal diffusion model (Section 3.2). Based on this, we introduce a texture diffusion model that generates high-quality textures despite the limited information from the satellite view. The textures are generated in parallel, with cross-attention enabling texture sharing to maintain stylistic consistency across buildings (Section 3.3). Finally, we incorporate background generation (Section 3.4) to enhance the completeness of the scene.

3.1 Multi-scale Scene Understanding and Geometric Initialization

Effective understanding and encoding of input information serve as prerequisites for generating high-quality results, especially with satellite imagery where top-down views provide limited information. Given a satellite image I, we propose a multi-scale scene understanding framework to extract visual and semantic characteristics while constructing an initial coarse 3D geometric structure. This information serves as the foundation for subsequent generation processes.

Multi-scale Feature Extraction Our framework simultaneously analyzes the scene at region and instance levels:

At the *region level*, we first compute a density map D_i through kernel density estimation to identify functional urban zones:

$$D_i = \sum_j k(\mathbf{p} - \mathbf{p}_j) \tag{1}$$

where k represents a Gaussian kernel and $\mathbf{p}_j$ denotes building locations. Based on this density distribution, we partition the scene into N regions $\{R_i\}_{i=1}^N$ using watershed segmentation. For each region, we extract semantic descriptors $T_i = \text{CLIP}(R_i)$ using CLIP [26] to characterize urban patterns such as "high-density commercial district".

At the *instance level*, we analyze individual urban structures by jointly considering their visual information and height data. For each instance j in region i, we first derive its visual and semantic characteristics using CLIP (e.g., “blue modern building”). These features are then fused with height information to enhance architectural traits (e.g., “flat roof”). Specifically, we design the height feature extractor $\omega(\cdot)$ to map the height map $H_{i,j}$ to architectural feature descriptors, including roof geometry and vertical proportions.

Formally, the instance-level feature is defined as:

$$t_{i,j} = \text{CLIP}(\text{instance}_{i,j}) \oplus \omega(H_{i,j}) \tag{2}$$

Finally, we organize the multi-scale information into a unified feature representation, denoted as T_{scene}. This consists of regional features T_i and instance-level features $t_{i,j}$, where each instance feature $t_{i,j}$ is grouped within its corresponding region T_i:

$$T_{\text{scene}} = \left\{ T_i, \{t_{i,j}\}_{j=1}^{N_i} \right\}_{i=1}^{N} \tag{3}$$

3D Coarse Geometry Generation. To generate an initial coarse 3D representation of the scene, we follow an approach similar to CityDreamer [37]. For each instance j with a depth map H_j and a semantic mask S_j, we project the pixels into 3D space using their corresponding depth values:

$$G_j = \{(x, y, H_j(x, y)) \mid (x, y) \in S_j\} \tag{4}$$

Here, each point (x, y, z) represents a 3D coordinate, where $z = H_j(x, y)$ is the depth value, and each point retains its semantic label. Next, we construct the initial 3D mesh M_j using Poisson surface reconstruction, which provides a coarse geometric model that serves as the foundation for further refinement.

3.2 UV-space Geometry Refinement

Motivation. While the coarse 3D geometry in Section 3.1 provides the basic geometric representation of the urban structure, it suffers from flat surfaces with minimal architectural details. Unlike previous works that directly generate textures on these flat geometries, we propose a refinement approach that uses a diffusion model to generate detailed UV normal maps to enhance surface geometry.

UV Normal Diffusion. As shown in Figure 2, for each object in the scene, we first unwrap its mesh to obtain a coarse UV normal map x_{coarse}. By designing N_1, our goal is to generate detailed UV normal maps x_0 from Gaussian noise z_t under the constraint of condition c_1. The formulation of our diffusion model is as follows:

$$\hat{x}_0 = N_1(x_t, t, c_1) \tag{5}$$

To address discontinuity issues in UV space, we introduce a condition vector c that integrates three key information sources:

$$c_1 = \{\gamma(x_{\text{coarse}}), \psi(x_{\text{shape}}), \phi(t_{\text{obj}})\} \tag{6}$$

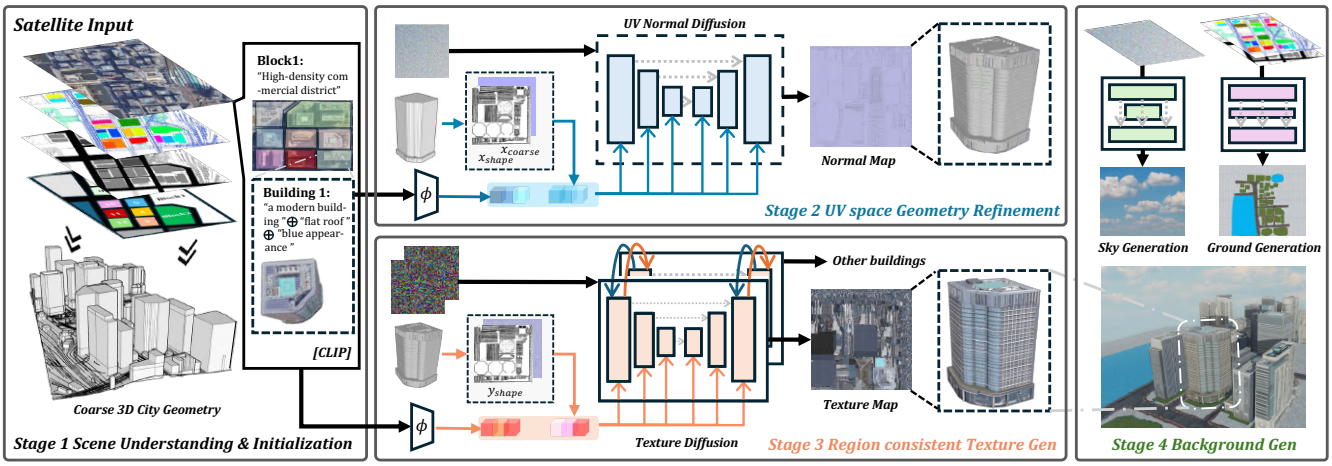

Figure 2: Overview of CitySculpt. Given a satellite image, we first perform multi-scale scene understanding and initialize a coarse 3D city geometry. Building upon this, the second stage refines the initial geometry by generating high-fidelity UV maps using our UV-Normal diffusion method. The third stage further enhances visual realism by synthesizing photorealistic textures while maintaining style consistency across buildings. Finally, the textured buildings are assembled into a complete city scene, including sky and ground generation.

The initial normal map x_{coarse} contains orientation information and is encoded by the normal encoder $\gamma(\cdot)$. The shape map $x_{\text{shape}} = [x_{\text{coord}}, x_{\text{boundary}}]$ captures structural information, with x_{coord} mapping 3D coordinates to UV space and x_{boundary} marking seam locations. We encode it through our boundary feature encoder $\psi(\cdot)$. Additionally, object-level semantic features t_{obj}, extracted from the multi-scale representation T in stage 1, are processed through the text encoder $\phi(\cdot)$. All these feature embeddings are injected into each UNet block through a cross-attention mechanism. The geometric features $\gamma(x_{\text{coarse}})$ and $\psi(x_{\text{shape}})$ are trainable, enabling the model to learn and refine representations of architectural geometry effectively.

The optimization objective combines three components:

$$L = \|\varepsilon - \varepsilon_\theta\|^2 + \lambda_1 L_{\text{unit}} + \lambda_2 L_{\text{cont}} \tag{7}$$

where the first term is the standard diffusion loss that predicts the added noise, $L_{\text{unit}} = \||\hat{n}|^2 - 1\|^2$ enforces the unit vector constraint on generated normals, and $L_{\text{cont}} = \|\hat{n}_{\text{boundary}_1} - \hat{n}_{\text{boundary}_2}\|^2$ minimizes discontinuities at UV seams by aligning normals at corresponding boundary locations. The hyperparameters λ_1 and λ_2 balance these objectives to produce geometrically valid and visually coherent normal maps.

3.3 Region-consistent Texture Generation

Motivation. While existing texture generation methods are effective at producing detailed textures for individual 3D assets, they face two major challenges in urban-scale scenarios. First, satellite imagery provides limited visual information due to its top-down perspective, lacking details on building façades and materials. More importantly, these methods struggle to maintain stylistic consistency across buildings, as they treat each one independently during texture generation.

To address these challenges, we design a diffusion-based network that generates high-quality textures by leveraging the multi-modal information extracted in Section 3.1 . In addition, we incorporate a cross-attention mechanism that enables parallel texture generation for buildings within the same region while facilitating style information sharing among them.

Model Architecture. Our goal is to generate high-quality textures $\hat{y}_0$ for each building by denoising a noisy input y_t under the guidance of structured conditional information. This is formulated as:

$$\hat{y}_0 = N_2(y_t, t, c_2) \tag{8}$$

To fully exploit the limited information available from satellite imagery, we design the conditioning vector c_2 from three components:

$$c_2 = \{\psi(y_{\text{shape}}), \phi(T_r), \{F^l\}\} \tag{9}$$

where the geometric features $y_{\text{shape}} = [x_{\text{shape}}, x_{\text{normal}}]$ are formed by concatenating the geometry representation x_{shape} and the detailed normal map x_{normal} from Section 3.2 . We encode y_{shape} using the same geometry encoder $\psi(\cdot)$, which is trainable during the training process. In addition, to ensure stylistic consistency across buildings within the same region, we encode the region-level feature T_r using the region encoder $\phi(\cdot)$as a control signal. We also design a parallel diffusion architecture that allows feature sharing across buildings within the same region, where $\{F^l\}$ denotes intermediate feature embeddings from other buildings in the region.

Cross-branch Feature Sharing Mechanism. The key to our region-consistent generation lies in the interaction between different generation branches within the same region. Instead of generating each building's texture independently, we enable information exchange through a cross-attention mechanism. In each diffusion

block, after processing the individual building features, we fuse them as follows:

$$F_i' = F_i + \psi(y_{\text{shape}}) + \phi(T_r) \tag{10}$$

where F_i denotes the original feature embedding of building i and F_i' is the fused feature representation. We then apply cross-attention across all buildings in the region:

$$M_i = \sum_{l \neq i} \text{softmax}(W_Q F_i' \cdot (W_K F_l')^T) W_V F_l' \tag{11}$$

Here, W_Q projects the current building's features F_i' into the query space, W_K projects the features F^l from neighboring buildings in the same region into the key space, and W_V projects the same features into the value space. This cross-attention mechanism computes a weighted sum of the neighboring features based on the similarity between the query and key, allowing each building to selectively attend to relevant style features from nearby buildings while preserving its own geometric properties as defined by F_i'.

Finally, the multi-building training objective is defined as:

$$L = \mathbb{E}_{\{y_t\},\varepsilon_i,t}\left[\sum_{i=1}^{N} \|\varepsilon_i - \varepsilon_{i\theta}(\{y_t\}, t, y_{\text{shape}}, F_{\text{region}_r})\|^2\right] \tag{12}$$

3.4 Background Generation

After generating detailed building geometries and textures, we complete the urban scene by synthesizing realistic sky and ground elements. As shown in Figure 2, our approach handles these environmental components differently based on their characteristics and requirements.

Sky Generation. For sky generation, we directly finetune the LDM [27] to produce realistic atmospheric conditions. For consistent scene illumination, we extract directional light parameters from the generated sky and apply a dual-component lighting model: one accounting for light incidence angles on surfaces and another handling shadow mapping based on light visibility calculations.

Ground Generation. For ground surfaces, we adopt a semantic-guided strategy to generate context-aware textures. Given the semantic map, we first identify different functional zones (e.g., roads, parks, plazas) and then apply appropriate textures accordingly:

$$G_{\text{ground}} = \Gamma(S_{\text{ground}}, T_{\text{region}}) \tag{13}$$

where S_{ground} is the semantic segmentation of ground areas and T_{region} represents region-specific style features. Our texture generator Γ processes each semantic region in parallel, efficiently generating composite ground textures while preserving scene consistency.

4 Experiments

4.1 Dataset

CitySculpt Dataset To the best of our knowledge, there is currently no open-source dataset providing high-quality 3D city assets suitable for urban-scale generation. To address this gap, we constructed the CitySculpt dataset, consisting of approximately 5,000 high-quality 3D assets of urban elements, including buildings, vehicles, roads, street lights, and vegetation. These assets were collected from online open resources [1] and processed to ensure consistent quality. To enhance the dataset's versatility, we performed size normalization on all models and annotated each asset with semantic category, dimensions, and descriptive attributes of function and style. For each asset, we rendered 9 multi-view images from three viewpoint categories: 3 overhead satellite perspectives (0°-15° from vertical), 3 drone-view angles (30°-45° from vertical), and 3 ground-level views (75°-90° from vertical). These viewpoints are positioned at 120° intervals along circular trajectories at different elevations. More details about the CitySculpt dataset can be found in supplementary material.

OSM Dataset. Following CityDreamer [37], we utilize the OpenStreetMap (OSM) dataset[2] for additional testing and evaluation. This dataset comprises satellite imagery from 80 cities worldwide, with corresponding semantic maps that classify areas into five categories (roads, buildings, green lands, construction sites, and water areas) and height fields derived from OSM data. For each geographic location in the dataset, corresponding 3D city models can be accessed via Google Earth Studio[3], offering real-world references for qualitative comparison. This dataset serves as a benchmark for evaluating the city generation capabilities of our method.

4.2 Satellite-view Mesh Generation

Metrics. To evaluate our single-view mesh generation results, we assess both geometric and texture quality:

Geometric Quality: Following [41], we employ Chamfer Distance (CD) and Volume IoU between generated and ground-truth meshes. CD quantifies surface accuracy by measuring bidirectional point-wise distances, while Volume IoU assesses 3D spatial overlap, providing complementary insights into shape fidelity.

Texture Quality: For evaluating texture fidelity, we render the textured models to 2D images from multiple viewpoints and compute PSNR, SSIM[34], and LPIPS[43] against ground-truth renderings. These metrics collectively assess pixel-level accuracy, structural similarity, and perceptual quality.

Baselines. We compare CitySculpt against state-of-the-art conditional mesh generation methods: Craftsman [12], InstantMesh [39] and CRM [35]. All methods are retrained using our CitySculpt dataset to ensure fair comparison. For evaluation, we randomly select 300 building assets from our test set and use their satellite views as input.

Qualitative Comparison. Figure 3 presents a qualitative comparison against the baseline methods. Given the satellite perspectives as input, CRM struggles to generate accurate geometric structures of buildings, even though the texture generation is relatively accurate. InstantMesh successfully generates complete geometric structures with reasonably accurate texture colors, but occasionally produces inaccurate building shapes. We hypothesize this is due to the limited texture information available from the satellite view inputs. Craftsman produces accurate geometric structures with surface materials that closely resemble those used in real-world buildings. However it fails to generate coherent textures for elements such as windows and wall surfaces.

[1] https://sketchfab.com
[2] https://openstreetmap.org
[3] https://earth.google.com/studio/

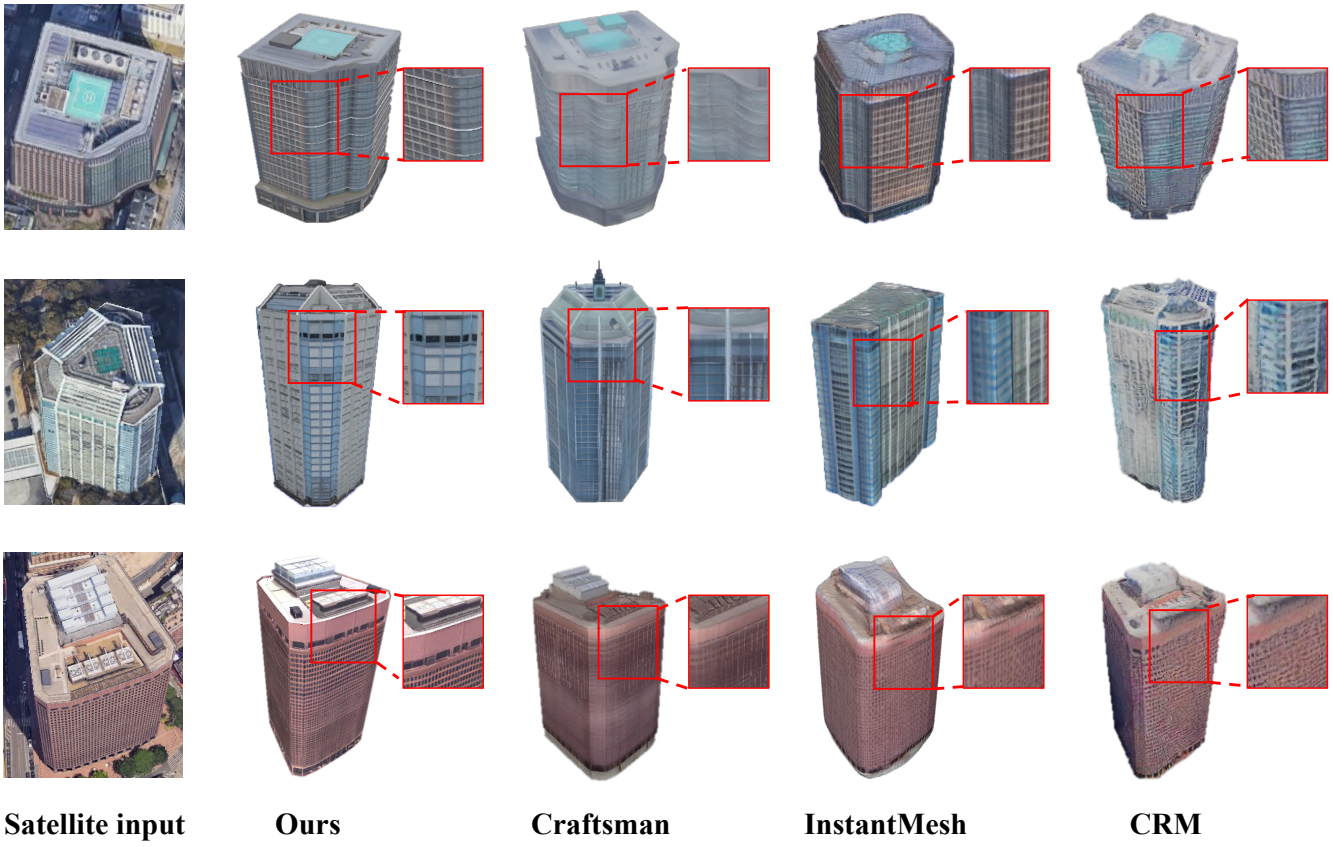

Figure 3: Qualitative comparison. Given satellite images from the real world, our method generates higher-quality buildings in both geometry and texture compared to baseline methods.

In contrast, our method accurately reconstructs building geometry while preserving key architectural details like balconies and windows. Regarding texture quality, our approach also generates textures that faithfully match the appearance of the satellite imagery, resulting in more photorealistic buildings.

Quantitative Comparison. Table 1 presents the quantitative metrics of the proposed approach compared to the baselines. Our method demonstrates significant improvements on geometric metrics, achieving the best performance on CD and Volume IoU. Moreover, we attain state-of-the-art results on visual metrics including PSNR and SSIM, demonstrating our approach's ability to generate high-fidelity building models with accurate textures. While our LPIPS score ranks second to InstantMesh, the comprehensive results across all metrics confirm our method's superior overall performance in generating both geometrically accurate and visually realistic building reconstructions.

4.3 City Generation

Metrics. We evaluate city-scale generation quality using complementary metrics that assess both geometric accuracy and visual quality:

Table 1: Quantitative comparison with baselines in building generation.

Methods	CD↓	Volume IoU↑	PSNR↑	SSIM↑	LPIPS↓
CRM	0.3063	0.3819	23.420	0.7636	0.2160
InstantMesh	0.2461	0.4430	24.852	0.8039	**0.1705**
Craftsman	0.2134	0.5821	22.965	0.8322	0.2204
Ours	**0.2016**	**0.6206**	**25.071**	**0.8710**	0.1793

Geometric Quality: Following CityDreamer [37], we employ Camera Error (CE) and Depth Error (DE) to evaluate the generated urban layouts. CE measures how accurately the generated city covers the reference distribution of building arrangements, while DE quantifies the diversity of generated structures compared to the reference dataset.

Texture Quality: For visual assessment, we compute Fréchet Inception Distance (FID) [9] and Kernel Inception Distance (KID) [1] between rendered frames and their ground truth images. These metrics measure both the distribution similarity and perceptual quality of our generated 3D assets. Additionally, we utilize no-reference image quality metrics BRISQUE [20] and NIQE [21] to evaluate the

Table 2: Quantitative comparison with baselines in city generation.

Method	CE ↓	DE ↓	FID ↓	KID ↓	NIQE ↓	BRISQUE ↓
CityDreamer	0.064	0.105	98.39	0.095	8.507	86.702
GaussianCity	0.061	0.093	87.03	0.089	7.687	75.340
Ours	**0.043**	**0.047**	**53.96**	**0.063**	**5.132**	**53.960**

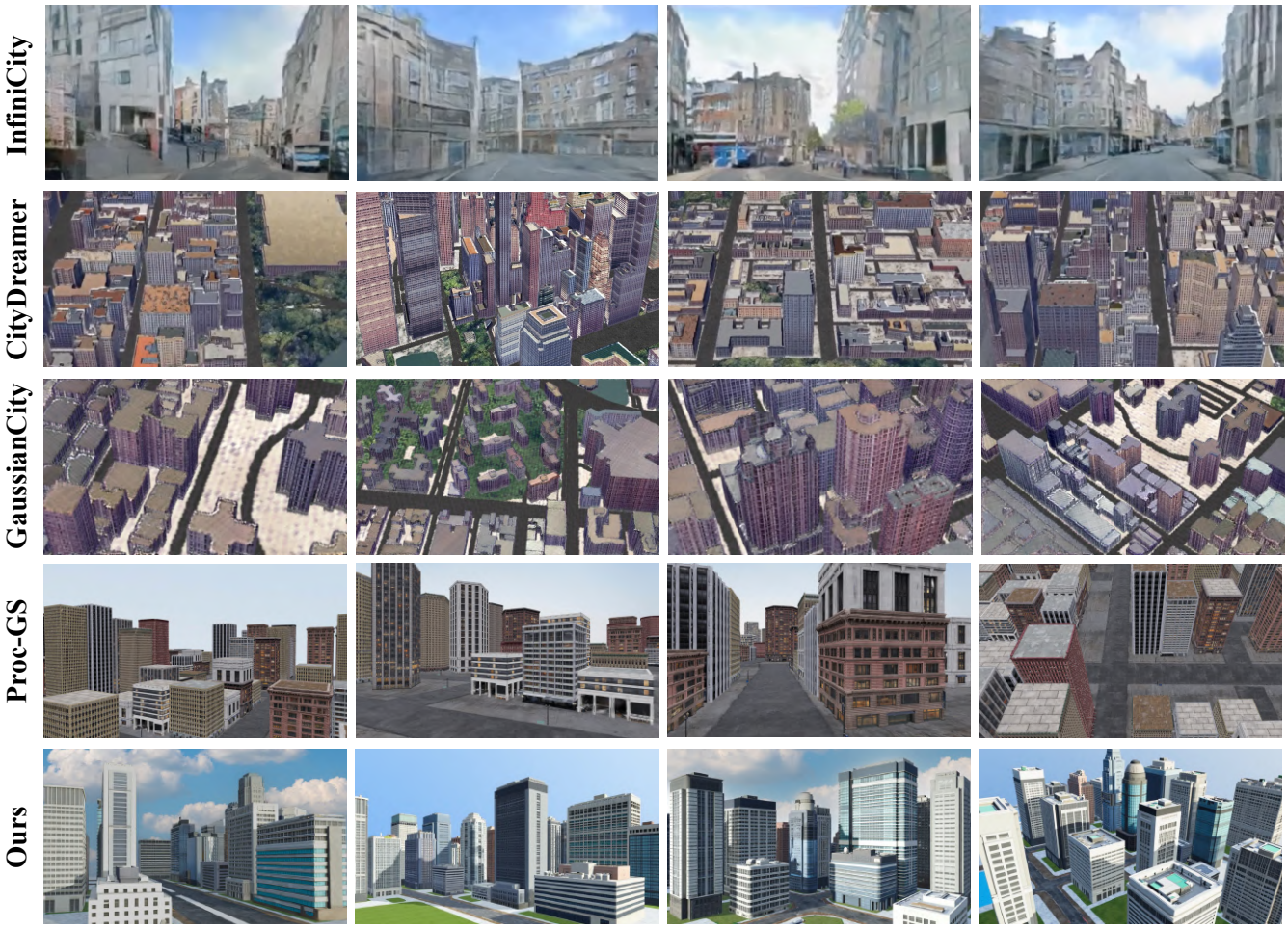

Figure 4: Qualitative comparison. Our method produces higher-quality 3D cities with better consistency compared with the baselines. We strongly recommend zooming in to examine the detailed differences.

perceptual quality of rendered frames, following standard practices in urban visualization evaluation.

Baselines. We compare CitySculpt against state-of-the-art city generation approaches including InfiniCity [16], CityDreamer [37], GaussianCity [38] and Proc-GS [13]. For evaluation, we randomly select 512 satellite images from OpenSatMap as test inputs. To ensure fair comparison, we process these images using each method's official released code with their default parameters. Each generated 3D city is then rendered from 40 different viewpoints to produce 20,480 result images for evaluation. For InfiniCity and Proc-GS we use the results provided by the authors as their code is not publicly available.

Qualitative Comparison. Figure 4 presents qualitative comparisons against several baselines. InfiniCity fails to generate complete structures, with the resulting buildings and roads showing shape defects and distortions. CityDreamer improves the overall scene integrity by directly generating textures on the 3D volume. However, the generated buildings are flat, lacking architectural details such as balconies and windows. GaussianCity enhances texture details using a Gaussian-based approach, but still suffers from the absence of essential geometric features. ProcGS reconstructs high-quality

buildings and assembles them into scenes, significantly improving visual quality. However, it lacks stylistic diversity, offering only a limited range of architectural styles.

In contrast, CitySculpt generates more realistic building structures, including windows, roofs, and other architectural details, avoiding the box-like shapes produced by baseline methods. Moreover, our method offers greater stylistic diversity with photorealistic textures that enhance visual fidelity across the generated cities.

Quantitative Comparison. Table 2 presents a quantitative comparison between our proposed method and the baseline methods. Our method shows significant improvements in both FID and KID, demonstrating superior texture generation quality. Additionally, CitySculpt achieves state-of-the-art results in NIQE and BRISQUE, further validating its effectiveness in generating high-quality and realistic textures. In terms of geometric evaluation, our method achieves the lowest DE and CE, proving its superior geometric accuracy and ability to generate detailed 3D structures.

4.4 Ablation Studies

To validate the effectiveness of our key components in city generation, we conduct ablation studies using 50 randomly selected satellite images from the OSM dataset.

Effectiveness of Multi-scale Scene Understanding. Multi-scale scene understanding helps leverage the limited information provided by satellite imagery by extracting multi-scale textual characteristics. To assess its impact, we compare our approach with two baselines: one that removes multi-scale scene understanding and only uses RGB images as input (RGB-Only), and another that extracts features from the entire satellite image without performing multi-scale analysis (Single-Scale). All other components remain unchanged during retraining.

The evaluation metrics include Camera Error (CE), Depth Error (DE), NIQE, and BRISQUE. The results are summarized in Table 3, showing the impact of multi-scale scene understanding on key metrics.

Table 3: Ablation study results for the effectiveness of multi-scale scene understanding in city generation.

Method	CE ↓	DE ↓	NIQE ↓	BRISQUE ↓
RGB-Only	0.074	0.112	6.324	72.564
Single-Scale	0.061	0.091	5.876	68.903
Ours	**0.043**	**0.047**	**5.132**	**53.960**

Effectiveness of Geometry Refinement. We utilize the UV normal diffusion framework to refine the geometry of buildings. To evaluate the effectiveness of our method, we compare our results with Craftsman, as it is one of the state-of-the-art methods in geometry refinement. We replaced our geometry refinement component with Craftsman while keeping all other components unchanged during retraining the city generation process.

Table 4 shows our method outperforms Craftsman across CE, DE, NIQE, and BRISQUE metrics, demonstrating the superior quality of our approach. Furthermore, we randomly select several building geometries for visualization. As shown in Figure 5, our method produces more detailed architectural features, including windows, balconies, and other architectural details.

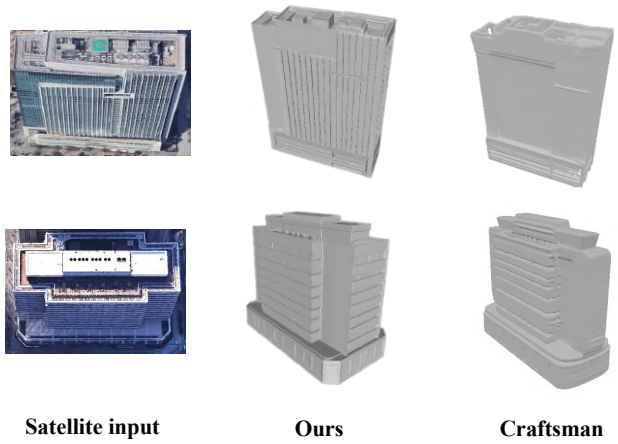

Figure 5: Qualitative comparison of geometry refinement between our method and Craftsman. Given the same input, our method generates richer architectural geometric details, including windows, balconies, and other structural elements.

Table 4: Quantitative comparison of geometry refinement between our method and Craftsman in city generation.

Method	CE ↓	DE ↓	NIQE ↓	BRISQUE ↓
Craftsman	0.049	0.053	6.245	78.234
Ours	**0.043**	**0.047**	**5.132**	**53.960**

Effectiveness of Texture Generation. Our texture generation approach is built upon the refined geometry. To evaluate its effectiveness, we perform experiments by removing the cross-attention mechanism between buildings (w/o cross-attention) and excluding all conditional information c_2 (w/o all condition). We evaluate the texture generation quality using the NIQE and BRISQUE metrics. Following DreamScene360 [47], we further employ CLIP Distance and Q-Align to assess the semantic alignment between the generated textures and the input imagery.

Table 5 shows that our method outperforms the baseline in all metrics, achieving superior texture quality and better alignment with real-world details.

Table 5: Quantitative comparison of texture generation quality.

Method	NIQE ↓	BRISQUE ↓	CLIP Dist ↓	Q-Align ↑
w/o condition	6.759	72.389	0.916	2.349
w/o cross-att	6.472	69.543	0.870	2.765
Ours	**5.132**	**53.960**	**0.802**	**3.410**

References

[1] Mikołaj Bińkowski, Danica J Sutherland, Michael Arbel, and Arthur Gretton. 2018. Demystifying mmd gans. *arXiv preprint arXiv:1801.01401* (2018).

[2] Yang Chen, Jingwen Chen, Yingwei Pan, Xinmei Tian, and Tao Mei. 2023. 3d creation at your fingertips: From text or image to 3d assets. In *Proceedings of the 31st ACM International Conference on Multimedia*. 9408–9410.

[3] Yuankun Chen, Dazhong Rong, and Yi Li. 2024. CrossViewDiff: A Cross-View Diffusion Model for Satellite-to-Ground Image Synthesis. In *International Conference on Artificial Neural Networks*. Springer, 287–302.

[4] Zhiqin Chen, Andrea Tagliasacchi, and Hao Zhang. 2020. Bsp-net: Generating compact meshes via binary space partitioning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 45–54.

[5] Jie Deng, Wenhao Chai, Jianshu Guo, Qixuan Huang, Wenhao Hu, Jenq-Neng Hwang, and Gaoang Wang. 2023. Citygen: Infinite and controllable 3d city layout generation. *arXiv preprint arXiv:2312.01508* (2023).

[6] Jie Deng, Wenhao Chai, Junsheng Huang, Zhonghan Zhao, Qixuan Huang, Mingyan Gao, Jianshu Guo, Shengyu Hao, Wenhao Hu, Jenq-Neng Hwang, et al. 2024. Citycraft: A real crafter for 3d city generation. *arXiv preprint arXiv:2406.04983* (2024).

[7] Simone Foti, Stefanos Zafeiriou, and Tolga Birdal. 2024. UV-free Texture Generation with Denoising and Geodesic Heat Diffusions. *Advances in Neural Information Processing Systems*.

[8] Wenyu Han, Congcong Wen, Lazarus Chok, Yan Liang Tan, Sheung Lung Chan, Hang Zhao, and Chen Feng. 2024. Autoencoding tree for city generation and applications. *ISPRS Journal of Photogrammetry and Remote Sensing* 208 (2024), 176–189.

[9] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. 2017. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *Advances in neural information processing systems* 30 (2017).

[10] Joon-Seok Kim, Hamdi Kavak, and Andrew Crooks. 2018. Procedural city generation beyond game development. *SIGSPATIAL Special* 10, 2 (2018), 34–41.

[11] Weijia Li, Jun He, Junyan Ye, Huaping Zhong, Zhimeng Zheng, Zilong Huang, Dahua Lin, and Conghui He. 2024. Crossviewdiff: A cross-view diffusion model for satellite-to-street view synthesis. *arXiv preprint arXiv:2408.14765* (2024).

[12] Weiyu Li, Jiarui Liu, Hongyu Yan, Rui Chen, Yixun Liang, Xuelin Chen, Ping Tan, and Xiaoxiao Long. 2024. CraftsMan3D: High-fidelity Mesh Generation with 3D Native Generation and Interactive Geometry Refiner.

[13] Yixuan Li, Xingjian Ran, Linning Xu, Tao Lu, Mulin Yu, Zhenzhi Wang, Yuanbo Xiangli, Dahua Lin, and Bo Dai. 2024. Proc-GS: Procedural building generation for city assembly with 3D Gaussians. *arXiv preprint arXiv:2412.07660* (2024).

[14] Zuoyue Li, Zhenqiang Li, Zhaopeng Cui, Marc Pollefeys, and Martin R Oswald. 2024. Sat2scene: 3d urban scene generation from satellite images with diffusion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 7141–7150.

[15] Zuoyue Li, Zhenqiang Li, Zhaopeng Cui, Rongjun Qin, Marc Pollefeys, and Martin R Oswald. 2021. Sat2vid: Street-view panoramic video synthesis from a single satellite image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 12436–12445.

[16] Chieh Hubert Lin, Hsin-Ying Lee, Willi Menapace, Menglei Chai, Aliaksandr Siarohin, Ming-Hsuan Yang, and Sergey Tulyakov. 2023. Infinicity: Infinite-scale city synthesis. In *Proceedings of the IEEE/CVF international conference on computer vision*. 22808–22818.

[17] 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).

[18] 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.

[19] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. 2019. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4460–4470.

[20] Anish Mittal, Anush Krishna Moorthy, and Alan Conrad Bovik. 2012. No-reference image quality assessment in the spatial domain. *IEEE Transactions on image processing* 21, 12 (2012), 4695–4708.

[21] Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. 2012. Making a "completely blind" image quality analyzer. *IEEE Signal processing letters* 20, 3 (2012), 209–212.

[22] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. 2020. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 3504–3515.

[23] Junyi Pan, Xiaoguang Han, Weikai Chen, Jiapeng Tang, and Kui Jia. 2019. Deep mesh reconstruction from single rgb images via topology modification networks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9964–9973.

[24] Ben Poole, Ajay Jain, Jonathan T Barron, and Ben Mildenhall. [n. d.]. DreamFusion: Text-to-3D using 2D Diffusion. In *The Eleventh International Conference on Learning Representations*.

[25] Sarun Poolkrajang and Anand Bhojan. 2024. Towards Generating 3D City Models with GAN and Computer Vision Methods.. In *VISIGRAPP (1): GRAPP, HUCAPP, IVAPP*. 211–219.

[26] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PmLR, 8748–8763.

[27] 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.

[28] Bingyu Shen, Boyang Li, and Walter J Scheirer. 2021. Automatic virtual 3d city generation for synthetic data collection. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*. 161–170.

[29] 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).

[30] 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).

[31] Yawar Siddiqui, Justus Thies, Fangchang Ma, Qi Shan, Matthias Nießner, and Angela Dai. 2022. Texturify: Generating textures on 3d shape surfaces. In *European Conference on Computer Vision*. Springer, 72–88.

[32] 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.

[33] Mikhail Tevelev, Danila Parygin, Timofey Kovalev, Anton Finogeev, and Alexey Churakov. 2024. Parametric Generation of Buildings and Structures Models Based on Data on Existing Infrastructure Objects. In *Novel & Intelligent Digital Systems Conferences*. Springer, 462–474.

[34] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13, 4 (2004), 600–612.

[35] Zhengyi Wang, Yikai Wang, Yifei Chen, Chendong Xiang, Shuo Chen, Dajiang Yu, Chongxuan Li, Hang Su, and Jun Zhu. 2024. Crm: Single image to 3d textured mesh with convolutional reconstruction model. In *European Conference on Computer Vision*. Springer, 57–74.

[36] Jianfeng Xiang, Zelong Lv, Sicheng Xu, Yu Deng, Ruicheng Wang, Bowen Zhang, Dong Chen, Xin Tong, and Jiaolong Yang. 2024. Structured 3D Latents for Scalable and Versatile 3D Generation. *arXiv preprint arXiv:2412.01506* (2024).

[37] Haozhe Xie, Zhaoxi Chen, Fangzhou Hong, and Ziwei Liu. 2024. Citydreamer: Compositional generative model of unbounded 3d cities. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 9666–9675.

[38] Haozhe Xie, Zhaoxi Chen, Fangzhou Hong, and Ziwei Liu. 2024. GaussianCity: Generative Gaussian splatting for unbounded 3D city generation. *arXiv preprint arXiv:2406.06526* (2024).

[39] Jiale Xu, Weihao Cheng, Yiming Gao, Xintao Wang, Shenghua Gao, and Ying Shan. 2024. Instantmesh: Efficient 3d mesh generation from a single image with sparse-view large reconstruction models. *arXiv preprint arXiv:2404.07191* (2024).

[40] Ningli Xu and Rongjun Qin. 2024. Geospecific View Generation Geometry-Context Aware High-Resolution Ground View Inference from Satellite Views. In *European Conference on Computer Vision*. Springer, 349–366.

[41] Haibo Yang, Yang Chen, Yingwei Pan, Ting Yao, Zhineng Chen, Chong-Wah Ngo, and Tao Mei. 2024. Hi3D: Pursuing High-Resolution Image-to-3D Generation with Video Diffusion Models. In *Proceedings of the 32nd ACM International Conference on Multimedia*. 6870–6879.

[42] Xin Yu, Peng Dai, Wenbo Li, Lan Ma, Zhengzhe Liu, and Xiaojuan Qi. 2023. Texture generation on 3d meshes with point-uv diffusion. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 4206–4216.

[43] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. 2018. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 586–595.

[44] Shougao Zhang, Mengqi Zhou, Yuxi Wang, Chuanchen Luo, Rongyu Wang, Yiwei Li, Zhaoxiang Zhang, and Junran Peng. 2024. Cityx: Controllable procedural content generation for unbounded 3d cities. *arXiv preprint arXiv:2407.17572* (2024).

[45] Linqi Zhou, Yilun Du, and Jiajun Wu. 2021. 3d shape generation and completion through point-voxel diffusion. In *Proceedings of the IEEE/CVF international conference on computer vision*. 5826–5835.

[46] Mengqi Zhou, Yuxi Wang, Jun Hou, Shougao Zhang, Yiwei Li, Chuanchen Luo, Junran Peng, and Zhaoxiang Zhang. 2024. SceneX: Procedural Controllable Large-scale Scene Generation. *arXiv preprint arXiv:2403.15698* (2024).

[47] Shijie Zhou, Zhiwen Fan, Dejia Xu, Haoran Chang, Pradyumna Chari, Tejas Bharadwaj, Suya You, Zhangyang Wang, and Achuta Kadambi. 2024. Dreamscene360: Unconstrained text-to-3d scene generation with panoramic gaussian splatting. In *European Conference on Computer Vision*. Springer, 324–342.