Automatic Generation of LOD3 Building Models for High-Density Cities: A Case Study of Hong Kong Using Multi-Source Data and an Adaptive Strategy

Guoxin Si[1], Wei Yao[2,3]

[1] Department of Land Surveying and Geo-Informatics, The Hong Kong Polytechnic University, Hung Hom, Hong Kong – m15624930071@163.com
[2] Spatial Intelligence and Urban Computing, Institute of Urban Environment, Chinese Academy of Sciences, Xiamen, China - wyao@iue.ac.cn
[3] State Key Lab for Ecological Security of Regions and Cities, Institute of Urban Environment, Chinese Academy of Sciences, Xiamen, China

Keywords: High-Density Urban Reconstruction, Multi-Source Data Fusion, LOD3 Modeling, Building Components.

Abstract

This paper presents a framework for upgrading LOD2 building models to LOD3 by directly exploiting texture information embedded in existing LOD2 models. A semi-automated annotation tool based on SAM2 enables instance-level segmentation of facade components from texture images, with 2D bounding boxes back-projected to 3D coordinates via UV mapping and barycentric interpolation. Recognized textures serve as queries for two parallel pipelines: metric learning with DINOv2 retrieves similar components from a curated BIM library, while TripoSR generates 3D components directly from single texture images. All retrieved or generated components are adaptively scaled and fused with the LOD2 model using Boolean operations based on constructive solid geometry. Experimental results demonstrate that the metric learning approach achieves 81.2% Precision@1 and 46.6% MAP@k with SubCenter ArcFace loss, suggesting the effectiveness of transferring DINOv2 features to BIM component retrieval. Although TripoSR-based generation is currently limited by data scarcity and input texture quality, it shows promising potential as a complementary pathway. The proposed framework offers a scalable and cost-effective solution for automated LOD3 reconstruction, contributing to high-fidelity digital twins in complex urban environments.

1. Introduction

Detailed 3D building models are fundamental to digital twins and smart cities, enabling applications such as urban planning, microclimate simulation, and energy analysis (Biljecki et al., 2015). Within CityGML (Consortium et al., 2012), the Level of Detail 3 (LOD3) models incorporate elements of the facade, such as windows and doors, offering the semantic richness required for fine-grained analytics. However, despite the widespread availability of city-wide LOD2 repositories (Wysocki et al., 2023), automatic LOD3 generation remains challenging, particularly in high-density cities like Hong Kong, where complex geometries and severe occlusions hinder automated processes.

Existing approaches generally fall into two categories. Multi-modal methods fuse mobile laser scanning (MLS) point clouds with images and LOD2 priors, as demonstrated by Scan2LoD3 (Wysocki et al., 2023) and the Grounding DINO-based pipeline of (Yarroudh et al., 2024). Although effective, these methods are costly and suffer from data shadowing in dense urban canyons. Image-based methods leverage street-view imagery to reduce acquisition costs, as shown by (Pang and Biljecki, 2022) and the Texture2LoD3 framework (Tang et al., 2025). However, they remain sensitive to occlusions and lighting variations.

In this study, we propose a novel approach that directly exploits the texture information embedded in existing LOD2 models. A dedicated desktop annotation program automatically identifies facade components from textures, and a parameterized retrieval mechanism searches a pre-built BIM library (Shao et al., 2024) for matching component models using texture features as queries, inspired by the texture library concept of (Lu et al., 2018). When no match is found, a generative module synthesizes new components. All retrieved or generated components are adaptively scaled and seamlessly fused into the original LOD2 framework. This study makes three key contributions:

1. A practical workflow for upgrading from LOD2 to LOD3. We present a streamlined methodology that directly utilizes the texture maps of existing LOD2 models to drive the reconstruction of LOD3 details. By integrating Segment Anything Model 2 (SAM2) with a UV-mapping-based back-projection algorithm, we establish a robust link between 2D image semantics and 3D geometry. This approach avoids the complexities of external data registration and provides a cost-effective way to incrementally upgrade city-scale models.

2. **SAM2-assisted semi-automated annotation.** We develop a desktop tool that leverages SAM2 to enable instance-level segmentation of facade components from LOD2 textures with minimal human input, addressing the labor-intensive bottleneck in existing template-based approaches.

3. **Dual-path component acquisition pipeline.** We implement a metric learning-based retrieval system that queries annotated textures against a pre-built BIM library (115 components). For unmatched cases, a TripoSR-based generative module produces candidate 3D models, serving as a complementary pathway pending further refinement for direct integration.

2. Related Works

2.1 Multi-source Data-based Upgrading

Upgrading existing LOD2 models using street-level data has proven to be an efficient and scalable pathway to LOD3, as it takes advantage of the ubiquity of city-wide LOD2 repositories (Yarroudh et al., 2024). Mobile Laser Scanning (MLS) point clouds have been widely adopted for this purpose due to their high geometric fidelity (Fang et al., 2025). Early works relied on geometric analysis, such as voxel-based segmentation and conditional clustering, to detect facade elements such as windows (Xu et al., 2018, Hoegner and Gleixner, 2022, Wang et al., 2025). To overcome the limitations of pure geometry, multi-modal methods emerged. The Scan2LoD3 framework, for instance, fuses point cloud data with image segmentation using ray-casting and Bayesian networks to reconstruct CityGML-compliant LOD3 models (Wysocki et al., 2023). Despite its sophistication, this approach demands highly accurate data registration and multi-modal inputs(Xu et al., 2017, Polewski and Yao, 2019), which limits its scalability (Tang et al., 2025).

2.2 2D Image-based Detection

Given the challenges associated with 3D data, many researchers have turned to 2D imagery for facade understanding. For example, refining existing models from single street-view images has been shown to yield better geometric accuracy than reconstruction from scratch (Pang and Biljecki, 2022). In another work, a modified Faster R-CNN was employed to detect facade details to enhance LOD2 models, although the method was constrained by the scarcity of depth-annotated datasets (Hensel et al., 2019, Yarroudh et al., 2024). The advent of foundational vision models has opened new possibilities. An automated pipeline projects MLS point clouds onto 2D images, applies the open-set detector Grounding DINO to identify windows and doors, and back-projects the detections into 3D space to upgrade LOD2.2 models to LOD3 (Yarroudh et al., 2024). Currently, Texture2LoD3 uses ubiquitous panoramic street-view images and existing LOD2 priors; by automatically adjusting panoramas and employing Semantic-SAM and CLIP for facade segmentation, it achieves accurate texturing and element detection without requiring MLS data (Tang et al., 2025).

2.3 Template-based Generation

Another class of methods uses prior knowledge and parametric templates to generate or complete building components. A representative workflow, developed by the Zurich ETH team, centers on the concept of “measurement macros” to allow efficient replication of repetitive facade elements (e.g. windows) through a copy-and-paste mechanism, effectively filling data gaps caused by occlusions (Gruen et al., 2020). Their system integrates interactive editing with procedural templates to correct and extend automatically generated models. In the context of BIM and GIS integration, an image-driven fuzzy system has been proposed to construct as-is IFC objects (Lu et al., 2018). A key innovation of this approach is an extensible texture library: by matching texture features extracted from images against the library, the system retrieves and assigns appropriate material types to recognized building elements, demonstrating the potential of texture as a semantic clue for component retrieval.

3. Methodology

As shown in Figure 1, our LOD3 generation process begins with an input LOD2-level BIM model. First, the Segment Anything Model 2 (SAM2) model is used to annotate the texture files of the model (Ravi et al., 2024). Based on the mapping between texture coordinates and vertex coordinates in the OBJ file, the 3D bounding box coordinates of the target components are back-calculated, and the texture images are cropped accordingly to obtain the component textures. The workflow then diverges into two parallel pipelines for the generation of candidate BIM components: one pipeline leverages metric learning - specifically, transfer training with DINOv2 (Oquab et al., 2023)—to retrieve the most similar components from a BIM library, while the other pipeline uses the TripoSR network (Tochilkin et al., 2024) to directly generate 3D components from cropped textures. Concurrently, point cloud data are extracted from the region defined by the 3D bounding box to compute the thickness of elements such as doors and windows. The candidate components produced by the two pipelines, together with the estimated thickness parameters, undergo parametric transformations to precisely fit the target bounding boxes. Finally, set operations are applied to merge the transformed components into the original BIM model, thus completing the model update.

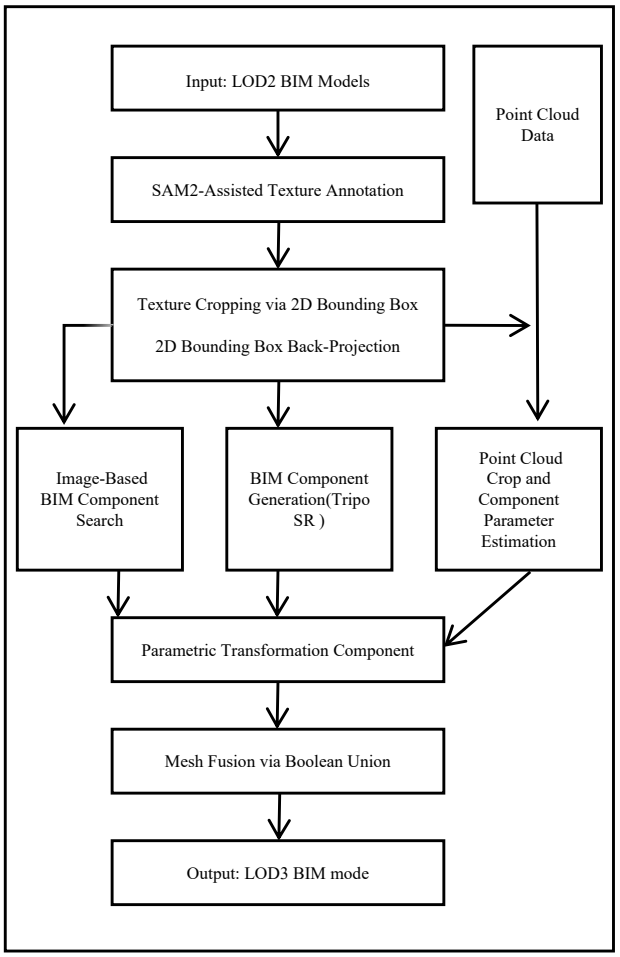

Figure 1. Flow diagram.

3.1 SAM2-Assisted Texture Annotation

This study employs SAM2 as the primary annotation tool for instance-level labeling of architectural texture images, following a semi-automated human-in-the-loop workflow. SAM2, a vision foundation model developed by Meta AI (Kirillov et al., 2023, Ravi et al., 2024), facilitates direct segmentation of building components (e.g., doors, windows, balconies) without requiring scene-specific transfer training. Human annotators provide few prompts to guide the model, substantially reducing manual effort while preserving annotation quality. This balanced integration of automation and human oversight ensures reliable texture labels for the reconstruction of the downstream LOD3 model.

3.2 2D Bounding Box Back-Projection

Given a 2D bounding box $\mathbf{B}$ annotated on a texture image, the goal is to compute its corresponding 3D position $\mathbf{P} \subset \mathbf{R}^3$. The mapping process consists of the following steps (Waechter et al., 2014).

First, the pixel coordinates (p_x, p_y) within the bounding box are converted to UV coordinates (u, v) as:

$$u = \frac{p_x}{W}; v = 1 - \frac{p_y}{H}, \tag{1}$$

where W and H are the width and height of the 2D bounding box, respectively. The v-coordinate is flipped because the UV coordinate system originates at the lower-left corner of the texture image.

For each sampled UV point (u, v), the triangular facet that contains it in UV space is identified. Let the UV-space vertices of the triangle be $\mathbf{A}_{uv}, \mathbf{B}_{uv}, \mathbf{C}_{uv}$. The barycentric coordinates (α, β, γ) are computed so that:

$$(u, v) = \alpha\mathbf{A}_{uv} + \beta\mathbf{B}_{uv} + \gamma\mathbf{C}_{uv} \tag{2}$$

Using the same barycentric weights, the corresponding 3D point $\mathbf{P}_{3d}$ is obtained by interpolating the 3D vertices $\mathbf{A}_{3d}, \mathbf{B}_{3d}, \mathbf{C}_{3d}$ of the same triangle:

$$\mathbf{P}_{3d} = \alpha\mathbf{A}_{3d} + \beta\mathbf{B}_{3d} + \gamma\mathbf{C}_{3d} \tag{3}$$

The four corners of the 2D bounding box—top-left (x, y), top-right $(x+w, y)$, bottom-right $(x+w, y+h)$, and bottom-left $(x, y+h)$—are sequentially mapped to 3D points $\mathbf{P}_0$ through $\mathbf{P}_3$, which are then connected to form a rectangular 3D region.

This mapping is implemented by parsing the OBJ file to extract vertex coordinates (format $v\ x\ y\ z$), texture coordinates (format $vt\ u\ v$) and face definitions (format $f\ v_1/vt_1/vn_1\ v_2/vt_2/vn_2\ v_3/vt_3/vn_3$), and by constructing efficient triangle data structures for the location of the points in the UV space.

3.3 Metric Learning for Image-based BIM Component Search

We employ a metric learning framework for BIM model image retrieval, adopting a transfer learning strategy, and leveraging pre-trained visual models for efficient feature extraction.

The core idea is to map the input images into a high-dimensional feature space where feature vectors of similar samples are close together and those of dissimilar samples are far apart. Specifically, DINOv2 is used as the backbone network to extract image features, followed by several trainable fully connected layers for dimension adjustment; end-to-end training is performed with a combination of metric learning loss functions.

DINOv2 is a visual Transformer proposed by Meta AI Research (Oquab et al., 2023), pre-trained on large-scale unlabeled images via self-distillation. Compared to traditional supervised models, the visual features extracted by DINOv2 exhibit stronger generalization capabilities and achieve excellent performance in various downstream tasks.

We adopt a transfer learning strategy: the pre-trained weights of DINOv2 are frozen, and only the subsequent fully connected layers are trained. This approach preserves the general visual features learned from large-scale datasets, effectively reducing the risk of overfitting, while significantly decreasing the number of trainable parameters and lowering computational cost and training time.

Various metric learning loss functions are integrated for model optimization, covering both angular margin-based losses and sample-pair-based losses. For the category based on angular margins, we employ ArcFace, CosFace, and their sub-center variant SubCenter ArcFace (Deng et al., 2019, Deng et al., 2020, Wang et al., 2018), which introduce an additive angular margin between normalized feature vectors and weight vectors to improve inter-class separability. Taking ArcFace as an example, its loss function is defined as:

$$\mathcal{L}_{\text{ArcFace}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s\cos(\theta_{y_i}+m)}}{e^{s\cos(\theta_{y_i}+m)} + \sum_{j\neq y_i} e^{s\cos\theta_j}} \tag{4}$$

where θ_j denotes the angle between the feature vector of the i-th sample and the weight vector of class j, m is the additive angular margin and s is a scaling factor.

To evaluate the retrieval performance of the proposed method, this study employs two widely used metrics in the information retrieval community: Mean Average Precision at k (MAP@k) and Precision@1 (Ramzi et al., 2025, Manzhos et al., 2025). For a given query q, let the retrieved list have length k, and denote the set of relevant documents as Ω^+, with $|\Omega^+|$ being the total number of relevant documents. The precision at position i is defined as:

$$P@i = \frac{\sum_{j=1}^{i}\text{rel}(j)}{i} \tag{5}$$

where $\text{rel}(j) \in \{0, 1\}$ indicates whether the j-th retrieved item is relevant. The Average Precision at k for a single query is then defined as:

$$\text{AP@}k = \frac{1}{\min(|\Omega^+|, k)}\sum_{i=1}^{k} P@i \cdot \text{rel}(i) \tag{6}$$

MAP@k is obtained by averaging AP@k over all queries Q:

$$\text{MAP@}k = \frac{1}{|Q|} \sum_{q \in Q} \text{AP@}k(q) \tag{7}$$

Precision@1 measures the relevance of the top-ranked result, defined as:

$$\text{Precision@1} = \frac{1}{|Q|} \sum_{q \in Q} \text{rel}(r_1) \tag{8}$$

where r_1 denotes the first item retrieved for the query q, and $\text{rel}(r_1) = 1$ if it is relevant and 0 otherwise.

These two metrics provide complementary evaluations of retrieval systems: MAP@k assesses the overall quality of the ranking, while Precision@1 focuses on the accuracy of the best prediction.

3.4 TripoSR for BIM Component Generation

For 3D component generation, we adopt *TripoSR* (Tochilkin et al., 2024)—a feedforward Transformer model based on the LRM (Hong et al., 2023) architecture that reconstructs high-quality 3D meshes from single RGB images in under 0.5 seconds. Its architecture comprises three main components: (i) a pre-trained DINOv1 (Caron et al., 2021) encoder for global and local feature extraction; (ii) an image-to-triplane decoder with self-attention and cross-attention Transformer layers that maps features to a compact triplane NeRF representation; and (iii) a lightweight MLP for predicting 3D color and density. During inference, volume rendering generates a volumetric representation from which the marching cubes algorithm extracts a textured 3D mesh.

3.5 Mesh Fusion via Boolean Union

To achieve seamless integration of generated 3D components with the LOD2 building model, we employ a Boolean union operation based on Constructive Solid Geometry (CSG) (Requicha, 1980). Let M_c denote the component mesh and M_l the LOD2 building mesh. The fused model is then expressed as the union of the two:

$$M_{\text{merged}} = M_c \cup M_l = \{\mathbf{p} \in \mathbf{R}^3 \mid \mathbf{p} \in M_c \text{ or } \mathbf{p} \in M_l\} \tag{9}$$

where $\mathbf{p}$ represents a point in three-dimensional Euclidean space. This operation preserves all exterior regions of both meshes while automatically handling the topological reconstruction of intersecting parts.

The implementation adopts the Binary Space Partitioning (BSP) algorithm(Naylor et al., 1990), which recursively partitions space to classify mesh triangles:

$$\text{Classify}(T, \triangle) = \begin{cases} \text{Front}, & d > \epsilon \\ \text{Back}, & d < -\epsilon \\ \text{Coplanar}, & \text{otherwise} \end{cases} \tag{10}$$

where T is the BSP tree, $\triangle$ is the triangle to be classified, d is the signed distance and $\epsilon = 10^{-6}$ is the numerical tolerance. The algorithm first constructs a BSP tree using triangles of the component mesh as splitting planes, then classifies triangles of the LOD2 mesh into front, back, or coplanar categories. Finally, it extracts the front-facing triangles and computes the intersecting edges to generate a closed manifold mesh.

Compared to voxel-based approximation methods, this strategy preserves a precise boundary representation and ensures topological correctness of the output, making it suitable for downstream applications such as 3D printing and physical simulation. The Boolean operations are performed in memory using the Computational Geometry Algorithms Library (CGAL), with OBJ format serving as the input/output container, requiring no modifications to the file format itself.

4. Experiments and Results

We implemented the proposed pipeline as an experimental desktop application developed in Python using `PyQt` following the Model–View–Controller (MVC) architectural pattern. The software runs on a workstation equipped with a 16-core CPU, 32 GB of RAM, and an NVIDIA RTX 3060 GPU. All computational tasks, including model training and inference, were performed locally on the same machine.

4.1 Software Interface Description

As shown in Figure 2, this study develops a Python-based software system that uses PyQt5 for the graphical user interface. It supports a complete workflow including BIM and point cloud loading, texture annotation, 3D back-projection, component matching and generation, and LOD3 merging. The main interface adopts a left-right split layout. The left panel organizes modules for file loading, point cloud control, SAM2 annotation, deep learning, editing, point cloud clipping, and export. The file loading module controls the loading, display, and clearing of BIM, point clouds, and textures. The point cloud control module enables adjustment of point size, display mode, and downsampling, etc. The SAM2 module supports point and box interaction, predefined categories (doors, windows, balconies), real-time instance display, texture extraction, and 2D-to-3D back-projection. Each annotated instance stores attributes such as ID, mask, category, and bounding box. Data export supports JSON annotation files and PNG textures, with 3D projections automatically saved as JSON. The deep learning module, corresponding to the component generation panel, controls the generation of BIM components via models. The editing module handles component modification and merging, the point cloud control module manages point cloud clipping and BIM parameter estimation, and the export module is responsible for exporting LOD3 BIM models.

The right side contains four side-by-side panels. The leftmost panel, based on PyVista, provides 3D visualization with BIM and point cloud overlay, interactive manipulation, and projection results shown as 3D rectangles and text labels. The second panel, built with OpenCV and PyQt5, enables high-resolution texture annotation with mouse-based selection and colored masks. The third panel obtains suitable BIM components via image matching or TripoSR and transforms them to the target bounding box location. The rightmost panel manages the BIM library, supporting component browsing, retrieval, and insertion.

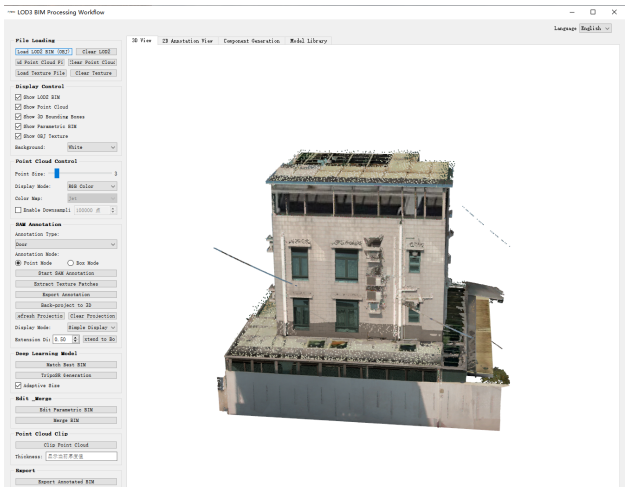

Figure 2. Software interface.

4.2 SAM2 Annotation and 3D Back-Projection

As shown in Figure 3, the software first loads the texture images exported from the LOD2 BIM model while preserving their original resolution. Through the interactive interface, the user provides SAM2 prompts by selecting a point or a box. In point selection mode, clicking several key points on the target component prompts the model to generate the corresponding instance segmentation mask; in box selection mode, drawing a rectangle around the target area enables SAM2 to perform fine segmentation within that region. The system automatically assigns a predefined category (e.g., door, window, balcony) to each segmented instance and computes its 2D bounding box coordinates. At any point, the user can click the “Back-project to 3D” button to convert the selected 2D bounding box into a 3D coordinate bounding box, which is then displayed alongside the LOD2 BIM model in the 3D view to facilitate three-dimensional visual verification.

4.3 Organization of Component BIM Library

To support metric learning, we constructed a BIM component dataset comprising three types of building elements: doors, windows, and balconies. The data comes from four publicly accessible platforms that offer free BIM models: SketchUp 3D Warehouse (`https://3dwarehouse.sketchup.com`), Free3D (`https://free3d.com`), BIMobject (`https://www.bimobject.com`) and Diwei (`https://www.3dwhere.com`). After downloading, the original model files were decompressed and uniformly converted to OBJ format to ensure data consistency. Each model was then individually inspected, with samples exhibiting geometric defects or missing textures removed, and the retained components were assigned standardized identifiers and names to form a well-structured BIM component dataset. In total, we collected 21 balcony, 94 doors, and 49 window components, which is 164 BIM components(Table 1). This dataset will serve as a foundation for feature extraction and similarity matching in subsequent metric learning tasks.

4.4 Metric Learning for BIM Component Retrieval

The experiments were conducted on a workstation equipped with a NVIDIA 3060 GPU (12GB memory) and implemented using the PyTorch framework. The DINOv2 model weights

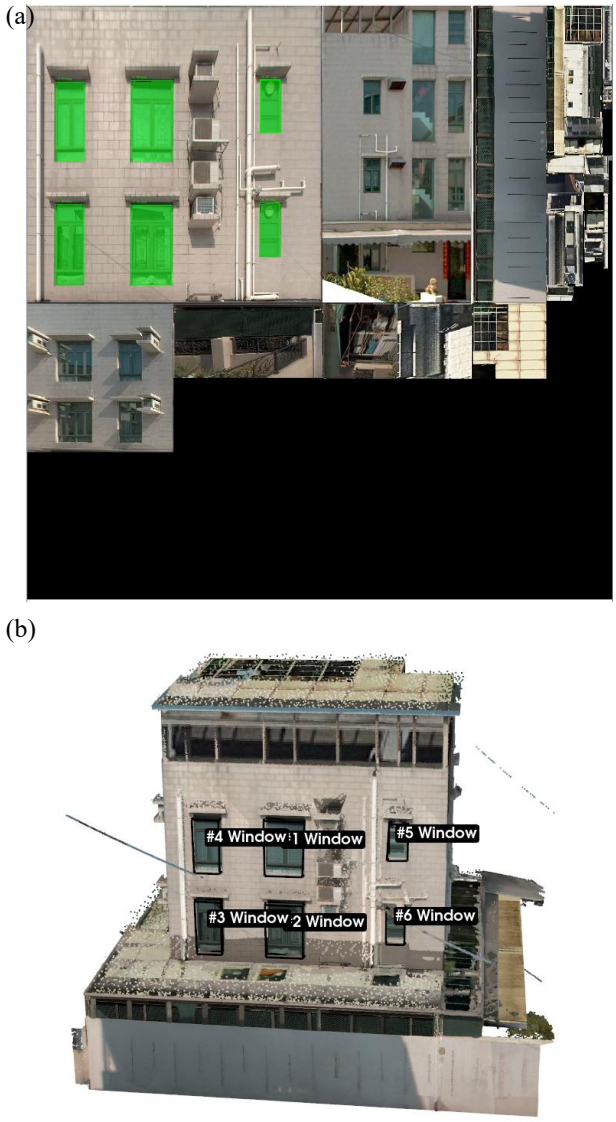

Figure 3. SAM2 annotation(a) and 3D back-projection(b).

were loaded from a local repository to avoid uncertainties associated with online downloads. Mixed precision was used during training to accelerate computation, and gradient clipping was applied to prevent gradient explosion. We evaluated using Precision@1 and MAP@k as defined in Section 3.3.

To avoid data redundancy, we performed a preliminary clustering of the 164 collected BIM components based on geometric features. For components with highly similar geometric structures, only the one with the best texture quality was retained, resulting in 115 representative components. To comprehensively capture three-dimensional geometric information, a multi-view rendering strategy was adopted: OBJ-format models were loaded using PyVista and rendered from six orthogonal perspectives (front, right, top, back, left, bottom). The bounding box was automatically calculated and the camera position and focal length were adjusted to ensure a proper proportion of the model in the rendered images. All rendered images were set to a resolution of 224×224 pixels and saved in PNG format to preserve texture and color information. The final training data set consisted of 115 BIM models, with six views per model, resulting in 690 images. These images were randomly split into

Component Type	Number
Balcony	21
Door	94
Window	49
Total	164

Table 1. Number of BIM components in the dataset.

non-overlapping training and test sets at an 8:2 ratio, and all images were normalized before being input into the model.

Loss Function	Precision@1	MAP@k
ArcFace Loss	79.3	42.6
CosFace Loss	74.2	22.8
SubCenter ArcFace Loss	81.2	46.6
MultiSimilarity Loss	74.9	35.2
Contrastive Loss	74.9	43.1

Table 2. Performance comparison of different loss functions.

Experimental results demonstrate that the transfer learning strategy of freezing DINOv2 weights and training only the fully connected layer proves effective in learning discriminative features of BIM models. Comparative experiments with different loss functions show that the SubCenter ArcFace loss achieved the best performance(Table 2), reaching 81.2% Precision@1 and 46.6% MAP@k in the test set, significantly outperforming other loss functions. ArcFace loss came second, with 79.3% Precision@1 and 42.6% MAP@k. CosFace loss, MultiSimilarity loss, and Contrastive loss performed relatively poorly, with CosFaceLoss achieving only 22.8% MAP@k, highlighting its limitations in handling the class distribution in this task. These results suggest that the general visual features extracted by DINOv2 can be effectively transferred to BIM model image retrieval tasks, and that SubCenter ArcFace loss, through its multi-subcenter mechanism, better captures intra-class diversity and inter-class discriminability, demonstrating clear advantages in class-imbalanced scenarios.

In terms of practical matching performance(Figure 4), the proposed transfer learning strategy has achieved good retrieval results and can relatively accurately identify the most similar BIM components. However, several issues remain to be addressed. On the one hand, the current BIM library is relatively small in scale and the component types are relatively limited. When the textures of cropped components vary significantly from those in the library, the matching results often lack practical reference value. Additionally, target component images cropped from LOD2 textures sometimes have low resolution, which may negatively affect matching accuracy. Therefore, this issue deserves focussed attention in future work.

4.5 TripoSR for BIM Component Generation

As a complementary approach, this study also explores a component generation pathway based on single-view 3D reconstruction, leveraging the TripoSR model to directly generate 3D BIM components from texture images. Based on the Transformer architecture, TripoSR reconstructs textured 3D meshes from a single RGB image in an end-to-end manner, offering fast inference and strong generalization capabilities. Due to the lack of high-quality 3D training data, we did not perform domain adaptation but directly applied the pretrained TripoSR model to the cropped texture images of architectural components. Experimental results indicate that issues such as low resolution and image blur in the input textures lead to notable gaps in geometric detail and texture fidelity between the generated meshes and

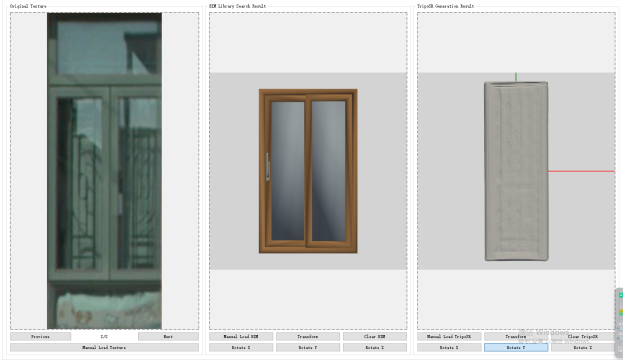

Figure 4. Results of BIM Component Retrieval and TripoSR-based 3D Reconstruction.

actual components, making them unsuitable for direct integration into LOD3 models. As shown in Figure 4, the generated window frames, for example, miss fine structural details. Additionally, the components targeted in this work are predominantly planar, whereas TripoSR tends to favor objects with more balanced proportions. Adapting TripoSR to better handle planar structures such as doors and windows thus represents a promising direction for future research.

Nevertheless, the image-to-3D pathway demonstrates significant potential in this task. As an important supplement to automated BIM component generation, it warrants continued exploration.

4.6 Mesh Fusion

The process of combining the LOD2 model with component BIM proceeds as follows: First, the texture image corresponding to the target bounding box is extracted and fed into both the metric learning model and TripoSR. The software then returns a list of candidate BIM components. The target bounding box is subsequently projected into the three-dimensional coordinate system, and the point cloud is cropped along the outward normal direction. The thickness of the component is estimated from the cropped point cloud, which, together with the length and width of the 3D bounding box, defines the dimensional parameters of the BIM component. The software automatically aligns the selected BIM component with the corresponding target bounding box location. For each target bounding box, fine adjustments such as rotation and mirroring can be applied. Once confirmed, the BIM component is merged with the LOD2 model via Boolean operations to generate an LOD3-level BIM model (Figure 5).

5. Conclusions

This paper presents a framework for upgrading LOD2 building models to LOD3 by leveraging texture information from existing models. A SAM2-assisted annotation tool enables instance-level texture segmentation, with 2D bounding boxes back-projected to 3D coordinates. A dual-path pipeline retrieves components from a BIM library via metric learning or generates them using TripoSR. The extracted or generated components are adaptively scaled and fused with the LOD2 model via Boolean operations.

Experimental results show that the metric learning approach achieves competitive retrieval accuracy, with SubCenter ArcFace loss outperforming other loss functions. TripoSR-based

Figure 5. Boolean fusion.

generation, while currently limited by data scarcity and input texture quality, demonstrates potential for synthesizing components. Future work will focus on expanding the BIM library, adapting TripoSR for planar structures, and integrating multi-view inputs to improve robustness.

References

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.

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., Joulin, A., 2021. Emerging properties in self-supervised vision transformers. *Proceedings of the IEEE/CVF international conference on computer vision*, 9650–9660.

Consortium, O. G. et al., 2012. OGC city geography markup language (CityGML) encoding standard. *Open geospatial consortium*.

Deng, J., Guo, J., Liu, T., Gong, M., Zafeiriou, S., 2020. Sub-center arcface: Boosting face recognition by large-scale noisy web faces. *European Conference on Computer Vision*, Springer, 741–757.

Deng, J., Guo, J., Xue, N., Zafeiriou, S., 2019. Arcface: Additive angular margin loss for deep face recognition. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 4690–4699.

Fang, L., Li, T., Lin, Y., Zhou, S., Yao, W., 2025. A coupled optical–radiometric modeling approach to removing reflection noise in TLS data of urban areas. *ISPRS Journal of Photogrammetry and Remote Sensing*, 220, 217-231.

Gruen, A., Schrotter, G., Schubiger, S., Qin, R., Xiong, B., Xiao, C., Li, J., Ling, X., Yao, S., 2020. An operable system for lod3 model generation using multi-source data and user-friendly interactive editing. Technical report, ETH Zurich.

Hensel, S., Goebbels, S., Kada, M., 2019. Facade reconstruction for textured LoD2 CityGML models based on deep learning and mixed integer linear programming. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 4, 37–44.

Hoegner, L., Gleixner, G., 2022. Automatic extraction of facades and windows from MLS point clouds using voxelspace and visibility analysis. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 43, 387–394.

Hong, Y., Zhang, K., Gu, J., Bi, S., Zhou, Y., Liu, D., Liu, F., Sunkavalli, K., Bui, T., Tan, H., 2023. Lrm: Large reconstruction model for single image to 3d. *arXiv preprint arXiv:2311.04400*.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y. et al., 2023. Segment anything. *Proceedings of the IEEE/CVF international conference on computer vision*, 4015–4026.

Lu, Q., Lee, S., Chen, L., 2018. Image-driven fuzzy-based system to construct as-is IFC BIM objects. *Automation in Construction*, 92, 68–87.

Manzhos, T., Ianevych, T., Melnyk, O., 2025. Average Precision at Cutoff k under Random Rankings: Expectation and Variance. *arXiv preprint arXiv:2511.02571*.

Naylor, B., Amanatides, J., Thibault, W., 1990. Merging BSP trees yields polyhedral set operations. *ACM Siggraph Computer Graphics*, 24(4), 115–124.

Oquab, M., Darcet, T., Moutakanni, T., Vo, H., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A. et al., 2023. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*.

Pang, H. E., Biljecki, F., 2022. 3D building reconstruction from single street view images using deep learning. *International Journal of Applied Earth Observation and Geoinformation*, 112, 102859.

Polewski, P., Yao, W., 2019. Scale invariant line-based co-registration of multimodal aerial data using L1 minimization of spatial and angular deviations. *ISPRS Journal of Photogrammetry and Remote Sensing*, 152, 79-93.

Ramzi, E., Audebert, N., Rambour, C., Araujo, A., Bitot, X., Thome, N., 2025. Optimization of rank losses for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Ravi, N., Gabeur, V., Hu, Y.-T., Hu, R., Ryali, C., Ma, T., Khedr, H., Rädle, R., Rolland, C., Gustafson, L. et al., 2024. Sam 2: Segment anything in images and videos. *arXiv preprint arXiv:2408.00714*.

Requicha, A. G., 1980. Representations for rigid solids: Theory, methods, and systems. *ACM Computing Surveys (CSUR)*, 12(4), 437–464.

Shao, J., Yao, W., Wang, P., He, Z., Luo, L., 2024. Urban GeoBIM Construction by Integrating Semantic LiDAR Point Clouds With as-Designed BIM Models. *IEEE Transactions on Geoscience and Remote Sensing*, 62, 1-12.

Tang, W., Li, W., Liang, X., Wysocki, O., Biljecki, F., Holst, C., Jutzi, B., 2025. Texture2lod3: Enabling lod3 building reconstruction with panoramic images. *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2016–2026.

Tochilkin, D., Pankratz, D., Liu, Z., Huang, Z., Letts, A., Li, Y., Liang, D., Laforte, C., Jampani, V., Cao, Y.-P., 2024. Triposr: Fast 3d object reconstruction from a single image. *arXiv preprint arXiv:2403.02151*.

Waechter, M., Moehrle, N., Goesele, M., 2014. Let there be color! large-scale texturing of 3d reconstructions. *European conference on computer vision*, Springer, 836–850.

Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, Z., Liu, W., 2018. Cosface: Large margin cosine loss for deep face recognition. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5265–5274.

Wang, P., Yao, W., Shao, J., He, Z., 2025. Test-time adaptation for geospatial point cloud semantic segmentation with distinct domain shifts. *ISPRS Journal of Photogrammetry and Remote Sensing*, 229, 422-435.

Wysocki, O., Xia, Y., Wysocki, M., Grilli, E., Hoegner, L., Cremers, D., Stilla, U., 2023. Scan2lod3: Reconstructing semantic 3d building models at lod3 using ray casting and bayesian networks. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6548–6558.

Xu, Y., Boerner, R., Yao, W., Hoegner, L., Stilla, U., 2017. AUTOMATED COARSE REGISTRATION OF POINT CLOUDS IN 3D URBAN SCENES USING VOXEL BASED PLANE CONSTRAINT. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2/W4, 185–191.

Xu, Y., Yao, W., Tuttas, S., Hoegner, L., Stilla, U., 2018. Unsupervised Segmentation of Point Clouds From Buildings Using Hierarchical Clustering Based on Gestalt Principles. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 11(11), 4270-4286.

Yarroudh, A., Kharroubi, A., Jeddoub, I., Billen, R., 2024. Automatic upgrade of 3D building models to LoD3 using 3D Point Clouds and Grounding DINO. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 48.