ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences, Volume X-4/W6-2025
20th 3D GeoInfo Conference 2025, 2–5 September 2025, Kashiwa, Japan

Virtual 3D City Model Generation in CityGML

Chenbo Zhao*, Yoshiki Ogawa, Lingfeng Liao, Yoshihide Sekimoto

Center for Spatial Information Science (CSIS), The University of Tokyo, Japan — (zhaocb, ogawa, lfliao, sekimoto)@csis.u-tokyo.ac.jp

Keywords: Virtual City, CityGML, Digital Cousin, 3D Model Generation, Level of detail (LOD).

Abstract

As the urban digital transformation continues to advance, virtual 3D city models have become essential tools for urban planning, traffic management, environmental assessment, and virtual reality applications. Current research largely focuses on constructing[1] high-fidelity city models based on the CityGML standard; however, challenges remain regarding data acquisition costs, complexity of generation processes, and customization capabilities. To address these issues, this study proposes an automated virtual city model generation method that integrates open data (such as OSM, DEM, and open-source LOD2 models) with the concept of digital cousin. This method efficiently generates 3D city models with varying levels of detail, from LOD 0 to LOD 2, by integrating and parameterizing multisource data, including relief, roads, city furniture, vegetation, and buildings. Moreover, it supports flexible user adjustments of key parameters, such as vegetation density, road width, traffic light intervals, building heights, and roof types. Compared with traditional methods that rely on expensive surveying data and labor-intensive manual operations, the proposed approach offers a low-cost, highly flexible, and scalable solution, thereby providing robust support for a wide range of urban simulation and decision-making applications. The code used in this study is as follows:
https://github.com/CBZhao2021/gen3D_virtualCity.git

Figure 1. Example of all features (relief, road, city furniture, vegetation, and building) visualization for virtual 3D city model generation.

1. Introduction

As global urbanization accelerates and digital transformation deepens, there is an increasing demand for high-quality real-time digital city models for urban planning, traffic management, environmental assessment, and virtual reality applications (Biljecki et al., 2015; Yao et al., 2018). Virtual 3D city models serve as vital bridges between real cities and the digital realm, providing an intuitive data foundation for decision support and simulation analyses (Singh et al., 2013). However, traditional high-fidelity city models typically rely on expensive survey data and complex manual processing workflows (Früh & Zakhor,

* Corresponding author

https://doi.org/10.5194/isprs-annals-X-4-W6-2025-257-2025

2004; Wang et al., 2014), which not only increase data acquisition costs but also limit the automation and customization of model generation.

With advancements in computational power and deep learning algorithms, numerous automated approaches for generating high-fidelity 3D models have emerged, such as those based on neural radiance fields (NeRF) (Mildenhall et al., 2021), 3D Gaussian splatting (Kerbl et al., 2023), and 3D city generation methods, such as CityDreamer (Xie et al., 2024a) and GaussianCity (Xie et al., 2024b). However, these methods produced outputs in the form of implicit NeRF representations, point clouds with Gaussian ellipsoidal distributions, or voxel-based models. Although their renderings may closely approximate real-world scenes, such representations are not well-suited for further exploration in urban planning and Geographic Information System (GIS) analyses, such as solar, wind, and transportation simulations.

By contrast, recent years have witnessed growing attention toward constructing city models based on the CityGML standard (Arroyo Ohori et al., 2018; Tan et al., 2023) by the Open Geospatial Consortium (OGC). CityGML not only describes the geometric form, topological relationships, and semantic information of urban structures but also supports multiple levels of detail (LOD), thereby providing flexible data representation for a wide range of applications. Supported by an open-source ecosystem and characterized by high-density semantic information, CityGML has attracted significant research interest and has been employed to build 3D city models that underpin further analyses and simulations such as interactive operations (Gröger & Plümer, 2012; Kolbe et al., 2005), energy simulation modelling (Malhotra et al., 2022), and smart city deployments (Prandi et al., 2013).

Although CityGML offers a powerful framework, constructing such detailed models at the city scale remains challenging, especially in terms of data acquisition and processing costs. Recently, the increasing availability of open data resources has mitigated this challenge.

The widespread availability of open data resources, including OpenStreetMap (OSM), digital elevation models (DEM), and open-source LOD2 models, makes it feasible to generate CityGML-formatted virtual city models using low-cost, high-coverage data. Inspired by the concept of digital cousins, which leverage existing database resources to achieve high-fidelity 3D city models at lower cost and with greater flexibility, this study proposes an automated method for generating virtual city models. Specifically, our approach integrates multisource open data and adopts the digital cousin philosophy (Dai et al., 2024) to efficiently construct city models with varying levels of detail from 0 to 2. Compared with traditional methods (Biljecki et al., 2016; Goetz, 2013; Isikdag & Zlatanova, 2009; Over et al., 2010), the proposed scheme offers significant advantages in terms of data cost, automation, and parameter customization, thereby meeting diverse requirements for model accuracy and detail across different scenarios.
The main contributions of this study are as follows:

1. We propose an automated framework for generating virtual city models based on open data and the concept of digital cousin concept.
2. Implement a multilevel model generation approach covering LOD 0 to LOD 2 for roads, city furniture, vegetation, and buildings, with support for the flexible adjustment of key parameters.
3. Validation of effectiveness of proposed method for rapidly generating virtual city models with different random seeds.

An example of the generation results is shown in Figure 1.

2. Related works

2.1 Virtual 3D City Model in CityGML

Previous research on CityGML primarily focused on the application of this data standard in areas such as illumination and energy simulation, the development and management of CityGML databases, and traffic modeling (Malhotra et al., 2022; Prandi et al., 2013; Singh et al., 2013; Yao et al., 2018). By contrast, relatively little work has been dedicated to generating CityGML models.

Among the few studies that have addressed the generation of CityGML data, Random3Dcity (Biljecki et al., 2016) can produce models ranging from LOD1 to LOD3. However, the generated building configurations do not correspond to real urban environments, because they are arranged in perfect square grids with a fixed number of structures per row and column, and the building templates themselves are entirely fictional. Similarly, although Goetz et al. (Goetz, 2013) proposed a method for generating CityGML LOD4 models from OSM data, their approach was limited to a very small number of buildings. Murtiyoso et al. (Murtiyoso et al., 2020) employed point-cloud reconstruction techniques to generate data conforming to the CityGML LOD2 standard; however, this method involves a lengthy process with complex operations.

More recently, Peters et al. (Peters et al., 2022) presented a fully automated, nationwide workflow that reconstructed LoD 1.2/1.3 and LoD 2.2 building models for all ~10 million buildings in the Netherlands by fusing country-wide LiDAR with cadastral footprints – offering an impressive demonstration of large-scale, open 3D data delivery. However, the pipeline still stops short of LoD 3-4 detail, lacks façade semantics and interiors, and its geometric accuracy remains constrained by input-data resolution and temporal inconsistencies. Complementing reconstruction-oriented studies, Shang et al. (Shang et al., 2024) introduced UrbanWorld, a generative "urban world model" that combines an urban-specific multimodal LLM with progressive 3D diffusion to create controllable, interactive city scenes from OSM layouts, height maps or text/image prompts. While UrbanWorld achieves state-of-the-art visual realism and supports embodied agent navigation, it does not yet export CityGML-compliant geometry, provides no guarantees of geospatial fidelity or topological correctness, and its evaluation is confined to perceptual metrics, limiting its applicability for analytical GIS workflows.

Considering these limitations, this study presents a high-fidelity and user-friendly virtual 3D City Model Generation method in CityGML that encompasses five key features: relief, roads, urban city furniture, vegetation, and buildings. The proposed approach leverages multisource open data to efficiently generate detailed and realistic urban models, thereby addressing the need for an automated generation process that is cost-effective and easily deployable for urban planning, simulations, and further analytical applications.

2.2 Virtual 3D City Model in Rendering

Recently, NeRF (Mildenhall et al., 2021) and 3D Gaussian Splatting (Kerbl et al., 2023) have emerged as prominent neural rendering methods. NeRF employs a neural network to model a continuous volume, map 3D spatial coordinates, and view directions for volume density and color values. The output is an implicit representation, which typically requires extensive training time and significant computational resources for per-pixel ray sampling and integration. By contrast, 3D Gaussian Splatting adopts a point cloud representation based on 3D Gaussian distributions, discretizing the scene into a set of Gaussian kernels, each encapsulating position, color, and scale information. Although these methods can theoretically be used to generate urban 3D models, some studies have demonstrated the rendering of large-scale urban 3D models (Xie et al., 2024b; Xu et al., 2023), the results produced are primarily suited for visual presentation and immersive experiences, and are less readily applicable to urban planning, GIS analysis, or subsequent physical simulations.

Moreover, diffusion model-based (Ho et al., 2020) approaches, such as CityCraft (Deng et al., 2024), Infinicity (Lin et al., 2023), CityDreamer (Xie et al., 2024a), and CityGen (Deng et al., 2023), first generate a 2D layout and then derive the visual appearance of urban 3D models using asset placement or neural rendering techniques. To apply these methods to urban model generation, additional steps are required to extract, segment, or reconstruct explicit geometric and semantic information, thereby satisfying the demands of digital twin applications, interactive operations, and detailed analyses. Furthermore, these approaches typically require high-performance graphics processing units (GPUs) for training and inference, which poses challenges to their practical deployment.

3. Datasets

This study utilized three types of data: LOD2 3D building data, DEM data, and OSM data. As all datasets are open source, they offer a reliable basis for the reproducibility of the research. These datasets are curated and provided in a downloadable format on GitHub.

3.1 3D Building Data in LOD2

The LOD2 3D building data utilized in this study were obtained from the PLATEAU project, an open-source 3D urban modeling initiative led by Japan's Ministry of Land, Infrastructure, Transport, and Tourism (MLIT). This project aims to promote the standardization and open accessibility of 3D urban model data across the country. PLATEAU provides 3D building datasets in the CityGML2.0 format[2]. Examples and application scenarios ranging from LOD0 to LOD3 are shown in Figure 2.

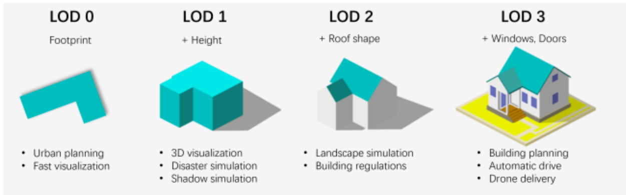

Figure 2. Examples and possible applications of LOD 0 to 3.

To meet the requirements for generating a virtual 3D city for this study, we first collected CityGML building data from various regions. The CityGML data were then converted into 69,551 individual buildings and saved in OBJ format. We convert CityGML data into .obj format to enable efficient geometric processing and visualization. Compared to the structurally complex and semantically rich but computationally heavy CityGML format, OBJ offers a lightweight and widely compatible representation of geometry, making it more suitable for modeling, reconstruction, and graphical analysis workflows. Because generating a virtual 3D city requires control over the roof types of the generated buildings, we manually labeled each building with one of the six roof categories. Table 1 presents the category names and their respective distributions.

Roof Type	Number
Flat	17,009
Stepped	11,903
Composite	21,703
Hip	2,979
Gable	15,920
Unconventional	37
Total	69,551

Table 1. Roof type and corresponding data number.

Examples of these roof types are shown in Figure 3.

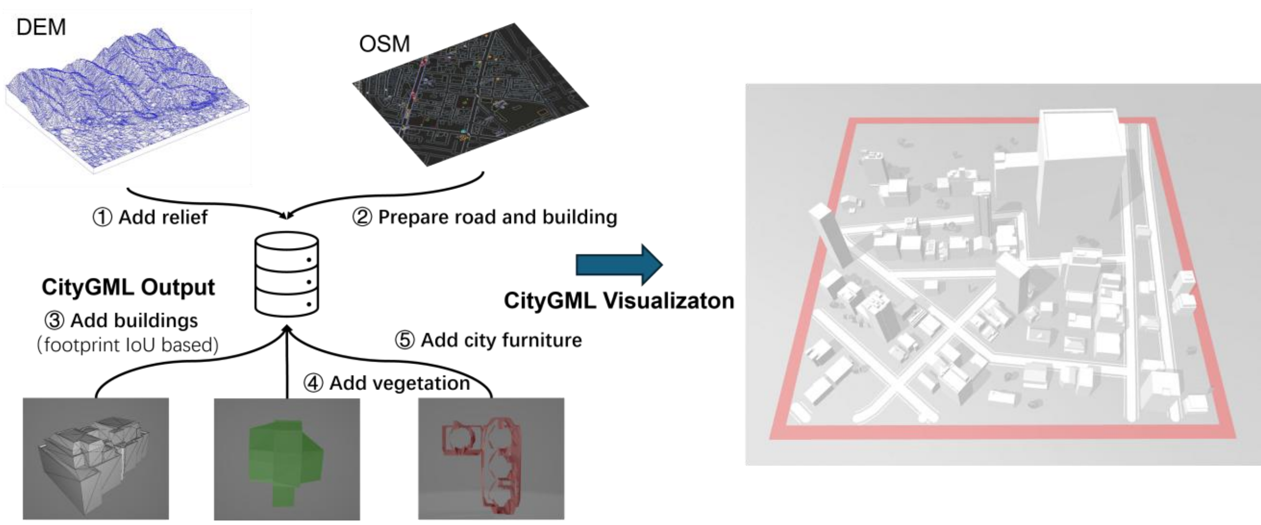

Figure 4. Workflow of virtual 3D city model generation in CityGML.

[1] https://www.geospatial.jp/ckan/dataset/plateau

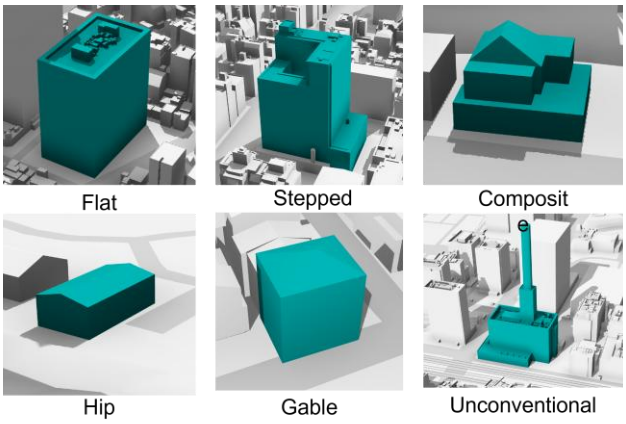

Figure 3. Examples of the six different roof types.

3.2 3D Vegetation and City Furniture Data in LOD2

Category	Number
Plant cover	192
Solitary vegetation	279
Electricity pole	10
Streetlight	8
Traffic light	9

Table 2. 3D vegetation and city furniture data

Similar to the LOD2 building models, we collected the CityGML data for vegetation and city furniture from various regions. These CityGML datasets were then converted into OBJ files for use in the 3D virtual city generation. The 3D vegetation data included plant cover and individual vegetation objects, whereas the city furniture data included electricity poles, streetlights, and traffic lights. Table 2 presents the category names and their respective distributions.

3.3 DEM and OSM Data for Relief Generation

To enable the generation of relief, roads, and city furniture in the 3D city model, we used existing DEM data and corresponding OSM data within Japan as the foundation for the generation process.

The DEM data[3] were obtained from the Basic Geospatial Information download service provided by the Geospatial Information Authority of Japan (GSI), with resolutions varying by region (1m, 5m, or 10m). The OSM[4] is a community-driven project that provides free and editable geographic data worldwide. From the OSM, we used road and building footprint data corresponding to the same areas as the DEM data.

4. Methodology

To generate a 3D virtual city, the code repository in this study was pre-equipped with the necessary 2D and 3D data resources, including the DEM data (in TIFF format) referenced in Figure 4; OSM road and building footprint data (in GeoJSON format); and individual object data for buildings, vegetation, and city furniture (in OBJ format) obtained from PLATEAU. The entire workflow for generating the 3D virtual city is illustrated in Figure 4 and was implemented using Python 3.9.

For the foundation of an entire 3D virtual city, it is necessary to generate relief within a specified area. To facilitate the subsequent calculations on a unified scale, this study adopted a base coordinate reference system defined by the planar rectangular coordinate system used in Tokyo, Japan (EPSG:30169). The scale of the virtual city was controlled via input parameters with a default setting of 200 m × 200 m, as shown in Figure 4①. A starting point was randomly selected in the actual planar coordinate system from the DEM TIFF file, and the corresponding pixel range to be clipped was computed based on the DEM resolution. After clipping, each pixel in the result is converted to real-world 3D coordinates. Finally, the 3D coordinates are transformed into a Trimesh object using Delaunay triangulation with the Trimesh library. We use Trimesh for mesh processing due to its lightweight design, robust performance, and seamless support for common 3D formats. It provides essential functions such as mesh repair, normal computation, and geometric analysis, making it well-suited for automated model generation workflows. Its active development and compatibility with Python also ensure reproducibility and integration with other tools in our pipeline.

Similarly, as shown in Figure 4②, the road and building footprint data are cropped using the starting point and cropping range selected during the DEM processing and subsequently converted into GeoDataFrame objects to serve as the planar basis for generating the 3D virtual city. To generate 3D roads, the widths of the roads and pedestrian pathways were controlled through the input parameters. Based on these widths, buffers were added to the linear road elements, and pedestrian paths were created. Finally, an ear-clipping algorithm is used to convert the buffered road surfaces into Trimesh objects.

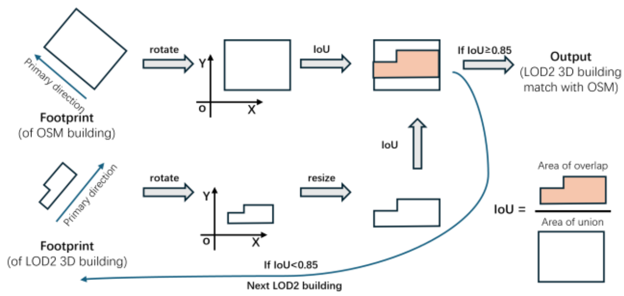

Figure 5. Detailed workflow of 3D building generation in virtual city.

The core component in generating a virtual 3D city is the creation of 3D buildings, as shown in Figure 4③. This part of the process employs a matching strategy based on pre-alignment, scale normalization, and Intersection over Union (IoU) to enhance the correspondence between building placements and the actual city layout, thereby producing more realistic and high-fidelity virtual city models. Building footprints were first extracted from the OSM data. To ensure consistency with the LOD2 building 3D data, the longest side of each building was used as the primary orientation during the preprocessing stage. Both the OSM footprints and LOD2 building data were then rotated so that their primary directions were aligned with the x-axis, and they were scaled according to their longest side to achieve a unified scale. For each OSM footprint, 100 buildings were randomly sampled from the LOD2 building 3D dataset, and the IoU was calculated sequentially for each building against the footprint. During the matching process, if the IoU of a building exceeded a preset threshold (default 0.85), the building was immediately selected and placed in the corresponding footprint, after which the process proceeded to the next footprint. If none of the sampled buildings met the

[2] https://service.gsi.go.jp/kiban/app
[3] https://www.openstreetmap.org/

This contribution has been peer-reviewed. The double-blind peer-review was conducted on the basis of the full paper.
https://doi.org/10.5194/isprs-annals-X-4-W6-2025-257-2025 | © Author(s) 2025. CC BY 4.0 License.

threshold, the building with the highest IoU was selected. The detailed workflow for building generation is illustrated in Figure 5. The matched building also considered the roof type ratio from the input parameters and selected the PLATEAU 3D building data from different roof types (Table 1).

For vegetation generation in a 3D virtual city, we controlled the placement using input parameters for vegetation density and high-to-low vegetation ratio. As shown in Figure 4④, within the DEM and OSM extent defined on the aforementioned planar coordinate system, a set number of points are randomly sampled based on the density parameter. Subsequently, high (⩾ 6 m) and low (< 6 m) PLATEAU 3D vegetation models were placed at the sampling points according to the specified ratio.

Subsequently, for city furniture elements such as traffic signals, streetlights, and utility poles (as shown in Figure 4), this study controls their generation via an input parameter for city furniture density. Specifically, a PLATEAU 3D city furniture model was placed at regular intervals along the road edges. At this point, a high-fidelity virtual 3D city comprising terrain, roads, buildings, vegetation, and city furniture elements is fully generated.

Finally, all elements except the terrain are vertically translated along the Z-axis to align with the terrain's elevation. Overlap detection is then conducted following the priority order of roads, buildings, vegetation, and city furniture, and any overlapping element instances are removed. Lastly, the various element components, represented as Trimesh objects, are exported into a CityGML file in accordance with the CityGML 3.0 specifications.

5. Experiments and Results

In this study, the virtual 3D city model generation operated solely on the CPU (AMD Ryzen 7 7745HX), achieving a generation speed of five buildings per second, and about 30s for a 200m x 200m tile. The generated content was configurable using the input parameters, and the controllable items are listed in Table 2.

Based on the control of the various input parameters listed in Table 2, the virtual 3D city model generation can be tailored to meet diverse requirements and ultimately exported to a

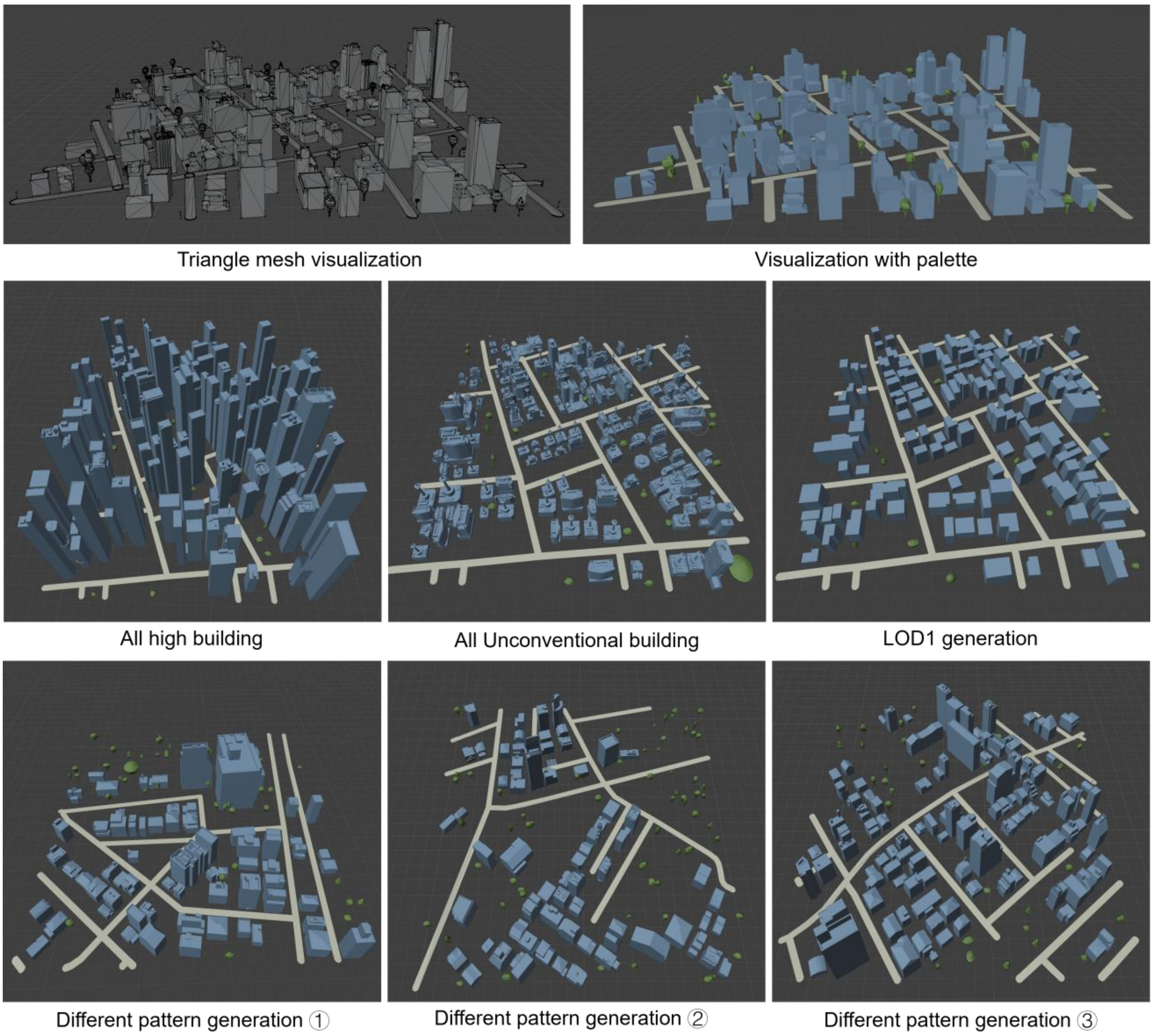

Figure 6. Generation examples of virtual 3D city model generation.

CityGML file. The generated results are shown in Figure 6, which includes cases produced under different visualization settings: models in which all building stories are adjusted to high-rise (10–30 floors); an extreme scenario in which the entire area features buildings with unconventional roof types, LOD1 models, as well as models with different styles generated under three distinct random seed conditions. The absolute positions of the generated elements were fixed when the random seed was fixed. Additional configurable features not shown in Figure 6 include the widths of the main roads and sidewalks, vegetation density, density and ratio of traffic signals to utility poles, and the proportion of high to low vegetation. This study generates cities with specified dimensions based on the spatial distribution logic of real-world features. As shown in Figure 7, even a 500 m × 500 m area can be generated with a quality level that closely approximates that of the real world.

The final output of this study was a CityGML file. After generating the 3D models for relief, roads, buildings, vegetation, and city furniture, each Trimesh object was converted into a structured XML representation using etree. For every individual feature, the process begins by creating a “cityObjectMember” node, within which a “MultiSurface” element and its child “surfaceMember” are established. Within the “surfaceMember,” a “Polygon” is generated and its boundary is described using “exterior” and “LinearRing” elements. The coordinates of all vertices are then sequentially written into a “posList” element, with the first vertex appended at the end to ensure the polygon is closed. Finally, these structured XML elements are organized according to the feature category and output to the CityGML file.

Table 2. Controllable features of virtual 3D city model generation.

Features	Range	Default
Random Seed	[0, 65535], int	1024
Building LOD	{0, 1, 2}, int	2
Min Floor Num	[1, 50], int	1
Max Floor Num	[1, 50], int	6
Flat Roof Ratio	[0.0, 1.0], float	0.2
Step. Roof Ratio	[0.0, 1.0], float	0.3
Com. Roof Ratio	[0.0, 1.0], float	0.3
Hip Roof Ratio	[0.0, 1.0], float	0.0
Gable Roof Ratio	[0.0, 1.0], float	0.2
Unc. Roof Ratio	[0.0, 1.0], float	0.0
Road LOD	{0, 1, 2}, int	2
Main Road width	[1.0, 10.0], float	2.0
Sub Road width	[1.0, 10.0], float	0.5
Vegetation LOD	{0, 1, 2}, int	2
Low Tree Ratio	[0.0, 1.0], float	0.5
High Tree Ratio	[0.0, 1.0], float	0.5
City Fur. LOD	{0, 1, 2}, int	2
Tele. Pole Ratio	[0.0, 1.0], float	0.5
Traf. Light Ratio	[0.0, 1.0], float	0.5
Relief LOD	{0, 1}, int	1
Output Path	string	-

6. Conclusion and Discussion

Unlike traditional digital twin approaches that require an exact replication of every detail of a real city, this study is inspired by the concept of the "digital cousin." This concept embodies the idea that in the transformation from physical to digital, it is not necessary for digital models to fully reproduce all the details of real entities. Instead, the digital cousins approach employs a flexible and broad methodology to digitally capture only the key

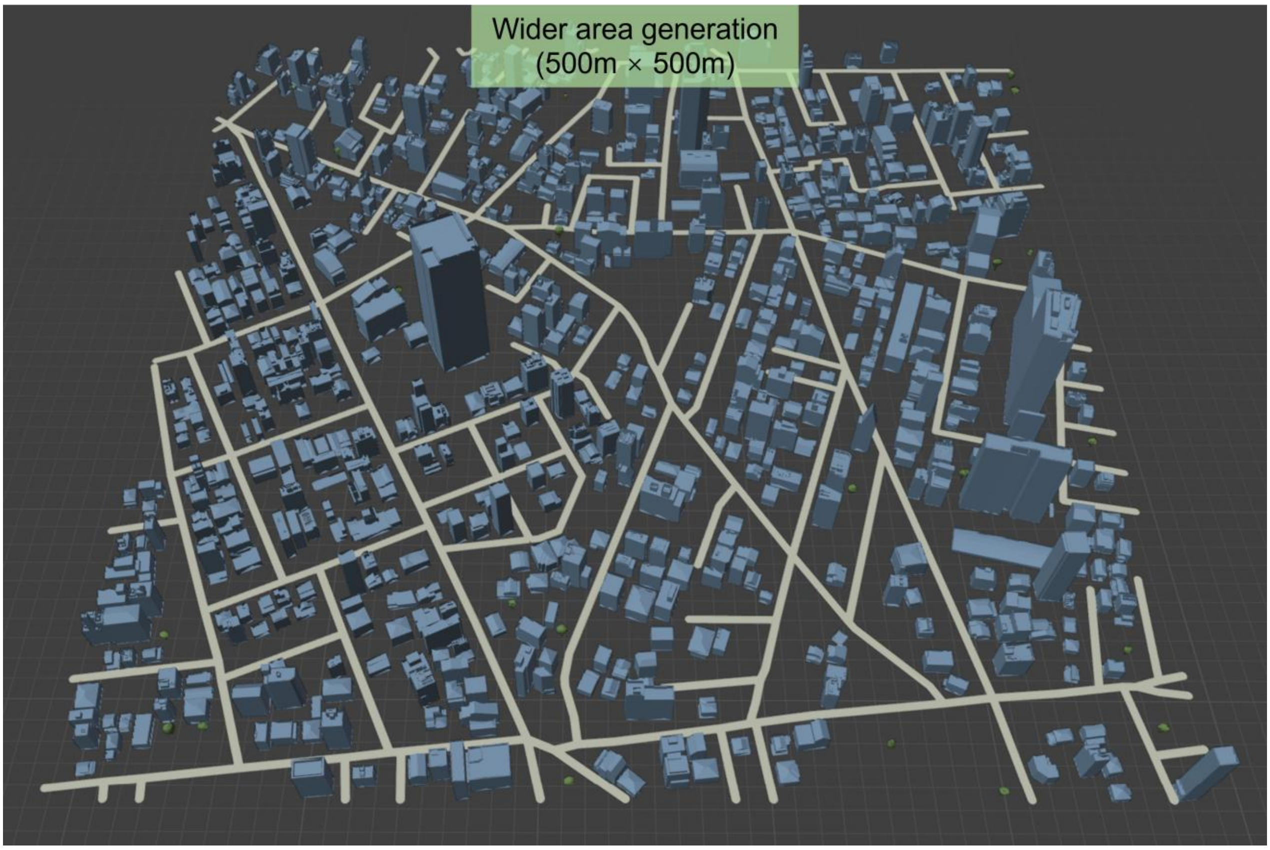

Figure 7. Generation example of 500 m × 500 m area.

features, thereby achieving 3D digital reconstruction of a city at a cost far lower than that of traditional methods. Based on open-source data, this study proposes a virtual 3D city model generation method that produces a high-fidelity CityGML output with a focus on the judicious sampling and reconstruction of real-world elements.

This study leveraged the spatial distribution information of actual physical features, such as DEM data and OSM-based terrain, roads, and building locations, while integrating open-source 3D model assets (e.g., LOD2 building models, vegetation, and urban facilities). Consequently, a virtual 3D city model covering an area of 40,000 square meters within 1 min. The produced model not only faithfully replicates the spatial distribution of real-world elements but also allows for flexible adjustment of key features, including buildings, roads, vegetation, and city furniture (as listed in Table 2), through input parameters, thereby meeting the research and application requirements in urban planning (Biljecki et al., 2015; Chen, 2011; Willenborg et al., 2017), environmental (Deininger et al., 2020) and (Nouvel et al., 2013) , disaster management (Riaz et al., 2023) and (Armand et al., 2021), and GIS analysis (Khayyal et al., 2022).

However, this study had certain limitations in achieving a digital replica of the real world. First, the generated building footprints did not fully correspond to the actual conditions, and the building height information was not reflected in the model. Second, key elements such as road widths and roadside traffic signs are not adequately represented. Additionally, accurately representing street trees and residential greenery in 3D models remains an unresolved challenge.

Considering these shortcomings, future research can be improved in the following ways.

For building models, future work could incorporate generative AI techniques (S. Chen et al., 2024; Siddiqui et al., 2024) in addition to the current generation approach to perform autoregressive training on 3D mesh objects, thereby predicting and replicating building details more accurately, particularly building height information. This can be achieved by integrating building morphology (Y. Chen et al., 2024) methods that combine the shape of the footprint with neighbourhood information.

For issues related to roads and traffic signage, semantic-segmentation-based techniques (Wang et al., 2023) can be explored to accurately extract road widths and the distribution of roadside facilities, thereby enhancing the level of detail and practical utility of the model.

For vegetation, further leveraging high-precision remote sensing data or deep learning techniques (Wang et al., 2023) can improve the representation of street trees and residential greenery in a 3D model, ensuring both realism and accuracy.

In summary, guided by the concept of digital cousins, this study presents a high-fidelity virtual 3D city model generation approach based on open-source data. This method achieves a low-cost and highly efficient digital reconstruction of urban environments while providing robust support for further in-depth research and practical applications. Future work will further expand and refine the capabilities of each submodule to satisfy the stringent requirements for digital twins in the fields of urban planning and digital governance.

Acknowledgements

Acknowledgements of support for the project/paper/author are welcome. Note, however, that for the paper to be submitted for review all acknowledgements must be anonymized.

References

Armand, P., Oldrini, O., Duchenne, C., & Perdriel, S. (2021). Topical 3D modelling and simulation of air dispersion hazards as a new paradigm to support emergency preparedness and response. *Environmental Modelling & Software*, *143*, 105129.

Arroyo Ohori, K., Biljecki, F., Kumar, K., Ledoux, H., & Stoter, J. (2018). Modeling cities and landscapes in 3D with CityGML. *Building Information Modeling: Technology Foundations and Industry Practice*, 199-215.

Biljecki, F., Ledoux, H., & Stoter, J. (2016). Generation of multi-LOD 3D city models in CityGML with the procedural modelling engine Random3Dcity.

Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S., & Çöltekin, A. (2015). Applications of 3D city models: State of the art review. *ISPRS International Journal of Geo-Information*, *4*(4), 2842-2889.

Chen, R. (2011). The development of 3D city model and its applications in urban planning. 2011 19th International Conference on Geoinformatics,

Chen, S., Chen, X., Pang, A., Zeng, X., Cheng, W., Fu, Y., Yin, F., Wang, B., Yu, J., & Yu, G. (2024). Meshxl: Neural coordinate field for generative 3d foundation models. *Advances in neural information processing systems*, *37*, 97141-97166.

Chen, Y., Sun, W., Yang, L., Yang, X., Zhou, X., Li, X., Li, S., & Tang, G. (2024). Refining urban morphology: An explainable machine learning method for estimating footprint-level building height. *Sustainable Cities and Society*, *112*, 105635.

Dai, T., Wong, J., Jiang, Y., Wang, C., Gokmen, C., Zhang, R., Wu, J., & Fei-Fei, L. (2024). Automated creation of digital cousins for robust policy learning. *arXiv preprint arXiv:2410.07408*.

Deininger, M. E., von der Grün, M., Piepereit, R., Schneider, S., Santhanavanich, T., Coors, V., & Voß, U. (2020). A continuous, semi-automated workflow: From 3D city models with geometric optimization and CFD simulations to visualization of wind in an urban environment. *ISPRS International Journal of Geo-Information*, *9*(11), 657.

Deng, J., Chai, W., Guo, J., Huang, Q., Hu, W., Hwang, J.-N., & Wang, G. (2023). Citygen: Infinite and controllable 3d city layout generation. *arXiv preprint arXiv:2312.01508*.

Deng, J., Chai, W., Huang, J., Zhao, Z., Huang, Q., Gao, M., Guo, J., Hao, S., Hu, W., & Hwang, J.-N. (2024). Citycraft: A real crafter for 3d city generation. *arXiv preprint arXiv:2406.04983*.

Früh, C., & Zakhor, A. (2004). An automated method for large-scale, ground-based city model acquisition. *International Journal of Computer Vision*, *60*, 5-24.

Goetz, M. (2013). Towards generating highly detailed 3D CityGML models from OpenStreetMap. *International Journal of Geographical Information Science*, *27*(5), 845-865.

Gröger, G., & Plümer, L. (2012). CityGML–Interoperable semantic 3D city models. *ISPRS Journal of Photogrammetry and Remote Sensing*, *71*, 12-33.

Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in neural information processing systems*, *33*, 6840-6851.

Isikdag, U., & Zlatanova, S. (2009). Towards defining a framework for automatic generation of buildings in CityGML using building Information Models. In *3D geo-information sciences* (pp. 79-96). Springer.

Kerbl, B., Kopanas, G., Leimkühler, T., & Drettakis, G. (2023). 3d gaussian splatting for real-time radiance field rendering. *ACM Trans. Graph.*, *42*(4), 139:131-139:114.

Khayyal, H. K., Zeidan, Z. M., & Beshr, A. (2022). Creation and spatial analysis of 3D city modeling based on GIS data. *Civil Engineering Journal*, *8*(1), 105.

Kolbe, T. H., Gröger, G., & Plümer, L. (2005). CityGML: Interoperable access to 3D city models. In *Geo-information for disaster management* (pp. 883-899). Springer.

Lin, C. H., Lee, H.-Y., Menapace, W., Chai, M., Siarohin, A., Yang, M.-H., & Tulyakov, S. (2023). Infinicity: Infinite-scale city synthesis. Proceedings of the IEEE/CVF international conference on computer vision,

Malhotra, A., Shamovich, M., Frisch, J., & van Treeck, C. (2022). Urban energy simulations using open CityGML models: A comparative analysis. *Energy and Buildings*, *255*, 111658.

Mildenhall, B., Srinivasan, P. P., Tancik, M., Barron, J. T., Ramamoorthi, R., & Ng, R. (2021). Nerf: Representing scenes as neural radiance fields for view synthesis. *Communications of the ACM*, *65*(1), 99-106.

Murtiyoso, A., Veriandi, M., Suwardhi, D., Soeksmantono, B., & Harto, A. B. (2020). Automatic Workflow for roof extraction and generation of 3D citygml models from low-cost uav image-derived point clouds. *ISPRS International Journal of Geo-Information*, *9*(12), 743.

Nouvel, R., Schulte, C., Eicker, U., Pietruschka, D., & Coors, V. (2013). CityGML-based 3D city model for energy diagnostics and urban energy policy support. Building Simulation 2013,

Over, M., Schilling, A., Neubauer, S., & Zipf, A. (2010). Generating web-based 3D City Models from OpenStreetMap: The current situation in Germany. *Computers, Environment and urban systems*, *34*(6), 496-507.

Peters, R., Dukai, B., Vitalis, S., van Liempt, J., & Stoter, J. (2022). Automated 3D reconstruction of LoD2 and LoD1 models for all 10 million buildings of the Netherlands. *Photogrammetric Engineering & Remote Sensing*, *88*(3), 165-170.

Prandi, F., De Amicis, R., Piffer, S., Soave, M., Cadzow, S., Gonzalez Boix, E., & D'hont, E. (2013). Using CityGML to deploy smart-city services for urban ecosystems. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *40*, 87-92.

Riaz, K., McAfee, M., & Gharbia, S. S. (2023). Management of climate resilience: exploring the potential of digital twin technology, 3D city modelling, and early warning systems. *Sensors*, *23*(5), 2659.

Shang, Y., Lin, Y., Zheng, Y., Fan, H., Ding, J., Feng, J., Chen, J., Tian, L., & Li, Y. (2024). UrbanWorld: An Urban World Model for 3D City Generation. *arXiv preprint arXiv:2407.11965*.

Siddiqui, Y., Alliegro, A., Artemov, A., Tommasi, T., Sirigatti, D., Rosov, V., Dai, A., & Nießner, M. (2024). Meshgpt: Generating triangle meshes with decoder-only transformers. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,

Singh, S. P., Jain, K., & Mandla, V. R. (2013). Virtual 3D city modeling: techniques and applications. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *40*, 73-91.

Tan, Y., Liang, Y., & Zhu, J. (2023). CityGML in the Integration of BIM and the GIS: Challenges and Opportunities. *Buildings*, *13*(7), 1758.

Wang, J., Lawson, G., & Shen, Y. (2014). Automatic high-fidelity 3D road network modeling based on 2D GIS data. *Advances in Engineering Software*, *76*, 86-98.

Wang, W., Dai, J., Chen, Z., Huang, Z., Li, Z., Zhu, X., Hu, X., Lu, T., Lu, L., & Li, H. (2023). Internimage: Exploring large-scale vision foundation models with deformable convolutions. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,

Willenborg, B., Sindram, M., & Kolbe, T. H. (2017). Applications of 3D city models for a better understanding of the built environment. *Trends in spatial analysis and modelling: decision-support and planning strategies*, 167-191.

Xie, H., Chen, Z., Hong, F., & Liu, Z. (2024a). Citydreamer: Compositional generative model of unbounded 3d cities. Proceedings of the IEEE/CVF conference on computer vision and pattern recognition,

Xie, H., Chen, Z., Hong, F., & Liu, Z. (2024b). GaussianCity: Generative Gaussian splatting for unbounded 3D city generation. *arXiv preprint arXiv:2406.06526*.

Xu, L., Xiangli, Y., Peng, S., Pan, X., Zhao, N., Theobalt, C., Dai, B., & Lin, D. (2023). Grid-guided neural radiance fields for large urban scenes. Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition,

Yao, Z., Nagel, C., Kunde, F., Hudra, G., Willkomm, P., Donaubauer, A., Adolphi, T., & Kolbe, T. H. (2018). 3DCityDB-a 3D geodatabase solution for the management, analysis, and visualization of semantic 3D city models based on CityGML. *Open Geospatial Data, Software and Standards*, *3*(1), 1-26.