

GeoTexBuild: 3D Building Model Generation from Map Footprints

Ruihe Wang

Junyan Yang

Qiao Wang[✉]

March 2025

Southeast University

Abstract

We introduce GeoTexBuild, a modular generative framework for creating 3D building models from map footprints. The proposed framework employs a three-stage process comprising height map generation, geometry reconstruction, and appearance stylization, culminating in building models with intricate geometry and appearance attributes. By integrating customized ControlNet and Text2Mesh models, we explore effective methods for controlling both geometric and visual attributes during the generation process. By this, we eliminate the problem of structural variations behind a single facade photo of the existing 3D generation techniques. Experimental results at each stage validate the capability of GeoTexBuild to generate detailed and accurate building models from footprints derived from site planning or map designs. Our framework significantly reduces manual labor in modeling buildings and can offer inspiration for designers.

Keywords: 3D Building Generation, Modular Framework, Geometric Control, Appearance Stylization.

1 Introduction

Architectural models play a critical role in urban planning [68], tourism, video game design, film production [61], and virtual reality [79] applications. However, the intrinsic complexity of architectural structures means that automated and controllable modeling tailored to specific design requirements remains an area in need of further exploration [81]. On the one hand, manually constructing many architectural models using CAD software [5] is both time-consuming and labor-intensive. On the other hand, although 3D reconstruction technologies [26, 40, 20, 71, 39, 32, 50, 2, 48] have advanced substantially in recent years, these methods are constrained in several respects: they are unable to generate novel buildings that do not exist in reality, often necessitate extensive and costly data collection [73, 92, 85], and therefore are not well-suited for application during the early planning and design stages.

Accurate 3D building modeling requires more than a single facade image due to the inherent complexity of architectural geometric structures [44]. This complexity encompasses not just the facade but also the structural design, height, and proportionality of components, which are inadequately captured by street-view imagery or identical facade designs [57]. As a result, this limitation necessitates the exploration of features that can more effectively encapsulate and regulate the overall geometric characteristics of buildings.

Unfortunately, existing 3D generation techniques [60, 63, 41, 89, 4, 76, 9, 67, 72, 82, 29, 96, 53], relying on single photos or textual prompts, often lack the geometric precision necessary for detailed structures. These methods depend heavily on pre-trained models and fail to account for critical design-specific parameters in geometry, rendering them inadequate for nuanced architectural applications.

A natural feature identified on a map can function as an effective geometric control parameter: building footprints (in Fig. 1). In urban planning and game design, professional systems and personnel

¹R. Wang was with the School of Information Science and Engineering, Southeast University, Nanjing, Jiangsu, China (email: rz_wang@seu.edu.cn)

²J. Yang was with the School of Architecture, Southeast University, Nanjing, Jiangsu, China (email: yangjy_seu@163.com)

³Q. Wang was with both School of Information Science and Engineering and the School of Economics and Management, Southeast University, Nanjing, Jiangsu, China (Corresponding author, email:qiaowang@seu.edu.cn)

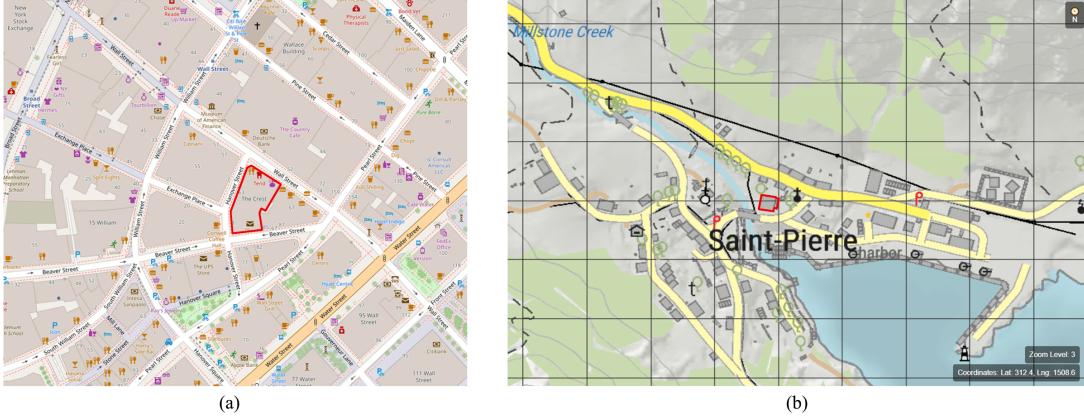


Figure 1: Building footprints in (a)real-world map on OpenStreetMap, and (b)fictitious map in game design. One of the footprints is highlighted in red to demonstrate its contour.

usually establish site planning [43] or map designs [49] beforehand, with these plans determining key parameters such as building location, floor area, and footprint shape. We aim to develop a generative approach that produces textured architectural models that are compliant with these designer-specified control conditions, effectively integrating predefined attributes with simple freehand sketches and textual prompts, thereby significantly reducing manual labor and offering inspiration for designers.

Table 1: Comparison of objectives and functions among building modeling methods.

Category	Method	Ref.	Forward Generation	Footprint Control	Roof Shape Control	Appearance	Supervision	Condition(Gen.)	Representation
Aerial photography	Yu et al.[13]		✗	○	✗	✗	Single 2D	-	Mesh
Grammar	Milde et al.[47]	✗	○	○	○	✗	Human knowledge	-	Wireframe(Graph)
	Storcz et al.[70]	✗	○	○	○	✗	Human knowledge	-	Voxel
Multi-view images	Dick et al.[15]	✗	○	○	○	✗	Multiple 2D	-	Mesh
	Schönberger et al.[65]	✗	○	○	○	○	Multiple 2D	-	Pcd
	Mildenhall et al.[48]	✗	○	○	○	○	Multiple 2D	-	NeRF
	Kerbl et al.[32]	✗	○	○	○	○	Multiple 2D	-	3DGS
	Li et al.[39]	✗	○	○	○	○	Multiple 2D	-	Mesh
	Debevec et al.[14]	✗	○	○	○	○	Multiple 2D	-	Mesh
Reconstruction	Nan et al.[52]	✗	○	○	○	✗	3D	-	Mesh
Point cloud	Paden et al.[56]	✗	○	○	○	✗	3D	-	Mesh
	Chen et al.[10]	✗	○	○	○	✗	3D	-	Mesh
	Wu et al.[83]	✗	○	○	○	✗	3D	-	Mesh
Procedural	Nishida et al.[54]	✓	✗	○	○	○	Single 2D	Image	Mesh
	Parish et al.[58]	✓	✓	✗	○	○	Human knowledge	Graph	Mesh
Lidar	Elakshner et al.[17]	✗	○	○	○	✗	3D	-	Wireframe(Graph)
	Huang et al.[27]	✗	○	○	○	✗	3D	-	Mesh
	Kada et al.[30]	✗	○	○	○	○	3D	-	Mesh
Satellite image	Partovi et al.[59]	✗	○	○	○	✗	Multispectral 2D	-	Mesh
Single image	Pang et al.[57]	✓	✗	✗	✗	✗	Single 2D	Image	Mesh
Image-to-3D	Wei et al.[81]	✓	✗	✗	✗	✗	3D	Image	Pcd
	Long et al.[41]	✓	✗	✗	✗	✓	3D	Image	Mesh
	Wu et al.[82]	✓	✗	✗	✗	✓	3D	Image	Mesh
Generation	Text-to-3D	Pool et al.[60]	✓	✗	✗	✓	Multiple 2D	Text	NeRF
	Raj et al.[63]	✓	✗	✗	✓	✓	Multiple 2D	Text	NeRF
	Chen et al.[9]	✓	✗	✗	✓	✓	Multiple 2D	Text	Mesh
	Yi et al.[89]	✓	✗	✗	✓	✓	Multiple 2D	Text	3DGS
	Bensadoun et al.[4]	✓	✗	✗	✓	✓	Multiple 2D	Text	Mesh
	Ukarapol et al.[76]	✓	✗	✗	✓	✓	Multiple 2D	Text	3DGS, Mesh
	Shi et al.[67]	✓	✗	✗	✓	✓	Multiple 2D	Text	Mv image, NeRF
	Nichol et al.[53]	✓	✗	✗	✓	✓	Multiple 2D	Text	Pcd
Text/Image-to-3D	Tang et al.[72]	✓	✗	✗	✓	✓	Multiple 2D	Text / image	3DGS
	Jun et al.[29]	✓	✗	✗	✓	✓	Multiple 2D	Text / image	Mesh
	Zhao et al.[96]	✓	✗	✗	✓	✓	Multiple 2D	Text / image	Mesh
Text, image-to-3D	Ours	✓	✓	✓	✓	✓	Single 2D, text	Image, text	Mesh

✗: incapable, ○: indirect control by extra data, ✓: capable.

To achieve this objective, we propose GeoTexBuild (Fig. 2), a generative framework of a three-stage strategy: height map generation, geometry reconstruction, and appearance stylization. In the initial step, we incorporate ControlNet [94] to combine information from building footprints and hand-drawn height maps, thereby enabling the controlled generation of roof structures. Subsequently, the overall geometry of the building is reconstructed from the generated roof structure along with specific input parameters, resulting in an untextured model. Finally, we employ the stylization functionality of Text2Mesh [46] to endow the model with detailed geometry and color into a fully textured building model. We compare the objectives and features of the proposed approach with other methods in Tab. 1.

In our experiment, we trained several customized ControlNets to investigate the impact of different image conditions on the control of roof structure generation. Additionally, we modified the Text2Mesh module to enhance the quality of stylizing for building generation. Extensive experiments from various perspectives substantiate the effectiveness of our framework in the generation of architectural models.

GeoTexBuild has the potential to bridge the gap between creative vision and technical execution,

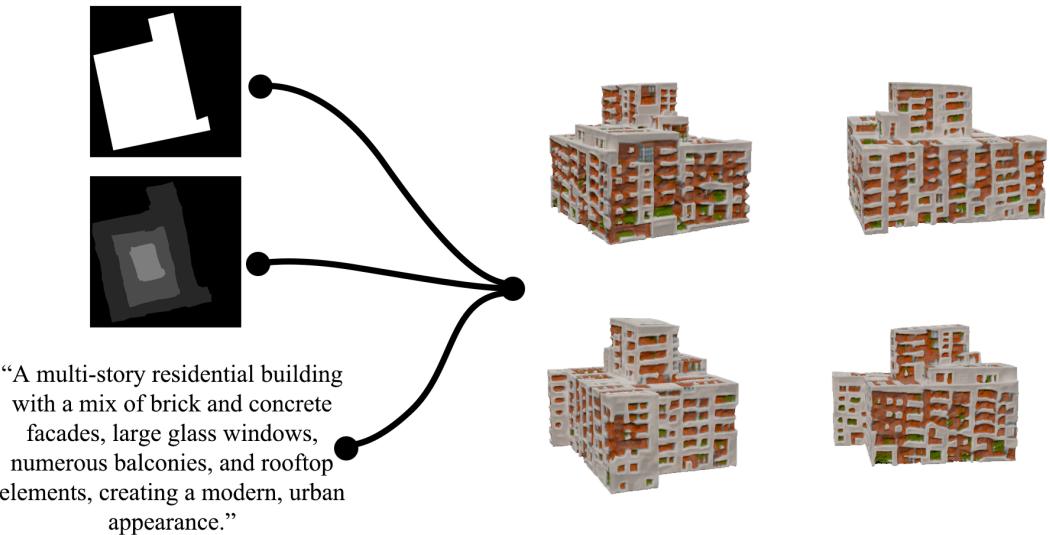


Figure 2: GeoTexBuild generates 3D textured building models from input footprint, height sketch, and text prompt.

enabling the effortless realization of highly customized architectural designs. Also, this approach could inspire advancements in adjacent fields, such as the combination of generative design for interiors and exteriors or automated urban aesthetics analytics for designing and planning. Modules in our modular framework can be replaced with newer pre-trained models, allowing us to leverage the latest and future advancements in the rapidly growing vision and language research, thus it can be a foundation for future innovation.

In summary, our contributions are as follows:

- We present a modular framework, GeoTexBuild, for generating building models from footprints with detailed controllable geometry and appearance attributes, thereby significantly reducing the required manual labor and offering inspiration for designers.
- We investigate how to control geometry attributes of building generation by separating the generation process into three phases and leveraging 2D spatial conditions in an orthogonal view.
- We train a customized ControlNet and a Text2Mesh module and integrate them into our framework for building generation, with a variant of the dataset Building3D and carefully selected architectural images.
- Experiments at each stage prove the effectiveness of our framework and fine-tuned models.

2 Related Work

2.1 Modeling Buildings from Reconstruction

In addition to manual modeling, the earliest methods for the automatic modeling of buildings include various approaches to reconstructing structures based on data obtained from reality or the expertise of professionals [42, 37, 1]. These approaches typically involve reconstructing buildings from satellite imagery [33, 59], aerial photography [8, 13], multiple photographs [15, 14, 16, 57, 65, 48, 32, 39], point cloud [52, 56, 84, 10, 23, 83], or lidar data [30, 17, 27]. Automatic modeling based on professional experience often employs procedural modeling techniques [54, 58] and methods for generating building grammars from collected data [47, 70].

In recent years, traditional methods have been integrated with deep learning or optimization techniques [8, 13, 57, 65, 52, 10, 83], either updating the computational logic within these methods or altering the data volume requirements. Furthermore, novel methods have emerged for general 3D reconstruction, extending beyond just buildings [7, 18, 19]. These innovative approaches [26, 20, 39, 32, 50, 2, 48] redefine 3D data representation or introduce new reconstruction algorithms that enhance both the efficiency and fidelity of the reconstruction. They are capable of concurrently restoring geometry and color to a photorealistic level.

These reconstruction methods have advanced our comprehension of modeling and generation processes, and have established both methodological frameworks and data foundations for generation.

2.2 3D Generation

There are two primary categories of 3D generation techniques: one emphasizing the generation of individual objects, and the other focusing on scene (background) generation [36, 28, 55, 93, 90]. Generative approaches for modeling 3D urban environments [98, 35, 87, 66] also exist, but their focus typically excludes the detailed geometry of a single building. Thus, this section predominantly centers on techniques for single-object generation.

The development of 3D generation technologies, which originated from reconstruction tasks, has accelerated significantly recently. Rather than aiming at generating specific objects, these technologies can generate various objects determined by their datasets. Some 3D generation models are capable of producing only shapes [86], while others can simultaneously generate both shapes and appearances [38].

Based on inputs, there are two mainstreams of 3D generation models: text-to-3D [34] and image-to-3D [38]. Based on their 3D representation, these models can be further classified into explicit representations (such as point cloud-based [53, 81, 97], mesh-based [69, 95], voxel-based [64], and 3DGS-based [89, 76]), implicit representations (such as NeRF-based [60, 63] and neural implicit surface-based [29]), and hybrid representations (such as latent-based [96, 45], triplane-based [25, 74], and multi-layer representation-based [90]).

Regarding generation strategies, models may employ multi-view optimization-based [72, 67, 76, 89, 63, 60], feedforward [41, 4, 82, 25, 74, 95, 88], or procedural approaches [22]. Based on training data, they can be divided into models trained on 3D data [4, 82, 25, 74, 95, 88], those trained on multi-view 2D images [72, 76, 89, 63, 60], and those trained on single-view images [81]. These models exhibit variation in terms of complexity, training cost, required data volume, generation quality, controllability, and ease of application.

2.3 3D Model Editing and Stylization

After the geometry generation stage, it is necessary to incorporate appearance into the 3D model. In other instances, modifications to the appearance of an existing geometric model are required. Such modifications are achieved through editing and appearance stylization techniques. Certain approaches focus solely on altering the visual attributes of 3D models, for example, by generating textures and adjusting color properties [12, 91, 3], whereas others emphasize the refinement of local geometric features [80]. Given the interdependence of geometry and appearance, some methods are designed to modify both local geometry and color attributes concurrently [46, 21, 77].

3 Approach

3.1 Formulation

We target the generation of 3D building models with detailed appearance conditioned by a footprint, a hint of roof structure, and a description of building style. Considering the trade-off between feasibility, control accuracy, and convenience, we input the footprint and roof structure as images, namely the mask and height sketch, and the style description as text. To this end, we propose GeoTexBuild, a modular framework that allows easy and accurate control and produces stylized 3D models in various building patterns.

To have a better understanding of our framework, we first introduce two important external modules in section 3.2, then give an overview of our pipeline in 3.3, and details of each component in 3.4.

3.2 Preliminaries

3.2.1 ControlNet

ControlNet [94] is a kind of powerful neural network architecture to guide text-to-image diffusion models [24] with spatially localized, task-specific image conditions. It attaches an additional trainable copy of the diffusion model to the backbone to inject the spatial information of the desired condition. Here, we give a brief review of its principles.

First, the spatial condition image c_i is passed to a tiny network $E(\cdot)$, to be encoded to feature space, as

$$c_f = E(c_i). \quad (1)$$

This vector is later used as an important input. Then, ControlNet implements several ingenious modifications to the original diffusion model. A pre-trained neural block $F(\cdot, \theta)$ with parameters theta, transforming input feature map x into output y can be written as

$$y = F(x; \theta). \quad (2)$$

The ControlNet freezes the parameters theta of the original block and adds a trainable clone with new parameters θ_c simultaneously. The added block takes the external conditioning vector c_f as input and is connected to the locked model with zero convolution layers, denoted as $Z(\cdot; \cdot)$. For a single neural block F , the complete ControlNet then output

$$y_c = F(x; \theta) + Z(F(x + Z(c_f; \theta_{z1}); \theta_c); \theta_{z2}), \quad (3)$$

where the number in the subscript indicates which layer and type the parameter belongs to.

For the whole Stable Diffusion model, multiple ControlNet blocks are inserted to achieve fine control of different scales on 'global' and 'detailed' contexts. The trainable copies are applied to each encoder level of the diffusion U-net, and the outputs are added to the skip connections and the middle block of the U-net. The ControlNet is a computationally efficient architecture since only the trainable copies need gradient backpropagation when fine-tuning, enabling high-speed, low memory consumption single-GPU training.

ControlNet allows multiple control methods, such as various kinds of edge maps, segmentation maps, depth maps, masks, and even skeleton poses. It also supports tandem or simultaneous controls on different aspects of the image, like shape and color. Thanks to the great convenience brought by ControlNet, we were able to implement fine control over not only the bottom shape from the footprint, but also the height and detailed structure of the roof from arbitrary sketches. This also helps us achieve low-cost training from only 2D conditions compared to other 3D generation models with 3D training data.

3.2.2 Text2Mesh

Text2Mesh [46] is a method for stylizing a given coarse mesh through natural language or image conditioning. This module considers global content as a large structure prescribed by the given 3D mesh, which defines the overall shape and topology, and style as the particular appearance, as determined by its color and fine-grained geometric details on the surface. It accomplishes mesh style editing with three main components: a neural style field (NSF), a differentiable render, and a similarity comparator.

The input mesh \mathcal{M} with vertices $\mathcal{V} \in \mathbb{R}^{n \times 3}$ and faces $\mathcal{F} \in \{1, \dots, n\}^{m \times 3}$ is first normalized to lie inside a unit bounding box and fixed throughout the whole operation as the reference global content. The neural style field is represented as three MLPs: N_s , N_c , and N_d , which map a point p in the unit space to a vector and a scalar, as

$$(c_p, d_p) = NSF(\gamma(p)) \in (\mathbb{R}^3, \mathbb{R}), \quad (4)$$

where $\gamma(\cdot)$ is the positional encoding operator. When the point is in the set of vertices of the mesh, the values output by the NSF are valid and passed to the mesh as vertex color c_p and displacement $d_p \cdot \vec{n}_p$ along the surface normal. In practice, the query points in the field are set to be vertices in \mathcal{V} , and they are then visualized by a differentiable renderer over the given triangulation. Text2Mesh renders $n_\theta = 5$ views of the modified mesh in each iteration and implements multiple 2D augmentations to the rendered image to enhance the details of the style modification.

Then, those augmented images are passed into the similarity comparator to compute the semantic loss. The core component of the comparator is a pre-trained CLIP model [62] that acts in a multi-modal

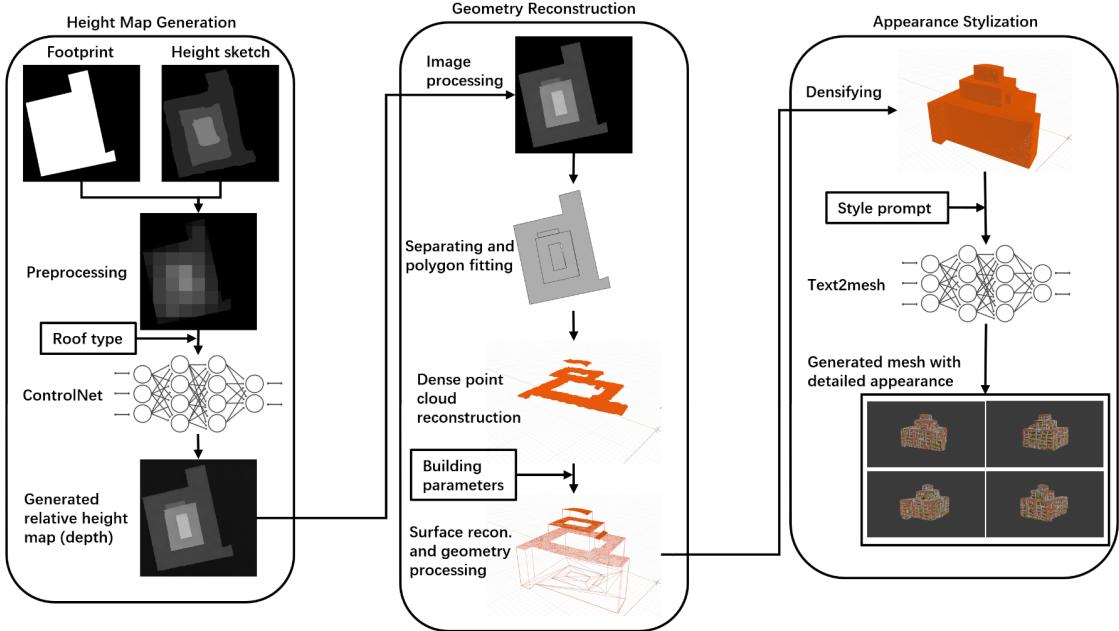


Figure 3: Illustration of our pipeline. Our framework has three main function blocks: height map generation, geometry reconstruction, and appearance stylization.

embedding space. The textual description t of the desired style is first passed into the text encoder, as $\phi_{target} = E_t(t) \in \mathbb{R}^{512}$. Then, the images follow, as $S^{type} = 1/n_\theta \sum_\theta (E_i(\psi_{type}(I_\theta))) \in \mathbb{R}^{512}$, where ψ_{type} is the 2D augmentation operator of the corresponding type. The loss is then:

$$L_{sim} = - \sum_S sim(S, \phi_{target}), \quad (5)$$

where $sim(a, b) = (a \cdot b) / (|a| \cdot |b|)$ is the cosine similarity between a and b . The loss backpropagates through the differentiable renderer and then updates the MLPs in the neural style field. When minimizing the loss, the similarity between the rendered image and the prompt increases, meaning the appearance of the mesh gradually approaches the textural description.

The Text2Mesh module allows us to modify detailed geometry and color attributes on the surface of the mesh, therefore, in our building model generation framework, it is used to refine the coarse mesh to give a fine appearance that matches the prompt words.

3.3 Overview of Pipeline

We show an illustration of our framework in Fig. 3. There are three main function blocks in our framework: height map generation, geometry reconstruction, and appearance stylization. We start with a given footprint and a height sketch drawn by hand. The ControlNet in the height map generator will output a regular and smooth height map, ready for geometry reconstruction. The next block will first process the generated image to remove the noise brought by the diffusion model, then separate each discontinuous part in the height map and fit their shapes to polygons. After that, we reconstruct a dense point cloud of the roof structure according to the pixel value on the height map and further build surfaces based on the point cloud. More surfaces, such as facades, are built according to input building parameters. At the end of geometry reconstruction, we process the coarse mesh to remove small holes and cracks, making the surface smooth and clean. Finally, we stylize the densified building mesh with the Text2Mesh module with an input text prompt. The output mesh is saved in a standard 3D data format for easy downstream applications or further manual modification.

3.4 Components of GeoTexBuild

3.4.1 Height Map Generation

Our height map generation block is based on a variant of ControlNet, which transfers a rectangular color palette to a regular colored image. The height sketch is the main control that determines the generated height map, and it can be easily drawn by hand in a few seconds. In order to be suitable for building generation, we have the following requirements: 1) The pixel value on the image only indicates the height of the corresponding point on the roof, thus, the input and output should be converted to grayscale, namely a depth map. 2) In order to constrain the bottom of the generated building to match the footprint exactly, the latter should be an input in a certain form. 3) To perform fine-grained control of roof type and shape, we should be able to use different palette grid sizes and a prompt that indicates roof types in several fixed patterns.

To this end, we define the height sketch as a grayscale image whose values represent the relative height of the structures on the roof, with a larger value (brighter) meaning higher. Relative means they are not absolute values, and can be shifted or scaled in later operations. Different from most ControlNets that use line edges to control generated shapes, we decided to input the footprint as a mask I_m in binary to achieve a stronger control with less confusion.

Preprocessing. We first process the sketch to a square brightness palette with a grid size n_g ,

$$I_p^{i,j} = \frac{1}{w_p^2} \sum_{\substack{(i-1)w_p < x \leq iw_p \\ (j-1)w_p < y \leq jw_p}} I_s(x, y), \quad (6)$$

where $w_p = w_s/n_g$ is the width of the palette grid, w_s is the width (equals to height) of the sketch image, $I_p^{i,j}$ is the pixel value of the square in the i th row and j th column. The brightness palette is then masked by the footprint to create a clear boundary, as

$$I_{mp} = I_p \odot I_m, \quad (7)$$

where \odot is the Hadamard product of matrices. Then, I_{mp} is the input of our fine-tuned brightness palette ControlNet for height map generation.

Applying ControlNet. To assist ControlNet in understanding different possible types of roofs from a single brightness palette, we construct several fixed patterns of prompt text t , describing whether the roof has multiple parts and its general shape. The prompts are listed in Tab. 2. Then, the generating process can be represented as

$$I_h = CN(I_{mp}, t). \quad (8)$$

In order to learn the map between the masked brightness palette image and the relative height map of roofs, we trained the ControlNet model with a carefully selected dataset of roof reconstructions of real-life architectures. The dataset and training details will be explained in section 4.1.

3.4.2 Geometry Reconstruction

Image processing. Once we obtain the relative height map, we can start to reconstruct the geometry of the building. First, as the relative height map is generated by a ControlNet model with zero convolution, it is inevitable that there will be some burst noise near the edges of the controls. Other noise will also exist in plain areas due to the nature of diffusion models. We filter the image with a bilateral filter [75] to smooth the height value while preserving the edges of different parts, as

$$I_{bf}(x) = \frac{1}{W(x)} \sum_{x' \in \Omega} I_h(x') \cdot g_r(\|x - x'\|) \cdot g_s(\|x - x'\|), \quad (9)$$

where $W(x)$ is the normalization factor, Ω is neighborhood of x , g_r and g_s are Gaussian kernels. To ensure sufficient smoothing, the filter size is set to 7 with $\sigma_{gr} = 9, \sigma_{gs} = 55$. We erode the mask by one pixel with morphological operator $\mathcal{E}(\cdot)$ to further remove bright spots near the edges,

$$I_{bfe} = I_{bf} \odot \mathcal{E}(I_m). \quad (10)$$

Separating and polygon fitting. In the processed image, we fit each color zone to a polygon with Vtracer, one of the state-of-the-art raster-to-vector graphics converters. In this module, the pixels

are clustered by the Kopelman Algorithm, then a binary tree is built bottom-up from the clusters for hierarchically laying them on the canvas. The clustering is based on differences in brightness as we set its color precision to 6, meaning a cluster has to have at least a color difference of $256/6 \approx 42$ to be considered a separate cluster. Lower or higher precision will lead to color zones missing or unnecessary holes in polygons. Those clusters will go through path walking, path simplification, smoothing, and curve fitting to be vectorized into polygons. We extract each polygon as an independent roof part from the output vector graph of Vtracer. With those vectorized polygons, we obtain the position of the vertices in the exterior $\{\partial\mathcal{S}_r\}$ of the reconstructed roofs $\{\mathcal{S}_r\}$.

Lifting image to point cloud. We reconstruct a dense point cloud from the height map pixels encircled by each vectorized polygon. The positions and height values of inner pixels are set directly as the (x, y) , and z -coordinates of the points. The point cloud is cleaned by removing statistical outliers and noisy points whose z value is lower than an input height limit h_{min} .

Reconstructing surfaces. Each part of the point cloud is sent to a normal estimator and a Poisson surface reconstruction [31] with octree depth set to 6, after which fairing is applied according to the roof type. Because we reconstruct on an open-boundary point cloud with normals in similar orientations, the reconstructed roof surfaces $\{\mathcal{S}_r\}$ will not be enclosed. This brings us an advantage and a disadvantage. The advantage is that we can adjust the ratio between the absolute building height and the roof’s height by multiplying the factor from the input building parameters. The disadvantage is that we find direct trimming after reconstruction will destroy the smooth boundary we try to preserve, leading to stripy facades. To address this, we take advantage of Boolean operations. We lift the surfaces up to the desired building height and extrude them top-down to ground for constructing meshes $\{\mathcal{M}_r\}$ of each roof part. Meanwhile, prisms $\{\mathcal{M}_{\partial\mathcal{S}_r}\}$ extruded bottom-up from $\{\partial\mathcal{S}_r\}$ are built. Then, the final mesh of the reconstruction is

$$\mathcal{M}_b = \cup(\mathcal{D}(\{\mathcal{M}_r\} \cap \{\mathcal{M}_{\partial\mathcal{S}_r}\})), \quad (11)$$

where \cap refers to the intersection of corresponding meshes in two sets, and morphological operator $\mathcal{D}(\cdot)$ refers to 1 unit dilation in x, y directions for each object. The dilation compensates for the erosion applied before and eliminates holes or cracks caused by polygon fitting.

3.4.3 Appearance Stylization

From geometry reconstruction, we obtain a mesh with clean surfaces without any fine structure or color. We enrich its appearance to make it similar to reality in this function block. The mesh \mathcal{M}_b is first remeshed to densify triangles on the surfaces to meet the resolution requirement of Text2Mesh. We set the octree depth to 8 to ensure that the vertices can accurately portray the color and displacement. The remeshed model and input prompt are passed to the Text2Mesh module with a fine-tuned CLIP.

We make a tiny adjustment in the renderer of Text2Mesh. In the original settings, n_θ views are rendered in a normal distribution around the front view. However, although it works well for small objects in the original paper, these views do not cover the surface of the building mesh evenly. We modify the distribution of azimuth angles to a uniform distribution in $[0, 2\pi)$ and the distribution of elevation angles to a normal distribution with variance $3\sigma = \pi/4$.

4 Experiments

To complete the whole generation process, we trained a customized ControlNet model on special renderings from the Building3D dataset for height map generation and fine-tuned a CLIP model on high-quality hand-selected images for appearance stylization. The details are in sections 4.1 and 4.3. The experiments on geometry reconstruction are in section 4.2, and the results and analysis of the complete pipeline are in section 4.4.

4.1 Customizing ControlNet

4.1.1 Dataset

In this module, we try to learn the mapping from a footprint with brightness conditions to a height map, thus, we need to create corresponding paired data. Open-source maps and city data, such as OpenStreetMap, often do not contain building height data with sufficient accuracy to show the height variation inside a house’s roof. We turned to city-scale building reconstruction data and extracted height information from their roofs. Luckily, Building3D [78] (Fig. 4) offers over 32k building reconstructions

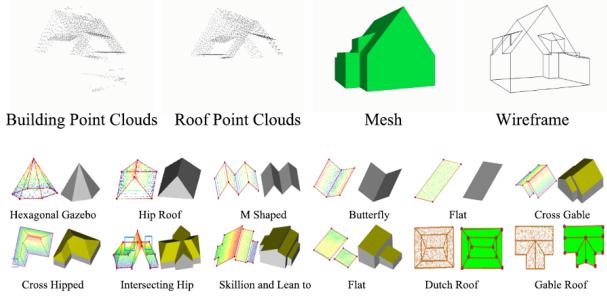


Figure 4: Data examples from the Building3D dataset. The dataset offers over 32k building reconstructions in multiple data forms. We rendered depth maps from surface-filled wireframes.

in multiple data forms in point cloud, wireframe, and mesh. We chose wireframes because it is accurate enough without being as noisy as naive point clouds. We inserted surfaces in the wireframes and rendered orthogonal top views in Blender; the depth maps and masks were used as paired data in ControlNet training. We augmented the dataset $8\times$ by rotation, flipping, and random cropping.

4.1.2 Training

When training, the rendered depth maps were used both as inputs and the ground truths. As inputs, they went through the preprocessing described in section 3.4.1, which outputs a masked brightness palette image. We set n_g a random integer from 5 to 9 to enable conditioning in different precisions. As ground truths, the depth maps are used for computing the loss of the noise predictor in the diffusion process.

Why can part of the input be the same as the output target? Is it only reconstructing images? This only happens in training. Recap that we target generating from hand-drawn sketches, but the reconstructed buildings from the real world do not have paired height sketches. Thanks to the preprocessing that fills the gap between the rendered depth and the hand-drawn sketches, we can train the model in a reconstruction way, but perform inference in a generation way.

To prepare prompts for different roof types, we classified the roofs based on the top structure and geometry of the wireframes. The wireframes were regarded as graphs, and divided into single-piece or multi-piece on their connectivity, simple or complex on the degrees of nodes, and pitched or flat on the range of z-coordinates of nodes. We construct simple patterns of prompts for each category shown in Tab. 2.

Table 2: Prompts for Different Roof Types

Connectivity	Node Degree	Z-Coordinate Range	Prompt
Multi-Piece	> 4(complex)	Pitched	Grayscale depth-map, multiple parts of complex slopes and layers, in polygon shape, black background
Multi-Piece	/	Flat	Grayscale depth-map, multiple flat layers, in polygon shape, black background
Multi-Piece	< 3(simple)	Pitched	Grayscale depth-map, multiple simple shed layers, in polygon shape, black background
Multi-Piece	=3, 4(medium)	Pitched	Grayscale depth map, multiple parts of slopes and layers, in polygon shape, black background
Single-Piece	> 4(complex)	Pitched	Grayscale depth map, a joint part of complex combinations of slopes, in polygon shape, black background
Single-Piece	/	Flat	Grayscale depth map, a flat layer, in polygon shape, black background
Single-Piece	< 3(simple)	Pitched	Grayscale depth map, a simple shed layer, in polygon shape, black background
Single-Piece	=3, 4(medium)	Pitched	Grayscale depth map, a joint part of simple slopes, in polygon shape, black background

We trained a custom ControlNet from the Stable Diffusion 1.5 base model on 260k paired images up to 32618 global steps. The batch size was set to 4 with gradient accumulation steps set to 4. The SD decoder was unlocked during the last 1000 steps with a learning rate change from 1e-5 to 2e-6, following the instructions of the ControlNet author. The entire training process was completed with a single Nvidia RTX 3090 GPU in 62 hours.

4.1.3 Comparing different control combinations

We compare different control combinations from a footprint to a height map here to demonstrate the superiority brought by the combination of mask and brightness palette.

Here are four models trained with the same hyperparameters in Fig. 5, but input different controls: (1)line edges only, (2)line edges and brightness palette, (3)masks only, (4)masks and brightness palette. The result shows that, first, in (1)(3), the height is generated randomly without the brightness palette; second, in (1), the line edges cause ambiguity in distinguishing the internal and external. As for (2),

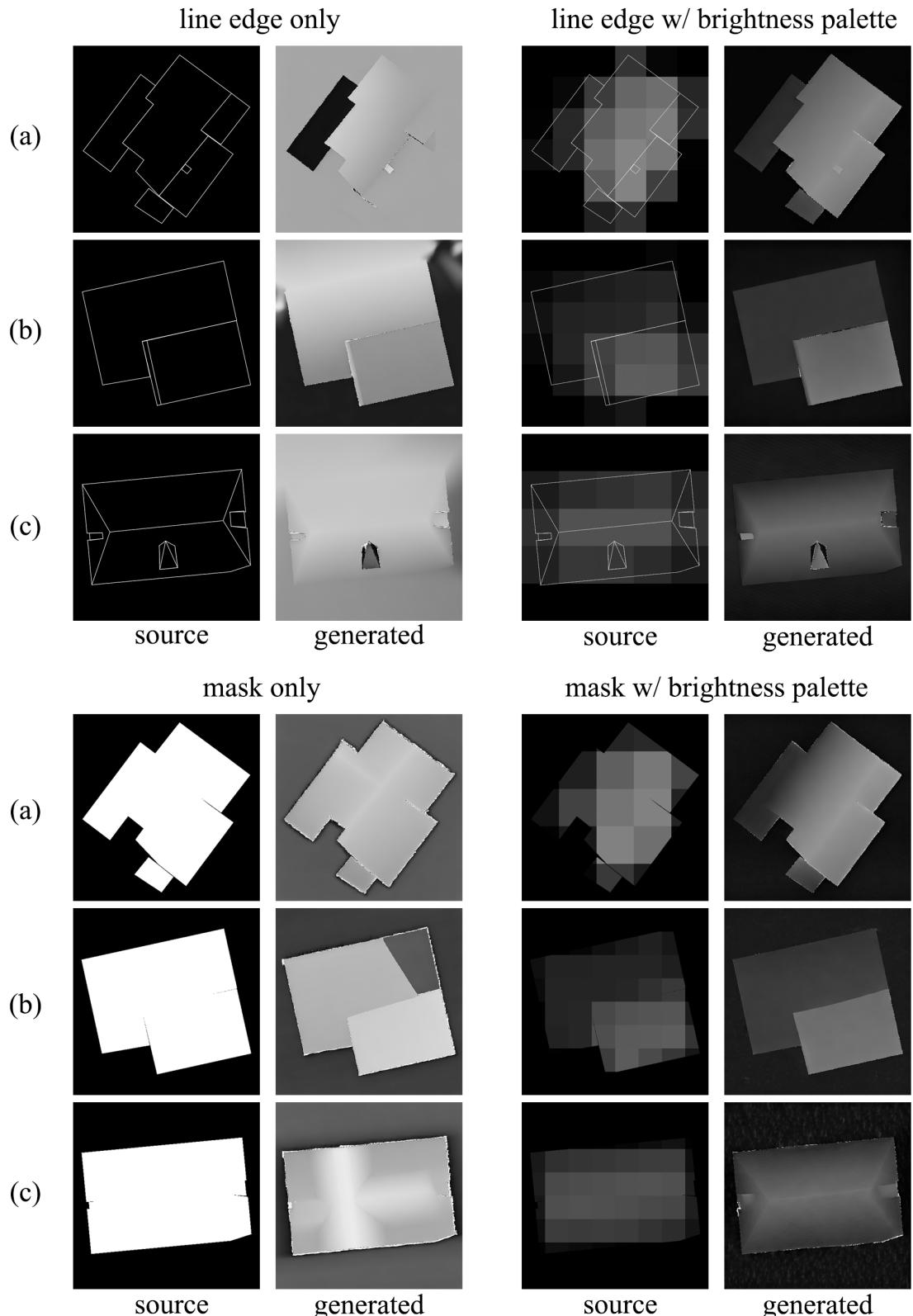


Figure 5: Four customized ControlNet models that input different geometric controls are trained with the same hyperparameters. The corresponding prompts used when inference is "grayscale depth map, a joint part of simple slopes, in polygon shape, black background" for (a)(b), and "grayscale depth map, a joint part of complex combinations of slopes, in polygon shape, black background" for (c).

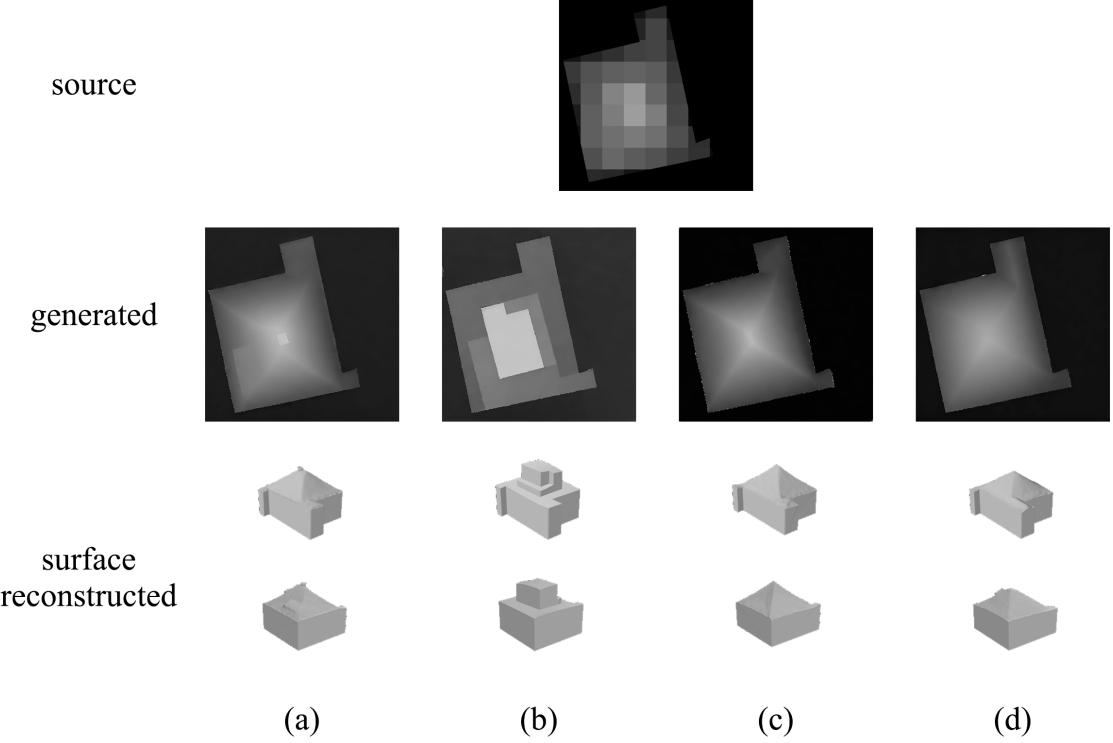


Figure 6: By changing different prompts when inputting the same control, we demonstrate that the model can understand different possible types of roofs from a single brightness palette. The corresponding prompts are "grayscale depth-map, multiple parts of complex slopes and layers, in polygon shape, black background" for (a), "grayscale depth-map, multiple flat layers, in polygon shape, black background" for (b), "grayscale depth map, a joint part of complex combinations of slopes, in polygon shape, black background" for (c), "grayscale depth map, a joint part of simple slopes, in polygon shape, black background" for (d). The reconstructed surfaces are also shown. The prompt in (a) indicates more layers, such as the structure of chimneys, dormer windows, and soffits. The results of (c) and (d) are similar due to the similarity of their prompts.

drawing line edges and height sketches simultaneously needs extra consideration of their compatibility, but shows similar quality compared to (4). Although (4) has lost some very small and delicate details, it generates less noise at the edges of the inner parts, which is difficult to filter.

Additionally, we demonstrate that by altering prompts for roof types, the model can produce corresponding height maps as in Fig. 6.

4.2 Geometry Reconstruction

In this component, there are mainly image and geometry operations without any trainable neural blocks. We built them with Easy3D [51] and Blender. Here, we demonstrate some comparisons of the ablation study.

w/o image processing. The reconstructed surfaces are shown in the second part of Fig. 7. The image processing stage helps to remove noise and keep a sharp boundary of the reconstructed roof parts.

w/o separating and polygon fitting. The reconstructed surfaces are shown in the third part of Fig. 7. In the absence of the separation and polygon fitting stage, various roof parts merge, resulting in over-smooth surfaces or bread-like boundary artifacts in Poisson reconstruction. While the impact on a single-layer roof is relatively minor, it becomes catastrophic when applied to a multi-layer roof in the second column.

w/o boolean operations. The reconstructed surfaces are shown in the fourth part of Fig. 7. Replacing the boolean operations with simple trimming will destroy the smooth boundary we try to preserve, leading to stripy facades.

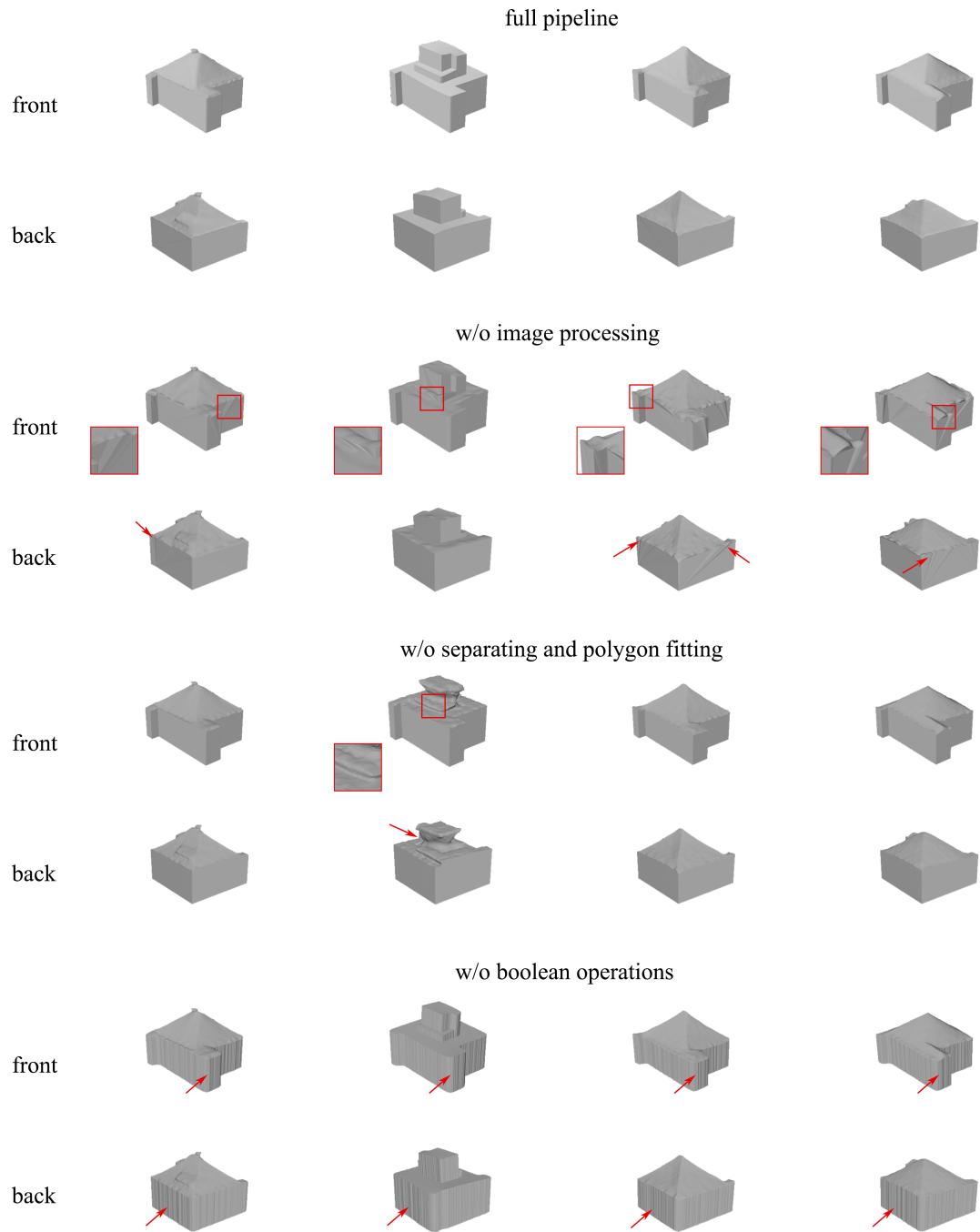


Figure 7: Results of ablation studies for geometry reconstruction. Please note the irregular shapes and uneven edges in the red boxes or pointed out by arrows, which are direct consequences of omitting crucial steps.

4.3 Customizing Text2Mesh

When we tried to use the default Text2Mesh with ViT-B/32 CLIP model to stylize the appearance, we found that the model often could not correctly distinguish the semantic parts of the image. In serious cases, it would even confuse the background (such as courtyards, landscapes, or trees) with the foreground (buildings), thus drawing the background on the target. Since the main guidance of the generation process is the similarity comparator, namely the CLIP model, we decided to fine-tune it using manually selected high-quality building images.

4.3.1 Data

To address the problem of background confusion, we eliminated the background of the images used for fine-tuning, leaving only the foreground buildings. In the renderer of Text2Mesh, the background is set to white with a mask, thus we filled the white color in images in the training data accordingly. We selected over 800 images from the Internet, covering various types of buildings with different view angles. The corresponding text descriptions of the style are generated by GPT-4 [6] with the request "Give me a prompt of the building in the uploaded image, describing its shape, design, material, color, and other features, no more than 30 words."

4.3.2 Training

We fine-tuned the Vit-B/32 CLIP model from the 'laion2b_s34b_b79k' checkpoint with the OpenCLIP [11] code base. The model was trained for 16 epochs with a batch size of 64. This was done with a single RTX 3090 in 20 minutes.

4.3.3 Comparison

We demonstrate the qualitative comparison between the fine-tuned model with our render viewpoint adjustment and the default Text2Mesh in the first two lines of Fig. 8 and quantitative comparison in Tab. 3. The default model tends to give the entire building a repeating texture or treat other backgrounds (such as greenery) as part of the building, while the fine-tuned model can generate regular building surfaces and details.

4.4 Results

Qualitative results are shown in Fig. 9. We display both the textured meshes and the difference between their top-down projection and the footprint in Fig. 10. Quantitative results of the CLIP similarity scores between the style prompt and rendered views are shown in Tab. 3. The numbers in the table are the average score of four fixed camera views as displayed in Fig. 8. The CLIP similarity is calculated as

$$sim_{CLIP} = \langle E_i(I), E_t(t) \rangle, \quad (12)$$

which is the inner product of the encoded image vector and the encoded text vector from CLIP encoders.

Table 3: CLIP similarity scores between the generated models and prompts

Prompts	1	2	3	4	5	Total avg.	GPU hours
Ours	0.3164	0.2934	0.2945	0.2845	0.3113	0.3000	63(RTX3090)
w/o finetune	0.2898	0.1669	0.2912	0.2612	0.2606	0.2540	/
GradeADreamer	0.2759	0.1726	0.2351	0.1020	0.2321	0.2035	/
TripoSR 1.4	0.3295	0.2641	0.3115	0.2709	0.3039	0.2960	21100(A100)
Hunyuan3D 1.0	0.2745	0.1725	0.2589	0.1882	0.2530	0.2294	/

1: "A small house with a gable roof built of stone and brick with square windows and doors, a traditional design."

2: "A multi-story residential building with a mix of brick and concrete facades, large glass windows, numerous balconies, and rooftop elements, creating a modern, urban appearance."

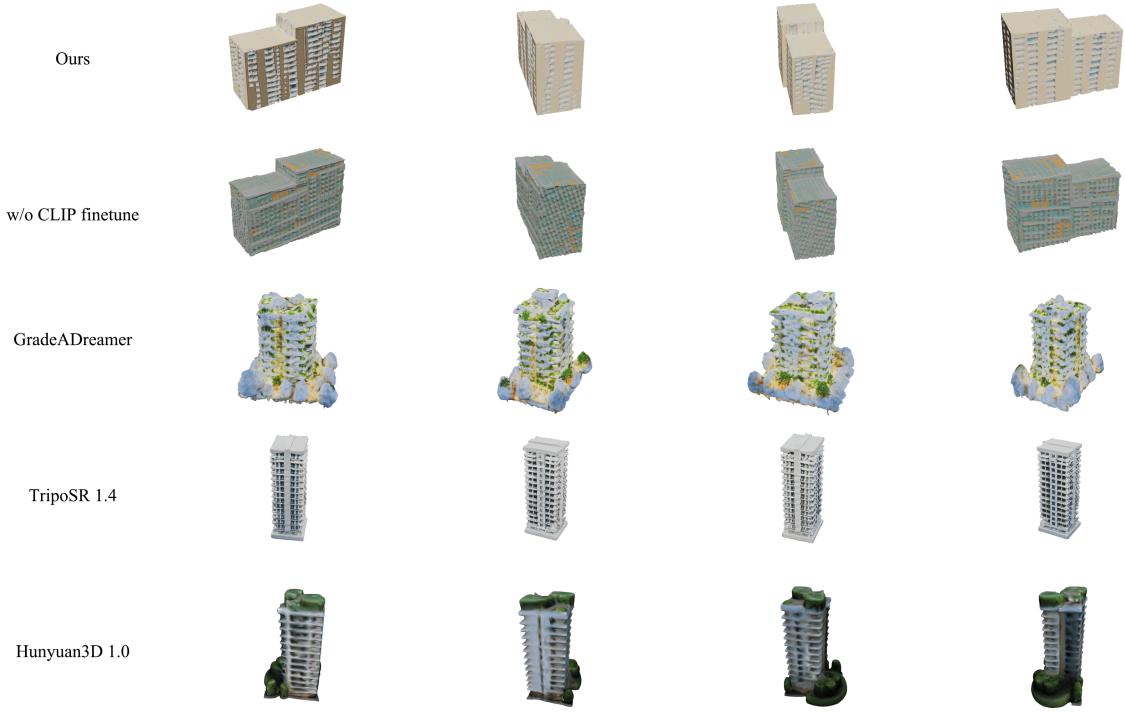
3: "A rectangular, multi-story building with a beige facade, grid-patterned walls, large blue-tinted windows, a red-tiled roof, and a single brown door, featuring classical design elements."

4: "Modern skyscraper complex with sleek, glass facade, geometric patterns, and a mix of blue and gray tones."

5: "A tall, modern high-rise building with numerous balconies, clean lines, upper with light color materials, flat roof, and textured base."

We also compare the generated mesh with a multi-view generation method based on 3D Gaussian splitting [76], a direct 3D generation method based on NeRF trained on 3D data [74], and a 3D-DiT model

Prompt: A tall, modern high-rise building with numerous balconies, clean lines, with light color materials, flat roof, and textured base.



Prompt: A multi-story residential building with a mix of brick and concrete facades, large glass windows, numerous balconies, and rooftop elements, creating a modern, urban appearance.

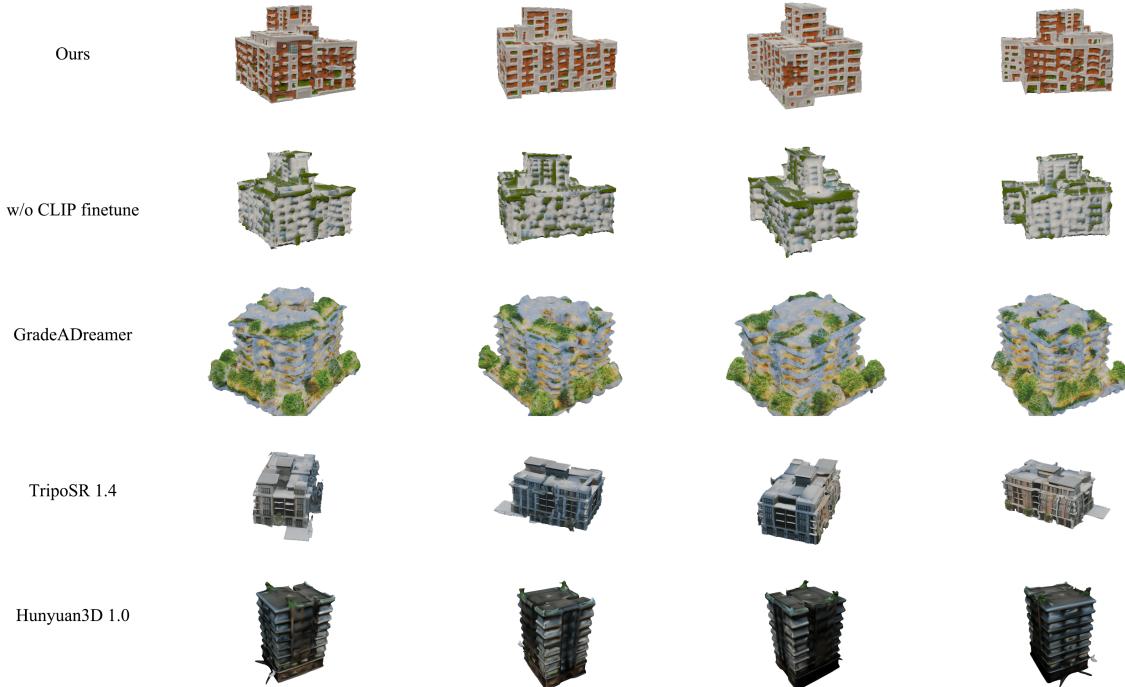


Figure 8: Comparisons among different generation methods. Notice that in other methods, there are no constraints on footprints, and the bottom shape of the generated models might vary. The comparisons of the footprints of our generated models are in Fig.10.

trained on 3D data [88]. Due to the lack of control methods on their footprint, we only demonstrate the comparison of appearance modeling from a text prompt. Besides this, the mesh extracted from 3DGs suffers from noise on the surfaces owing to the ellipsoid-like shape of 3D Gaussians. Although direct 3D generation methods appear to produce detailed geometry and textures, please note that they were trained by more expensive 3D data and up to 176 A100 GPUs for more than 21,100 GPU hours.

4.5 Limitations and Further Research

As discussed in Section 4.4, end-to-end direct 3D generative models [95, 74, 25, 82, 4, 41], trained on comprehensive 3D datasets with extensive GPU resources, may yield superior fine geometry and texture resolution. However, constraints related to data formats and computational capacity have precluded the implementation of effective geometric control in these feedforward models, as well as their integration into our pipeline. Consequently, we have adopted an alternative approach that decomposes the generation process into discrete steps and proposes a modular framework, wherein each stage generates target data under specific conditions. Investigating methods to integrate geometric control beyond renderings or text, while reducing the training costs of direct 3D generative models to facilitate domain-specific fine-tuning, represents a promising avenue for future research.

5 Conclusion

In this work, we presented GeoTexBuild, a modular framework for generating textured 3D building models from a footprint, height sketch, and style prompt. We investigated the way of controlling geometry attributes of buildings by separating the generation process and leveraging height maps. Three components achieved height map generation, geometry reconstruction, appearance stylization, and building up the whole system.

The framework accomplished fine-grain control of roof structures and detailed appearance by integrating customized ControlNet and Text2Mesh modules. The generated building meshes can be easily used in downstream tasks or further manipulation, thereby significantly reducing manual labor and offering inspiration for designers. Experiments from various perspectives substantiate the effectiveness of our framework in the generation of architectural models.

GeoTexBuild bridges creative vision and technical execution, enabling highly customized architectural designs. It also lays the groundwork for innovations like generative design integration and automated urban aesthetics analytics.

References

- [1] Al., M.B.E.: Deep learning for 3d building reconstruction: A review. In: 24th ISPRS Congress on Imaging Today, Foreseeing Tomorrow (2022)
- [2] Barron, J.T., Mildenhall, B., Verbin, D., Srinivasan, P.P., Hedman, P.: Mip-nerf 360: Unbounded anti-aliased neural radiance fields. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 5470–5479 (2022)
- [3] Bensadoun, R., Kleiman, Y., Azuri, I., Harosh, O., Vedaldi, A., Neverova, N., Gafni, O.: Meta 3d texturegen: Fast and consistent texture generation for 3d objects (2024), <https://arxiv.org/abs/2407.02430>
- [4] Bensadoun, R., Monnier, T., Kleiman, Y., Kokkinos, F., Siddiqui, Y., Kariya, M., Harosh, O., Shapovalov, R., Graham, B., Garreau, E., Karnewar, A., Cao, A., Azuri, I., Makarov, I., Le, E.T., Toisoul, A., Novotny, D., Gafni, O., Neverova, N., Vedaldi, A.: Meta 3d gen (2024), <https://arxiv.org/abs/2407.02599>
- [5] Brown, F., Cooper, G., Ford, S., Aouad, G., Brandon, P., Child, T., Kirkham, J., Oxman, R., Young, B.: An integrated approach to cad: modelling concepts in building design and construction. Design Studies **16**(3), 327–347 (1995). [https://doi.org/https://doi.org/10.1016/0142-694X\(94\)00002-U](https://doi.org/https://doi.org/10.1016/0142-694X(94)00002-U), <https://www.sciencedirect.com/science/article/pii/0142694X9400002U>
- [6] Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R.,

Prompt:

Modern multi-story building with flat roof, large windows, light and dark facade materials, small rooftop structure, possible commercial or institutional use.



Modern, multi-level house with flat roof, large wooden deck, spacious terrace, sleek lines, minimalist design, large windows, and outdoor furniture.



A modern architectural building with stacked cubic glass structures, reflective panels, sleek entrance, and geometric contemporary design.



A modern, multi-level building with a mix of white and gray materials, featuring large glass windows, vertical lines, and a unique, layered design.



Tall, modern high-rise, rectangular shape, beige and gray facade, numerous windows and balconies, clean lines, rooftop structures for ventilation. residential or commercial high-rise.



A multi-story residential building with a mix of brick and concrete facades, large glass windows, numerous balconies, and rooftop elements, creating a modern, urban appearance.



Modern skyscraper complex with sleek, glass facade, geometric patterns, and a mix of blue and gray tones.



Modern two-story house with white walls, pitched dark gray roof with red trim, multiple large windows, chimneys.



Modern single-story house with slanted roof, stone and wood elements, large windows, wooden door with glass panels, stone columns, potted plants, warm exterior lighting.



Elegant stone house with steeply pitched roof, multiple gables, arched windows, grand entrance, prominent chimney, mix of traditional and modern elements.



A charming small wooden house with a gable roof, the house features a front porch with a railing and steps leading up to the entrance.



Small orange wooden house with steeply pitched roof, white central door, two flanking windows, small side window, simple rustic design, countryside setting.



Small wooden house with steeply pitched roof, white central door, two flanking windows, small side window, simple rustic design, countryside setting.



A small house with a gable roof built of stone and brick with square windows and doors, a traditional design.



Figure 9: Qualitative results with their style prompts.

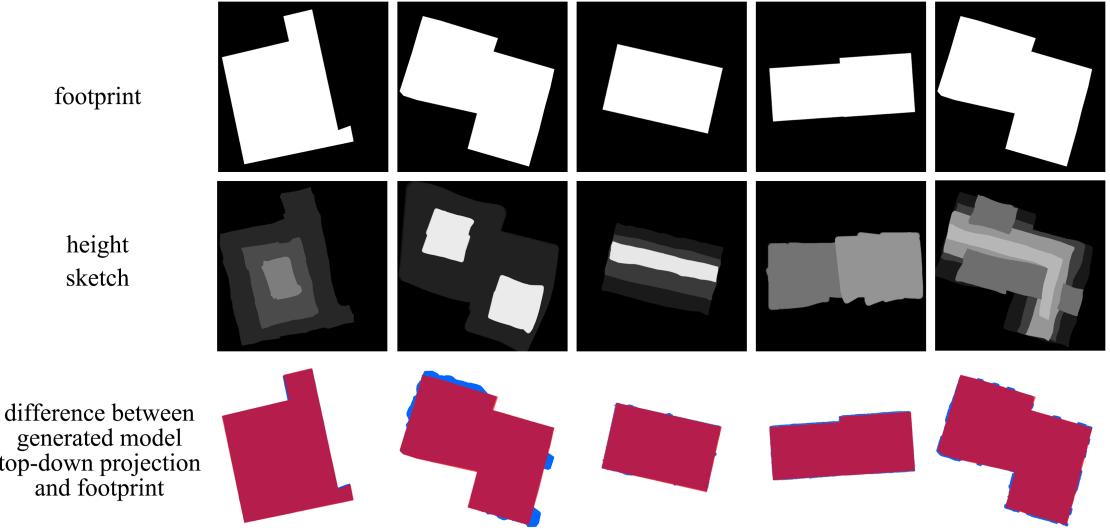


Figure 10: We demonstrate the difference between the footprints and the top-down projections of the generated models. The projection is colored blue, and the footprint is magenta. Consequently, the regions where the projection surpasses the target are represented in blue, while the areas where it falls short are indicated in red.

- Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D.: Language models are few-shot learners. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS '20, Curran Associates Inc., Red Hook, NY, USA (2020)
- [7] Chen, G., Wang, W.: A survey on 3d gaussian splatting (2025), <https://arxiv.org/abs/2401.03890>
 - [8] Chen, Q., Wang, L., Waslander, S.L., Liu, X.: An end-to-end shape modeling framework for vectorized building outline generation from aerial images. ISPRS Journal of Photogrammetry and Remote Sensing **170**, 114–126 (2020). <https://doi.org/https://doi.org/10.1016/j.isprsjprs.2020.10.008>, <https://www.sciencedirect.com/science/article/pii/S092427162030280X>
 - [9] Chen, R., Chen, Y., Jiao, N., Jia, K.: Fantasia3d: Disentangling geometry and appearance for high-quality text-to-3d content creation. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 22246–22256 (October 2023)
 - [10] Chen, Z., Ledoux, H., Khademi, S., Nan, L.: Reconstructing compact building models from point clouds using deep implicit fields. ISPRS Journal of Photogrammetry and Remote Sensing **194**, 58–73 (2022). <https://doi.org/https://doi.org/10.1016/j.isprsjprs.2022.09.017>, <https://www.sciencedirect.com/science/article/pii/S0924271622002611>
 - [11] Cherti, M., Beaumont, R., Wightman, R., Wortsman, M., Ilharco, G., Gordon, C., Schuhmann, C., Schmidt, L., Jitsev, J.: Reproducible scaling laws for contrastive language-image learning. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 2818–2829 (2022), <https://api.semanticscholar.org/CorpusID:254636568>
 - [12] Cheskidova, E., Arganaidi, A., Rancea, D.I., Haag, O.: Geometry aware texturing. In: SIGGRAPH Asia 2023 Posters. SA '23, Association for Computing Machinery, New York, NY, USA (2023). <https://doi.org/10.1145/3610542.3626152>
 - [13] Dawen, Y., Wei, S., Liu, J., Ji, S.: Advanced approach for automatic reconstruction of 3d buildings from aerial images. ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences **XLIII-B2-2020**, 541–546 (08 2020). <https://doi.org/10.5194/isprs-archives-XLIII-B2-2020-541-2020>

- [14] Debevec, P.E., Taylor, C.J., Malik, J.: Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques. p. 11–20. SIGGRAPH '96, Association for Computing Machinery, New York, NY, USA (1996). <https://doi.org/10.1145/237170.237191>, <https://doi.org/10.1145/237170.237191>
- [15] Dick, A., Torr, P., Cipolla, R.: Automatic 3d modelling of architecture. (01 2000)
- [16] Dick, A., Torr, P., Cipolla, R.: Modelling and interpretation of architecture from several images. International Journal of Computer Vision **60** (11 2004). <https://doi.org/10.1023/B:VISI.0000029665.07652.61>
- [17] Elaksher, A., Bethel, J.: Reconstructing 3d buildings from lidar data. International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences **34** (05 2002)
- [18] Fei, B., Xu, J., Zhang, R., Zhou, Q., Yang, W., He, Y.: 3d gaussian splatting as new era: A survey. IEEE transactions on visualization and computer graphics **PP** (2024), <https://api.semanticscholar.org/CorpusID:267626952>
- [19] Gao, K., Gao, Y., He, H., Lu, D., Xu, L., Li, J.: Nerf: Neural radiance field in 3d vision, a comprehensive review (2023), <https://arxiv.org/abs/2210.00379>
- [20] Guédon, A., Lepetit, V.: Sugar: Surface-aligned gaussian splatting for efficient 3d mesh reconstruction and high-quality mesh rendering. CVPR (2024)
- [21] Haque, A., Tancik, M., Efros, A.A., Holynski, A., Kanazawa, A.: Instruct-nerf2nerf: Editing 3d scenes with instructions. 2023 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 19683–19693 (2023), <https://api.semanticscholar.org/CorpusID:257663414>
- [22] Hart, J.C.: The object instancing paradigm for linear fractal modeling. In: Proceedings of the Conference on Graphics Interface '92. p. 224–231. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1992)
- [23] Henricsson, O., Gruen, A.: Automated 3-d reconstruction of buildings and visualization of city models (10 1996)
- [24] Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. In: Proceedings of the 34th International Conference on Neural Information Processing Systems. NIPS '20, Curran Associates Inc., Red Hook, NY, USA (2020)
- [25] Hong, Y., Zhang, K., Gu, J., Bi, S., Zhou, Y., Liu, D., Liu, F., Sunkavalli, K., Bui, T., Tan, H.: LRM: Large reconstruction model for single image to 3d. In: The Twelfth International Conference on Learning Representations (2024), <https://openreview.net/forum?id=s11U8vvsFF>
- [26] Huang, B., Yu, Z., Chen, A., Geiger, A., Gao, S.: 2d gaussian splatting for geometrically accurate radiance fields. In: SIGGRAPH 2024 Conference Papers. Association for Computing Machinery (2024). <https://doi.org/10.1145/3641519.3657428>
- [27] Huang, J., Stoter, J., Peters, R., Nan, L.: City3d: Large-scale building reconstruction from airborne lidar point clouds. Remote Sensing **14**(9) (2022). <https://doi.org/10.3390/rs14092254>, <https://www.mdpi.com/2072-4292/14/9/2254>
- [28] Hwang, I., Kim, H., Kim, Y.M.: Text2scene: Text-driven indoor scene stylization with part-aware details. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 1890–1899 (June 2023)
- [29] Jun, H., Nichol, A.: Shap-e: Generating conditional 3d implicit functions (2023), <https://arxiv.org/abs/2305.02463>
- [30] Kada, M., McKinley, L.: 3d building reconstruction from lidar based on a cell decomposition approach. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences **38** (09 2009)

- [31] Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: Proceedings of the Fourth Eurographics Symposium on Geometry Processing. p. 61–70. SGP ’06, Eurographics Association, Goslar, DEU (2006)
- [32] Kerbl, B., Kopanas, G., Leimkühler, T., Drettakis, G.: 3d gaussian splatting for real-time radiance field rendering. ACM Transactions on Graphics **42**(4) (July 2023), <https://repo-sam.inria.fr/fungraph/3d-gaussian-splatting/>
- [33] Lafarge, F., Descombes, X., Zerubia, J., Deseilligny, M.: An automatic building reconstruction method: A structural approach using high resolution satellite images. In: International Conference on Image Processing. pp. 1205 – 1208 (11 2006). <https://doi.org/10.1109/ICIP.2006.312541>
- [34] Lee, H.H., Savva, M., Chang, A.X.: Text-to-3d shape generation (2024)
- [35] Lee, J., Lee, S., Jo, C., Im, W., Seon, J., Yoon, S.E.: SemCity: Semantic Scene Generation with Triplane Diffusion . In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 28337–28347. IEEE Computer Society, Los Alamitos, CA, USA (Jun 2024). <https://doi.org/10.1109/CVPR52733.2024.02677>, <https://doi.ieee.org/10.1109/CVPR52733.2024.02677>
- [36] Li, H., Shi, H., Zhang, W., Wu, W., Liao, Y., Wang, L., Lee, L.H., Zhou, P.Y.: Dreamscape: 3d gaussian-based text-to-3d scene generation via formation pattern sampling. In: Leonardis, A., Ricci, E., Roth, S., Russakovsky, O., Sattler, T., Varol, G. (eds.) Computer Vision – ECCV 2024. pp. 214–230. Springer Nature Switzerland, Cham (2025)
- [37] Li, Q., Mou, L., Sun, Y., Hua, Y., Shi, Y., Zhu, X.X.: A review of building extraction from remote sensing imagery: Geometrical structures and semantic attributes. IEEE Transactions on Geoscience and Remote Sensing **62**, 1–15 (2024). <https://doi.org/10.1109/TGRS.2024.3369723>
- [38] Li, X., Zhang, Q., Kang, D., Cheng, W., Gao, Y., Zhang, J., Liang, Z., Liao, J., Cao, Y.P., Shan, Y.: Advances in 3d generation: A survey (2024), <https://arxiv.org/abs/2401.17807>
- [39] Li, Z., Müller, T., Evans, A., Taylor, R.H., Unberath, M., Liu, M.Y., Lin, C.H.: Neuralangelo: High-fidelity neural surface reconstruction. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2023)
- [40] Lin, J., Li, Z., Tang, X., Liu, J., Liu, S., Liu, J., Lu, Y., Wu, X., Xu, S., Yan, Y., Yang, W.: Vastgaussian: Vast 3d gaussians for large scene reconstruction. In: CVPR (2024)
- [41] Long, X., Guo, Y., Lin, C., Liu, Y., Dou, Z., Liu, L., Ma, Y., Zhang, S.H., Habermann, M., Theobalt, C., Wang, W.: Wonder3d: Single image to 3d using cross-domain diffusion. 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 9970–9980 (2023), <https://api.semanticscholar.org/CorpusID:264436465>
- [42] Luo, H., Zhang, J., Liu, X., Zhang, L., Liu, J.: Large-scale 3d reconstruction from multi-view imagery: A comprehensive review. Remote Sensing **16**(5) (2024). <https://doi.org/10.3390/rs16050773>, <https://www.mdpi.com/2072-4292/16/5/773>
- [43] Lynch, K., Hack, G.: Site planning. MIT press (1984)
- [44] Merritt, F.S.: Building design and construction handbook. McGraw-Hill (2001)
- [45] Metzer, G., Richardson, E., Patashnik, O., Giryes, R., Cohen-Or, D.: Latent-nerf for shape-guided generation of 3d shapes and textures. 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) pp. 12663–12673 (2022), <https://api.semanticscholar.org/CorpusID:253510536>
- [46] Michel, O., Bar-On, R., Liu, R., Benaim, S., Hanocka, R.: Text2mesh: Text-driven neural stylization for meshes. In: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 13482–13492. IEEE Computer Society, Los Alamitos, CA, USA (Jun 2022). <https://doi.org/10.1109/CVPR52688.2022.01313>, <https://doi.ieee.org/10.1109/CVPR52688.2022.01313>

- [47] Milde, J., Zhang, Y., Brenner, C., Plümer, L., Sester, M.: Building reconstruction using a structural description based on a formal grammar. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences **37** (01 2008)
- [48] Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R.: Nerf: Representing scenes as neural radiance fields for view synthesis. In: ECCV (2020)
- [49] Moore, M.: Basics of game design. CRC Press (2016)
- [50] Müller, T., Evans, A., Schied, C., Keller, A.: Instant neural graphics primitives with a multiresolution hash encoding. ACM Trans. Graph. **41**(4), 102:1–102:15 (Jul 2022). <https://doi.org/10.1145/3528223.3530127>, <https://doi.org/10.1145/3528223.3530127>
- [51] Nan, L.: Easy3d: a lightweight, easy-to-use, and efficient c++ library for processing and rendering 3d data. Journal of Open Source Software **6**(64), 3255 (2021). <https://doi.org/10.21105/joss.03255>, <https://doi.org/10.21105/joss.03255>
- [52] Nan, L., Wonka, P.: Polyfit: Polygonal surface reconstruction from point clouds. In: 2017 IEEE International Conference on Computer Vision (ICCV). pp. 2372–2380 (2017). <https://doi.org/10.1109/ICCV.2017.258>
- [53] Nichol, A., Jun, H., Dhariwal, P., Mishkin, P., Chen, M.: Point-e: A system for generating 3d point clouds from complex prompts (2022), <https://arxiv.org/abs/2212.08751>
- [54] Nishida, G., Bousseau, A., Aliaga, D.G.: Procedural modeling of a building from a single image. Computer Graphics Forum **37**(2), 415–429 (2018). <https://doi.org/https://doi.org/10.1111/cgf.13372>
- [55] Ouyang, H., Heal, K., Lombardi, S., Sun, T.: Text2immersion: Generative immersive scene with 3d gaussians (2023), <https://arxiv.org/abs/2312.09242>
- [56] Padén, I., Peters, R., García-Sánchez, C., Ledoux, H.: Automatic high-detailed building reconstruction workflow for urban microscale simulations. Building and Environment **265**, 111978 (2024). <https://doi.org/https://doi.org/10.1016/j.buildenv.2024.111978>, <https://www.sciencedirect.com/science/article/pii/S0360132324008205>
- [57] Pang, H.E., Biljecki, F.: 3d building reconstruction from single street view images using deep learning. International Journal of Applied Earth Observation and Geoinformation **112**, 102859 (2022). <https://doi.org/https://doi.org/10.1016/j.jag.2022.102859>, <https://www.sciencedirect.com/science/article/pii/S1569843222000619>
- [58] Parish, Y.I.H., Müller, P.: Procedural modeling of cities. In: Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques. p. 301–308. SIGGRAPH '01, Association for Computing Machinery, New York, NY, USA (2001). <https://doi.org/10.1145/383259.383292>, <https://doi.org/10.1145/383259.383292>
- [59] Partovi, T., Fraundorfer, F., Bahmanyar, R., Huang, H., Reinartz, P.: Automatic 3-d building model reconstruction from very high resolution stereo satellite imagery. Remote Sensing **11**(14) (2019). <https://doi.org/10.3390/rs11141660>, <https://www.mdpi.com/2072-4292/11/14/1660>
- [60] Poole, B., Jain, A., Barron, J.T., Mildenhall, B.: Dreamfusion: Text-to-3d using 2d diffusion. arXiv (2022)
- [61] Presa Reyes, M.E., Chen, S.C.: A 3d virtual environment for storm surge flooding animation. In: 2017 IEEE Third International Conference on Multimedia Big Data (BigMM). pp. 244–245 (2017). <https://doi.org/10.1109/BigMM.2017.754>
- [62] Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., Sutskever, I.: Learning transferable visual models from natural language supervision. In: International Conference on Machine Learning (2021), <https://api.semanticscholar.org/CorpusID:231591445>
- [63] Raj, A., Kaza, S., Poole, B., Niemeyer, M., Mildenhall, B., Ruiz, N., Zada, S., Aberman, K., Rubenstein, M., Barron, J., Li, Y., Jampani, V.: Dreambooth3d: Subject-driven text-to-3d generation. ICCV (2023)

- [64] Ren, X., Huang, J., Zeng, X., Museth, K., Fidler, S., Williams, F.: Xcube: Large-scale 3d generative modeling using sparse voxel hierarchies. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 4209–4219 (June 2024)
- [65] Schönberger, J.L., Frahm, J.M.: Structure-from-motion revisited. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 4104–4113 (2016), <https://api.semanticscholar.org/CorpusID:1728538>
- [66] Shang, Y., Lin, Y., Zheng, Y., Fan, H., Ding, J., Feng, J., Chen, J., Tian, L., Li, Y.: Urbanworld: An urban world model for 3d city generation (2024), <https://arxiv.org/abs/2407.11965>
- [67] Shi, Y., Wang, P., Ye, J., Mai, L., Li, K., Yang, X.: Mvdream: Multi-view diffusion for 3d generation. arXiv:2308.16512 (2023)
- [68] Shiode, N.: 3d urban models: Recent developments in the digital modelling of urban environments in three-dimensions. GeoJournal **52**, 263–269 (04 2012). <https://doi.org/10.1023/A:1014276309416>
- [69] Siddiqui, Y., Monnier, T., Kokkinos, F., Kariya, M., Kleiman, Y., Garreau, E., Gafni, O., Neverova, N., Vedaldi, A., Shapovalov, R., Novotny, D.: Meta 3d assetgen: Text-to-mesh generation with high-quality geometry, texture, and pbr materials (2024), <https://arxiv.org/abs/2407.02445>
- [70] Storcz, T., Ercsey, Z., Horváth, K.R., Kovács, Z., Dávid, B., Kistelekdi, I.: Energy design synthesis: Algorithmic generation of building shape configurations. Energies **16**(5) (2023). <https://doi.org/10.3390/en16052254>, <https://www.mdpi.com/1996-1073/16/5/2254>
- [71] Tancik, M., Casser, V., Yan, X., Pradhan, S., Mildenhall, B., Srinivasan, P.P., Barron, J.T., Kretzschmar, H.: Block-nerf: Scalable large scene neural view synthesis. CVPR (2022)
- [72] Tang, J., Ren, J., Zhou, H., Liu, Z., Zeng, G.: Dreamgaussian: Generative gaussian splatting for efficient 3d content creation. ICLR (2024)
- [73] Tao, Y., Ángel Muñoz-Bañón, M., Zhang, L., Wang, J., Fu, L.F.T., Fallon, M.: The oxford spires dataset: Benchmarking large-scale lidar-visual localisation, reconstruction and radiance field methods (2024), <https://arxiv.org/abs/2411.10546>
- [74] Tochilkin, D., Pankratz, D., Liu, Z., Huang, Z., Letts, A., Li, Y., Liang, D., Laforte, C., Jampani, V., Cao, Y.P.: Tripocr: Fast 3d object reconstruction from a single image (2024), <https://arxiv.org/abs/2403.02151>
- [75] Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: Proceedings of the Sixth International Conference on Computer Vision. p. 839. ICCV '98, IEEE Computer Society, USA (1998)
- [76] Ukarapol, T., Pruvost, K.: Gradeadreamer: Enhanced text-to-3d generation using gaussian splatting and multi-view diffusion (2024)
- [77] Wang, J., Fang, J., Zhang, X., Xie, L., Tian, Q.: GaussianEditor: Editing 3D Gaussians Delicately with Text Instructions . In: 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). IEEE Computer Society, Los Alamitos, CA, USA (Jun 2024). <https://doi.org/10.1109/CVPR52733.2024.01975>, <https://doi.ieee.org/10.1109/CVPR52733.2024.01975>
- [78] Wang, R., Huang, S., Yang, H.: Building3d: An urban-scale dataset and benchmarks for learning roof structures from point clouds. 2023 IEEE/CVF International Conference on Computer Vision (ICCV) pp. 20019–20029 (2023), <https://api.semanticscholar.org/CorpusID:260125967>
- [79] Wang, S., Mao, Z., Zeng, C., Gong, H., Li, S., Chen, B.: A new method of virtual reality based on unity3d. In: 2010 18th International Conference on Geoinformatics. pp. 1–5 (2010). <https://doi.org/10.1109/GEOINFORMATICS.2010.5567608>
- [80] Wang, X., Zhu, J., Ye, Q., Huo, Y., Ran, Y., Zhong, Z., Chen, J.: Seal-3D: Interactive Pixel-Level Editing for Neural Radiance Fields . In: 2023 IEEE/CVF International Conference on Computer Vision (ICCV). pp. 17637–17647. IEEE Computer Society, Los Alamitos, CA, USA (Oct 2023). <https://doi.org/10.1109/ICCV51070.2023.01621>, <https://doi.ieee.org/10.1109/ICCV51070.2023.01621>

- [81] Wei, Y., Vosselman, G., Yang, M.Y.: Buildiff: 3d building shape generation using single-image conditional point cloud diffusion models. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops. pp. 2910–2919 (October 2023)
- [82] Wu, S., Lin, Y., Zhang, F., Zeng, Y., Xu, J., Torr, P., Cao, X., Yao, Y.: Direct3d: Scalable image-to-3d generation via 3d latent diffusion transformer. Advances in Neural Information Processing Systems **37**, 121859–121881 (2024)
- [83] Wu, Y., Xue, F., Li, M., Chen, S.H.: A novel building section skeleton for compact 3d reconstruction from point clouds: A study of high-density urban scenes. ISPRS Journal of Photogrammetry and Remote Sensing **209**, 85–100 (2024). <https://doi.org/https://doi.org/10.1016/j.isprsjprs.2024.01.020>, <https://www.sciencedirect.com/science/article/pii/S0924271624000297>
- [84] Wysocki, O., Hoegner, L., Stilla, U.: Mls2lod3: Refining low lod3 building models with mls point clouds to reconstruct semantic lod3 building models. In: Recent Advances in 3D Geoinformation Science. pp. 367–380. Springer Nature Switzerland, Cham (2024)
- [85] Xiong, B., Zheng, N., Liu, J., Li, Z.: Gauu-scene v2: Assessing the reliability of image-based metrics with expansive lidar image dataset using 3dgs and nerf. ArXiv **abs/2404.04880** (2024), <https://api.semanticscholar.org/CorpusID:269005139>
- [86] Xu, Q.C., Mu, T.J., Yang, Y.: A survey of deep learning-based 3d shape generation. Computational Visual Media **9**(3), 407–442 (Sep 2023). <https://doi.org/10.1007/s41095-022-0321-5>
- [87] Xu, Y., Ng, Y., Wang, Y., Sa, I., Duan, Y., Li, Y., Ji, P., Li, H.: Sketch2scene: Automatic generation of interactive 3d game scenes from user’s casual sketches. ArXiv **abs/2408.04567** (2024), <https://api.semanticscholar.org/CorpusID:271769138>
- [88] Yang, X., Shi, H., Zhang, B., Yang, F., Wang, J., Zhao, H., Liu, X., Wang, X., Lin, Q., Yu, J., Wang, L., Xu, J., He, Z., Chen, Z., Liu, S., Wu, J., Lian, Y., Yang, S., Liu, Y., Yang, Y., Wang, D., Jiang, J., Guo, C.: Hunyuan3d 1.0: A unified framework for text-to-3d and image-to-3d generation (2025), <https://arxiv.org/abs/2411.02293>
- [89] Yi, T., Fang, J., Wang, J., Wu, G., Xie, L., Zhang, X., Liu, W., Tian, Q., Wang, X.: Gaussian-dreamer: Fast generation from text to 3d gaussians by bridging 2d and 3d diffusion models. In: CVPR (2024)
- [90] Yu, H.X., Duan, H., Herrmann, C., Freeman, W.T., Wu, J.: Wonderworld: Interactive 3d scene generation from a single image. CoRR **abs/2406.09394** (2024), <https://doi.org/10.48550/arXiv.2406.09394>
- [91] Yu, X.Y., Yu, J.X., Zhou, L.B., Wei, Y., Ou, L.L.: Instantstylegaussian: Efficient art style transfer with 3d gaussian splatting (2024), <https://arxiv.org/abs/2408.04249>
- [92] Yu, Y., Ha, D., Lee, K., Choi, J., Koo, B.: Archshapesnet: a novel dataset for benchmarking architectural building information modeling element classification algorithms. Journal of Computational Design and Engineering **9**(4), 1449–1466 (07 2022). <https://doi.org/10.1093/jcde/qwac064>, <https://doi.org/10.1093/jcde/qwac064>
- [93] Zhang, J., Li, X., Wan, Z., Wang, C., Liao, J.: Text2NeRF: Text-Driven 3D Scene Generation With Neural Radiance Fields . IEEE Transactions on Visualization & Computer Graphics **30**(12), 7749–7762 (Dec 2024). <https://doi.org/10.1109/TVCG.2024.3361502>, <https://doi.ieee.org/10.1109/TVCG.2024.3361502>
- [94] Zhang, L., Rao, A., Agrawala, M.: Adding conditional control to text-to-image diffusion models. In: IEEE International Conference on Computer Vision (ICCV) (2023)
- [95] Zhao, Z., Lai, Z., Lin, Q., Zhao, Y., Liu, H., Yang, S., Feng, Y., Yang, M., Zhang, S., Yang, X., Shi, H., Liu, S., Wu, J., Lian, Y., Yang, F., Tang, R., He, Z., Wang, X., Liu, J., Zuo, X., Chen, Z., Lei, B., Weng, H., Xu, J., Zhu, Y., Liu, X., Xu, L., Hu, C., Yang, S., Zhang, S., Liu, Y., Huang, T., Wang, L., Zhang, J., Chen, M., Dong, L., Jia, Y., Cai, Y., Yu, J., Tang, Y., Zhang, H., Ye, Z., He, P., Wu, R., Zhang, C., Tan, Y., Xiao, J., Tao, Y., Zhu, J., Xue, J., Liu, K., Zhao, C., Wu, X., Hu, Z., Qin, L., Peng, J., Li, Z., Chen, M., Zhang, X., Niu, L., Wang, P., Wang, Y., Kuang, H., Fan, Z.,

- Zheng, X., Zhuang, W., He, Y., Liu, T., Yang, Y., Wang, D., Liu, Y., Jiang, J., Huang, J., Guo, C.: Hunyuan3d 2.0: Scaling diffusion models for high resolution textured 3d assets generation (2025), <https://arxiv.org/abs/2501.12202>
- [96] Zhao, Z., Liu, W., Chen, X., Zeng, X., Wang, R., Cheng, P., FU, B., Chen, T., YU, G., Gao, S.: Michelangelo: Conditional 3d shape generation based on shape-image-text aligned latent representation. In: Thirty-seventh Conference on Neural Information Processing Systems (2023), <https://openreview.net/forum?id=xmxgMij3LY>
- [97] Zhou, L., Du, Y., Wu, J.: 3d shape generation and completion through point-voxel diffusion. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 5826–5835 (October 2021)
- [98] Zhuang, J., Li, G., Xu, H., Xu, J., Tian, R.: Text-to-city: Controllable 3d urban block generation with latent diffusion model. In: CAADRIA 2024 (04 2024). <https://doi.org/10.52842/conf.caadria.2024.2.169>