Texture2LoD3: Enabling LoD3 Building Reconstruction With Panoramic Images

Supplementary Material

6. Parameter Settings

Processing Hardware The experiments were conducted on an OMEN HP Laptop 17 with NVIDIA® GeForce RTX™ 4090 Laptop-GPU (16 GB GDDR6), Intel® Core™ i9-Processor 13. Generation, 32 GB DDR5-5200 MHz RAM (2x 16 GB).

B-Rep Preprocessing For facade extraction, a ray-casting approach uses multiple rays per camera view. We integrate camera parameters by setting the camera offset to 0.01 m and assuming a camera height of 1.7 m above the building's lower bound. PCA-based local plane fitting was used for re-triangulation of the fragmented triangular faces.

Geo-Spatial Data Extraction and FOV Computation Building footprints were extracted from CityGML files by parsing the first `posList` element in the `GroundSurface`. Coordinates were converted from EPSG:25832 to EPSG:4326. For field-of-view estimation, horizontal angles were interpolated (10 samples) between the adjusted left and right angles, where the inward offset was set as one-twentieth of the overall FOV (e.g., for a 60° FOV, the offset was 3° for both sides). Five pitch samples were also generated within a $\pm 5°$ range around the optimal pitch computed from wall surfaces.

Panoramic Image Auto-rectification The rectification module uses default configuration parameters from the original method [66]. Each panorama was partitioned into tiles with overlapping regions in our implementation, and a consensus zenith was computed via SVD. The pitch and roll angles for re-projection were derived from the best-fit zenith and further refined by histogram-based aggregation.

Building Facade Segmentation Semantic-SAM was used to generate around 100 to 200 masks on average per street-level image. For semantic filtering, a CLIP confidence threshold of 0.05 was applied. Subsequent morphological processing used a rectangular kernel from size 25×25 to 100×100 to ensure the artifacts on the contour's boundary would not influence the quadrilateral fitting; we also removed connected components smaller than a certain number of pixels, which was set to 2000 on average.

Facade Mask Quadrilateral Fitting After preprocessing the binary masks with a Gaussian blur (kernel size 25×25) and morphological operations, the quadrilateral fitter was applied with the following parameters: Polygons with more than 10 vertices were simplified using an initial epsilon of 0.1, a maximum epsilon of 0.4, and an epsilon increment of 0.02. No additional expansion margin was used. The resulting quadrilaterals were rectified to axis-aligned bounding boxes for perspective transformation.

Texturing by Ray-Casting Rays were cast from the camera using the 10 interpolated horizontal angles and five pitch samples. Intersection points were projected onto the locally fitted facade plane to compute UV texture coordinates. Texture sampling employs bilinear interpolation to ensure a smooth mapping onto the simplified mesh.

Facade Elements Semantic Segmentation Parameters We utilized the Mask2Former model with a Swin-Large backbone, initializing from weights pre-trained on ADE20K. We implemented training procedures for both models with consistent hyperparameters: Batch size of four, AdamW optimizer with a learning rate of 5e-5, and weight decay of 1e-4. Models were trained for 20 epochs with early stopping based on validation loss. Data augmentation included random horizontal flipping and brightness/contrast adjustments to improve generalization. Evaluation metrics included mean Intersection over Union (mIoU) and per-class IoU. Visualization of segmentation results alongside ground truth masks provides qualitative insight into model performance, particularly for challenging cases such as closely spaced windows or irregular architectural elements. Our experimental setup ensured fair comparison across all models by maintaining consistent image resolution, data splits, and evaluation protocols.

7. Further Details on the Selected Baseline Semantic Segmentation Methods

We evaluated the performance of four state-of-the-art semantic segmentation approaches on the task of facade opening detection: SegFormer [63], MaskFormer [8], Mask2Former [9], and Grounded SAM2 [43] (Segment Anything Model with semantic capabilities). Each model represents a different architectural paradigm in the evolution of transformer-based segmentation methods. For the close-set supervised methods, SegFormer [63] combines the hierarchical structure of CNNs with the global modeling capabilities of transformers, utilizing a hierarchical transformer encoder and a lightweight MLP decoder. MaskFormer [8] approaches semantic segmentation as a mask classification problem rather than per-pixel classification. It generates a set of binary masks with associated class predictions, combining the strengths of both semantic and instance segmentation paradigms. Mask2Former [9] advances instance and semantic segmentation through its masked attention mechanism and transformer decoder architecture. For the supervised methods, we leveraged the pre-trained on ADE20K [65], fine-tuned on the CMP dataset [56]. Grounded SAM2 [42] extends the capabilities

of the Segment Anything Model by incorporating semantic grounding, enabling it to perform semantic segmentation with prompt guidance. For our experiments, we used the text prompt *"window"* and *"door"*.

8. Geo-Spatial Data Extraction and FOV Computation

To complement the model preprocessing, we incorporated a geospatial analysis pipeline that served two purposes: (i) extraction of building footprints in a GIS-friendly format and (ii) computation of the camera's field-of-view (FOV) for each building.

GeoJSON Conversion from CityGML.
Building models stored in CityGML files were parsed to extract the `GroundSurface` coordinates. The extracted 3D coordinates (typically in meters) were converted into 2D polygons by retaining the (x,y) components. A coordinate transformation (e.g., from EPSG:25832 to EPSG:4326) was then applied to generate GeoJSON-compliant building footprints. This conversion facilitated integration with external GIS tools and provides a reliable spatial reference for subsequent FOV analysis.

9. Generation of Cropped Perspective Images with Building ID Labeling

After determining each panorama's field-of-view (FOV) as described in Sec. 8, we further generate *cropped perspective images* of the building facades and label them with the corresponding building IDs. The overall pipeline is illustrated on the left side of Figure 8, where each cropped perspective image is annotated with an ID matching the building footprint in the CityGML data.

Overview of the Pipeline

1. **Panorama Cropping Based on FOV** For each panorama, the relevant horizontal span is identified by computing the left and right boundaries of the view. The panorama is then cropped accordingly to focus on the portion containing the target building facade.
2. **Building Region Detection** Detect facade bounding boxes within the cropped panorama using Grounding DINO [33], retaining only the highest-confidence box covering the image center.
3. **Perspective Transformation** Using the bounding box coordinates, a perspective transformation is applied to extract and rectify the facade. This step accounts for the camera's heading and pitch, generating a front-to-parallel view of the building surface.
4. **Building ID Labeling** The resulting perspective image is saved with a filename or metadata embedding the *building ID*. This ID is typically derived from the CityGML data or an external GIS database, ensuring each cropped image can be uniquely matched to the corresponding building footprint.

By following this pipeline, we obtain cropped, perspective-corrected facade images automatically labeled with building IDs. These labeled images are then used to transfer IDs to unlabeled rectified image tiles via feature-based matching (right side of Fig. 8). Section 10 provides full details of this ID association process.

10. Building ID Association

As illustrated in Fig. 8, our objective is to automatically associate labeled building images obtained from CityGML data (which contains building *footprints*) with unlabeled rectified image tiles obtained through a generic panorama rectification process. This step enables us to assign building IDs to the previously unlabelled image tiles. The process consists of the following steps:

1. **Data Preparation and Grouping** We begin by extracting unique building IDs from the object detection and CityGML's provided footprints and obtaining labeled building images through projection or rendering processes (left side of Fig. 8). Simultaneously, panorama images are rectified and split into unlabeled tiles that primarily contain building facades and outlines (right side of Fig. 8).
2. **Feature Extraction and Matching** To match images of the same building from different perspectives, we employ the SIFT algorithm for keypoint detection and descriptor extraction [34]. We further utilize BFMatcher, KNN, and Lowe's Ratio Test to perform precise feature matching. A threshold on the number of inlier matches is applied to filter out false correspondences.
3. **Building ID Association** If a labeled image and an unlabeled rectified tile pass the feature matching threshold (e.g., sufficient inlier matches), we associate the building ID from the labeled image with the rectified tile. This process allows automatic annotation of the previously unlabeled tiles.

By following this approach, the building images with known IDs (examples shown on the left in Fig. 8) can be linked with rectified unlabeled facade tiles (examples on the right in Fig. 8), enabling automatic ID assignment. Experimental results demonstrate that this method achieves robust and accurate multi-view building matching.

11. Further Details on the ReLoD3 Texture Dataset Benchmark Creation

Extraction of Ground-Truth Openings. We extracted precise opening masks directly from 3D building models in the CityGML format to establish reliable ground truth for evaluating semantic segmentation models on facade openings. Our approach leveraged the explicit geometry infor-

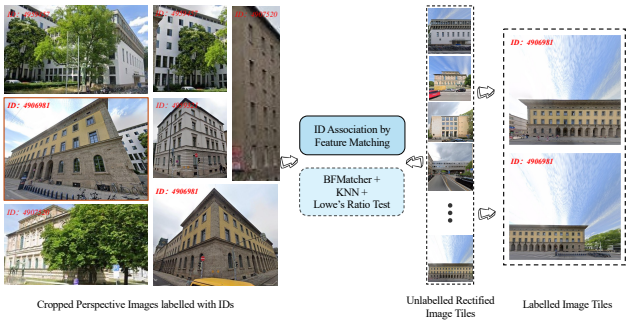


Figure 8. Pipeline of building ID association. The left side illustrates labeled building images obtained from CityGML data, while the right side presents rectified unlabelled facade tiles. The association is performed using feature matching (BFMatcher + KNN + Lowe's Ratio Test) to automatically establish correspondences and assign IDs.

mation available in LoD3 building models, where architectural elements such as doors and windows are explicitly modeled. The extraction process started by identifying wall surfaces (`bldg:WallSurface`) in the CityGML file and their associated opening elements. For each wall, we extracted the 3D coordinates of the facade polygon and all opening polygons. These 3D points were then projected onto a 2D plane using Principal Component Analysis (PCA) to obtain the facade's principal plane. After projection, we converted the 2D points to Shapely [12] polygons for geometric operations. To address potential topology issues in closely positioned openings (e.g., adjacent windows), we implemented a proximity-based grouping algorithm that merged openings within a specified distance threshold (0.1 meters). The facade polygon and opening polygons were combined through boolean operations, where openings were subtracted from the facade to create a comprehensive representation of the wall structure. More details are presented under the project page: *[URL anonymized for the submission]*.

Automatic Download of the Street-View Images. To efficiently acquire street-view images corresponding to building facades, we have designed an automated download process. This process leverages the implementation of [48]. The workflow is as follows:

1. **Sampling Point Generation** Starting from the predefined start and end coordinates, we use linear interpolation to generate multiple sampling points along the line connecting these coordinates. These points cover the area around the building, ensuring that the collected panorama images contain the relevant building facades.
2. **Panorama Query and Download** We query for nearby panorama images for each sampling point. The unique panorama ID is checked against a set of already downloaded IDs to avoid duplicate downloads.
3. **Metadata Recording** During the download process, the script collects metadata for each panorama, including panorama ID, latitude, longitude, heading (in both radians and degrees), capture date, and location; Then, it stores it in a CSV file. This metadata facilitates later association with the CityGML data and further analysis.

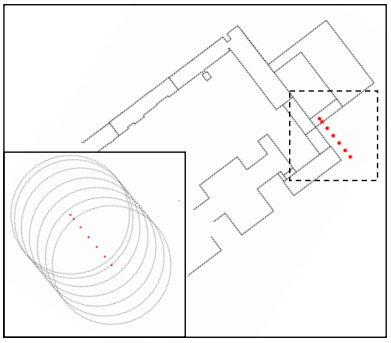


Figure 9. Schematic illustration of building footprint (black), sampling points (red), and the buffer area (gray dashed circles). The buffer defines a maximum distance from each sampling point within which building facades can be captured or considered visible. This ensures coverage of the building's facade from multiple vantage points and avoids unnecessary distant panoramas.

As illustrated in Fig. 9, the *buffer* is a circular region around each sampling point (with a user-defined radius, e.g., 50 meters). Only those building surfaces (or facade elements) intersecting this buffer are considered relevant for capturing street-view panoramas. This automated workflow ensures high spatial consistency between the street-view images and the building data while significantly improving the efficiency of data collection, thereby providing a robust foundation for subsequent facade texturing and analysis tasks.

Manual 4-point Projection of Perspective Images The manually projected perspective terrestrial optical images of the digital camera (Sony $\alpha 7$) were acquired specifically for validating automatic texturing purposes. The campaign was designed to capture the building model facades with a minimum number of photographs per triangle in the existing LoD2 building models to ensure texture consistency without any additional image stitching.

The *4-point projection* refers to the texturing implementation of the proprietary SketchUp Pro [53] software with the CityEditor [2] plugin. While the default SketchUp Pro allows for the manual identification of four image-to-model projection points, the CityEditor allows the loading of CityGML building models into the SketchUp software. Additionally, LoD3 ground-truth models were loaded to guide the manual projection process. Nevertheless, owing to still persistent distortions, the deviations between the ground-truth LoD3 and manual projection exist. As such,

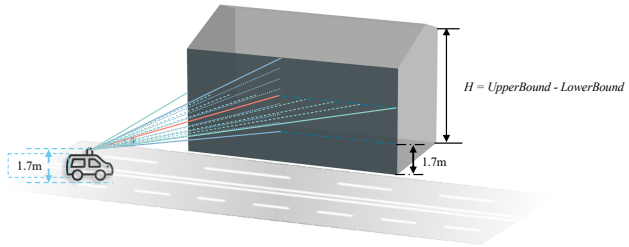


Figure 10. An illustration of our raycasting-based texturing setup. The camera (e.g., mounted on a vehicle at 1.7 m height) casts multiple rays toward the building's facade, which extends from the lower bound to the upper bound obtained from the GML data. We sample horizontal angles between the left and right viewing directions and interpolate a small range of pitch angles to capture the relevant parts of the facade.

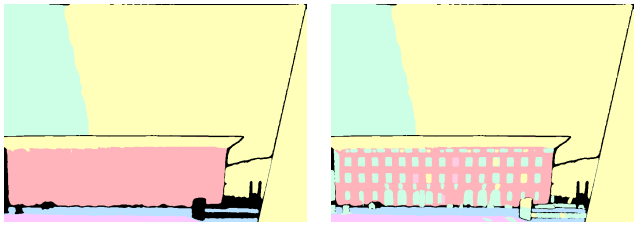
the distortion-free and cm-accurate LoD3 masks shall be treated as the ground truth.

12. Texturing after triangulation

We first employ our wireframe preprocessing pipeline (Sec. 3.1) to enable robust texturing of building facades to convert highly subdivided B-Reps into minimal quadrilateral faces. After this simplification step, we perform ray casting from known camera poses to identify which faces are visible from each viewpoint. Figure 10 illustrates how the camera, positioned at 1.7 m above the ground, casts rays spanning a specified field of view. The building facade's lower and upper vertical bounds are derived from CityGML data, ensuring that our texturing pipeline only samples the relevant portions of the geometry. For each B-Rep:

1. We compute the camera origin and direction based on geographic coordinates and a small offset from the facade.
2. We cast multiple rays spanning the horizontal viewing angles (from left to right and a range of pitch angles around the facade's center.
3. We collect all intersected faces and compute appropriate UV coordinates for texturing. Faces whose normals point inwards are automatically flipped to ensure the texture is placed on the exterior surface.

Finally, once all relevant faces are identified, we project the corresponding panoramic images onto these faces using a planar mapping approach (Eq. (13)). This step ensures that the final textured facade remains visually coherent and avoids the distortions that can arise when projecting onto densely triangulated B-Reps. The resulting textured model forms the basis for subsequent facade analysis and segmentation (Sec. 3.3).



(a) Coarse segmentation (10 masks) (b) Fine-grained segmentation (127 masks)

Figure 11. Comparison of segmentation results using different numbers of retained candidate masks. A small number of masks (left) leads to fewer, larger segments capturing the main facade region. In contrast, a larger number of masks (right) produces more detailed but also more fragmented subregions.

13. Building Facade Segmentation: Influence of Candidate Masks

This step aims to detect and isolate the main building facade from the textured geometry. Our approach employs a semantic segmentation pipeline built upon Semantic-SAM, which automatically generates a set of candidate masks for each panoramic or perspective image. We then filter these masks to retain only those corresponding to the "building facade" class, discarding irrelevant classes such as sky, road, or cars. Small floating artifacts are removed via connected-component analysis, and we apply morphological smoothing to obtain a clean, consolidated facade mask suitable for further processing.

Figure 11 demonstrates how adjusting the quantity of retained candidate masks affects the final segmentation. In Fig. 11(a), retaining only 10 masks results in coarser segmentation with fewer, larger regions that effectively capture the overall facade shape. Such coarse segmentation is often advantageous when the primary goal is to isolate the facade with minimal clutter. Conversely, Fig. 11(b) shows a more fine-grained segmentation derived from 127 candidate masks, revealing additional details such as windows or ornamental features. While this can benefit downstream tasks requiring higher granularity, it also increases the likelihood of fragmented subregions that complicate facade isolation.

14. Test-time Alignment for Mask Evaluation

Due to the inherent transformation challenges in panorama rectification, we implement a test-time scale and shift adjustment procedure when evaluating predicted segmentation masks against ground truth masks. This adjustment is necessary because the rectification process introduces unavoidable geometric distortions, causing the segmented objects to lose their absolute scale and position relative to the original panoramic view. Our method employs a two-stage optimization approach: First, conducting a coarse grid

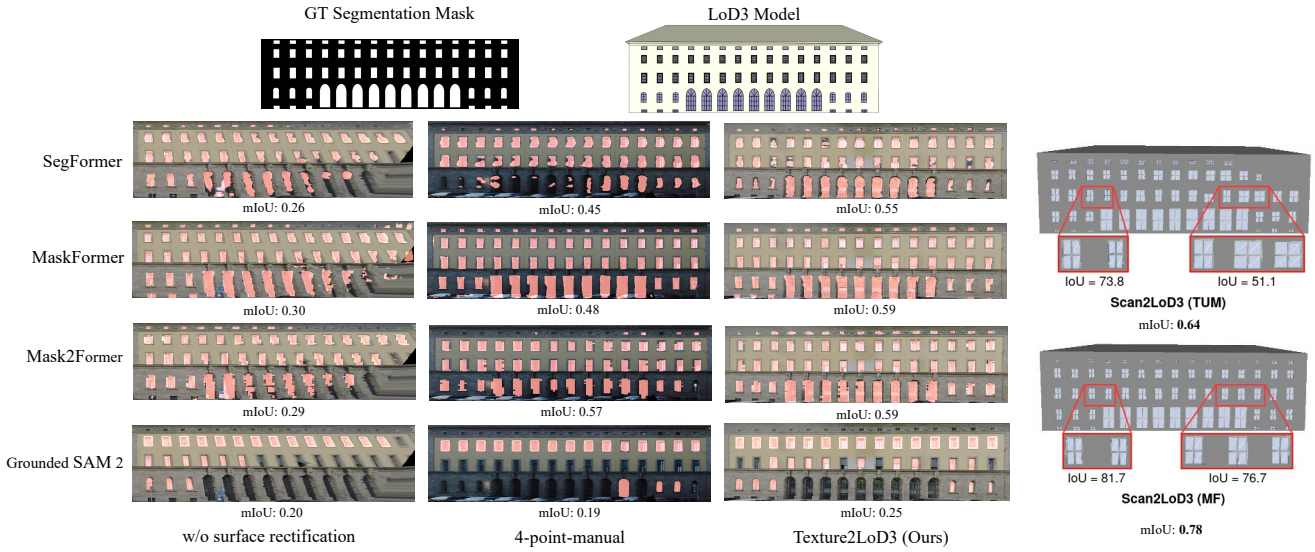


Figure 12. Comparison of the baselines and the Texture2LoD3 method to the Scan2LoD3 method leveraging multi-modal fusion of laser scanning, 3D model priors, and street-level images. Such an approach clearly outperforms only image and model combinations. Yet such a multi-modal setup is scarcely available in practical scenarios, unlike street-level images and 3D models. Figure parts copied and edited from the original Scan2LoD3 article, where experiments were conducted on the same object, courtesy of Wysocki et al. [59].

search over a constrained parameter space (scale factors between 0.75 and 1.2, and pixel translations within ± 100 pixels), followed by a finer search within a more focused range around the best parameters identified in the first stage. For each candidate transformation, we compute the Intersection over Union (IoU) between the predicted mask and the transformed ground truth mask, selecting the parameters that maximize this metric. This alignment procedure ensures a fair comparison between prediction and ground truth by compensating for the scale and positional discrepancies introduced during the rectification process without altering the structural integrity of the segmentation boundaries.

15. Comparison to the Scan2LoD3 method

As mentioned in Related Work (Section 2), there are methods leveraging the accuracy of laser scanning, building priors, and images to reconstruct LoD3 building models. We acknowledge that this approach yields superior performance to our work owing to the use of accurate laser scanning modality and physics-oriented ray analysis. Due to that fact, this comparison is out of the scope of the main publication part. Nevertheless, such a comparison is worth showcasing modalities' limitations, primarily since experiments were performed partially on the same objects. Here, we selected an excerpt from the Wysocki et al. [59] Scan2LoD3 method that performed the analysis on the same building (the so-called *building 23*). As we show in Figure 12, the performance on the same facade increases significantly owing to the laser scanner modality. It scored 78% while using high accuracy scanner, and 64% when using lower grade Velodyne scanner. This experiment shows a minimum of 5% and a maximum of 14% increase compared to the best baseline image-based segmentation. Yet, as we elaborate in Related Work (Section 2), such a multi-modal setup is still scarcely available, in contrast to the ubiquitous street-level images and 3D prior models.