Texture2LoD3: Enabling LoD3 Building Reconstruction With Panoramic Images

Wenzhao Tang[*1] , Weihang Li[*1,2] , Xiucheng Liang[3] , Olaf Wysocki[1] ,
Filip Biljecki[3] , Christoph Holst[1] , Boris Jutzi[1]

[1] Technical University of Munich, [2] Munich Center for Machine Learning,
[3] National University of Singapore

(wenzhao.tang, ... boris.jutzi)@tum.de ; (xiucheng, filip)@nus.edu.sg

* equal contribution

Abstract

Despite recent advancements in surface reconstruction, Level of Detail (LoD) 3 building reconstruction remains an unresolved challenge. The main issue pertains to the object-oriented modelling paradigm, which requires georeferencing, watertight geometry, facade semantics, and low-poly representation – Contrasting unstructured mesh-oriented models. In Texture2LoD3, we introduce a novel method leveraging the ubiquity of 3D building model priors and panoramic street-level images, enabling the reconstruction of LoD3 building models. We observe that prior low-detail building models can serve as valid planar targets for ortho-rectifying street-level panoramic images. Moreover, deploying segmentation on accurately textured low-level building surfaces supports maintaining essential georeferencing, watertight geometry, and low-poly representation for LoD3 reconstruction. In the absence of LoD3 validation data, we additionally introduce the ReLoD3 dataset, on which we experimentally demonstrate that our method leads to improved facade segmentation accuracy by 11% and can replace costly manual projections. We believe that Texture2LoD3 can scale the adoption of LoD3 models, opening applications in estimating building solar potential or enhancing autonomous driving simulations. The project website, code, and data are available here: https://wenzhaotang.github.io/Texture2LoD3/.

1. Introduction

Photogrammetry and computer vision researchers have always seen detailed semantic 3D building reconstruction as a fundamental challenge [17, 50]. Recent developments in open source and proprietary software have shown that reconstruction using 2D building footprints and aerial observations enables country-wide reconstruction up to the

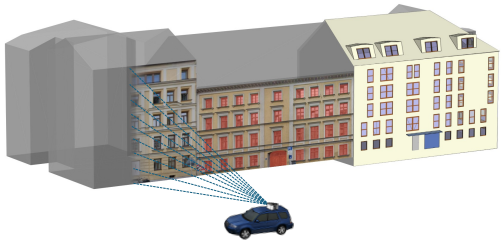


Figure 1. Texture2LoD3 proposes leveraging ubiquitous street-level images and low-level building models for accurate ortho-texturing (left): Enabling accurate semantic segmentation (center) and facade-rich level of detail (LoD)3 reconstruction (right).

LoD2 displaying complex roof shapes and simplified facades [17, 46, 60]. Unlike the mesh-oriented models, the semantic 3D building models defined by the international CityGML standard [15] are georeferenced, watertight, and have low-poly boundary representation (B-Rep), enabling multiple applications [4]. Remarkably, such models remain under-explored modality for methods development, given their ubiquity, e.g., open data on 215 million buildings in Switzerland, the Netherlands, the US, or Poland [60, 61].

Unlike low-detail LoD1 and LoD2, LoD3 models are characterized by additional detailed facade representation and remain scarcely available despite novel methods presence [21, 24, 39, 57, 59]. One of the main issues pertains to the source data availability, assuming either accurate mobile laser scanning (MLS) observations or ortho-rectified textures, which in practice are often unavailable.

Despite worldwide availability of panoramic street-level images such as Google Street View (GSV) or Mapillary [23] and the growth in image-based training datasets, facade elements remain frequently unlabeled and limited to ortho-rectified image views [29, 56, 57, 59]. Applying such training sets to perspective and panoramic images remains unfeasible due to drastic geometry representation changes

in facade elements, e.g., the closer rectangular windows are to the vanishing point, the more they resemble lines.

As we exemplify in Fig. 1, our Texture2LoD3 proposes a method harnessing the potential of widely available panoramic street-view images and ubiquitous low-level semantic 3D building models. We leverage the georeferencing of two modalities for their global matching while low-poly planar representation of 3D models for the image ortho-rectification target. By utilizing prior low-poly models, we satisfy requirements of georeferencing, watertightness, low-poly representation, and geometrical consistency for LoD3 reconstruction: Formulating it as a refinement strategy [59] of low-level models to high-detail LoD3 models by reconstructing only the required facade elements, segmented from a projected image onto a planar surface. Our main contributions are as follows:

- We propose the effective projection of panoramic images to ortho-rectified images by leveraging ubiquitous semantic 3D building models as targets
- We improve facade semantic segmentation performance on 3D surfaces by accurate texturing: Enabling accurate LoD3 facade element reconstruction
- We introduce the first-of-its-kind open texturing benchmark dataset, ReLoD3, comprising synchronised LoD3 models, panoramic images, and manually textured low-level LoD2 building models

2. Related Works

3D Facade Segmentation The recent years have witnessed a surge in semantic 3D facade segmentation methods both on point clouds, images, and in combination with prior 3D models. Since the current research suggests that street-level and drone-based point clouds accurately depict 3D facade geometry, multiple point-cloud-based methods have been proposed [14, 35, 40, 52]. Recent benchmark data results, such as ZAHA [62] and ArCH [35], imply that the challenge is still unsolved and remains challenging due to under-represented classes, sparsity of objects in point clouds, and frequently indistinct 3D geometry features [45, 49].

Other approaches rely only on image-based input, capitalizing on rich optical features and 2D image grid representation. Various methods have been proposed to tackle this challenge, such as non-learning [37, 50], gramma-based [5, 36], and recently deep learning approaches [7, 20, 24, 32, 57]. Owing to the ubiquity of image training data, even the standard Mask-RCNN [19] proves relatively efficient after the subsequent fine-tunning on the facade image databases [59]. However, these methods perform well only under the assumption that an image is ortho-rectified; it makes generalization challenging since facade elements are prone to the dire geometry change under perspective and barrel distortions. This applies to classical methods as well which explicitly concentrate on line and point extraction for matching images with models for texturing [22, 25, 26, 51]. In practice, ortho-rectified images are rare and limited just to a few benchmarks or manual projections, yielding unsatisfactory results on non-rectified real-world data [6, 11, 27, 29, 44, 56].

An alternative approach is to exploit information from 3D models, optical images, and laser scanning point clouds to achieve accurate 3D facade segmentation [55, 59]. For example, Scan2LoD3 [59] introduces a method where uncertainty-aware ray analysis of laser points with 3D models yield conflict maps indicating openings, which can serve as evidence for late-fusion of 3D segmented point clouds and 2D segmented optical images. However, the availability of such multi-modal setups is currently limited and assumes their heterogeneous accurate projection onto the model surface.

LoD3 Building Reconstruction Semantic 3D building reconstruction is a long-standing challenge in photogrammetry and computer vision [50]. For years, the international standard CityGML [3, 16] has been defining the formal description of such models, where LoD1 stands for simple cuboid models, LoD2 for polyhedral models with detailed roof shape, and LoD3 for detailed roof shapes complemented with a detailed facade representation. The primary difference to the standard mesh models is that semantic 3D building models are georeferenced; comprise object-level geometry and semantics; have a hierarchical data model that also describes the object-to-object relationship; display watertight and low-poly geometry facilitating volumetric space interpretation by integrating externally observable surfaces within a boundary representation (B-Rep) [16, 28, 60].

Despite recent advancements in LoD3 building reconstruction, LoD3 models remain scarce [18, 20, 38, 39, 47, 57, 59]. One of the main remaining issues is the robustness of methods when deployed at scale. Most of the methods assume that a specialized method of acquisition is required: It sets a high requirement for the practical methods' deployment, as these methods assume targeted accurate co-registration of multiple subsequent images and complete object coverage without adjacent buildings, e.g., single house acquired by a 360-degree drone flight [24, 39]. Alternatively, the above-mentioned Scan2LoD3 [59] can mitigate such issues by introducing additional conflict maps of the ray-to-prior-model analysis.

3. Method

As shown in Fig. 2, our Texture2LoD3 method commences with the image-to-object matching of widely-available georeferenced panoramic images and ubiquitous low-level semantic 3D building models (Sec. 3.1). This process is followed by 3D model B-Rep surface simplification (top-branch), while panoramic images are rectified (Sec. 3.2) and building facades are segmented (bottom-branch) (Sec. 3.3).

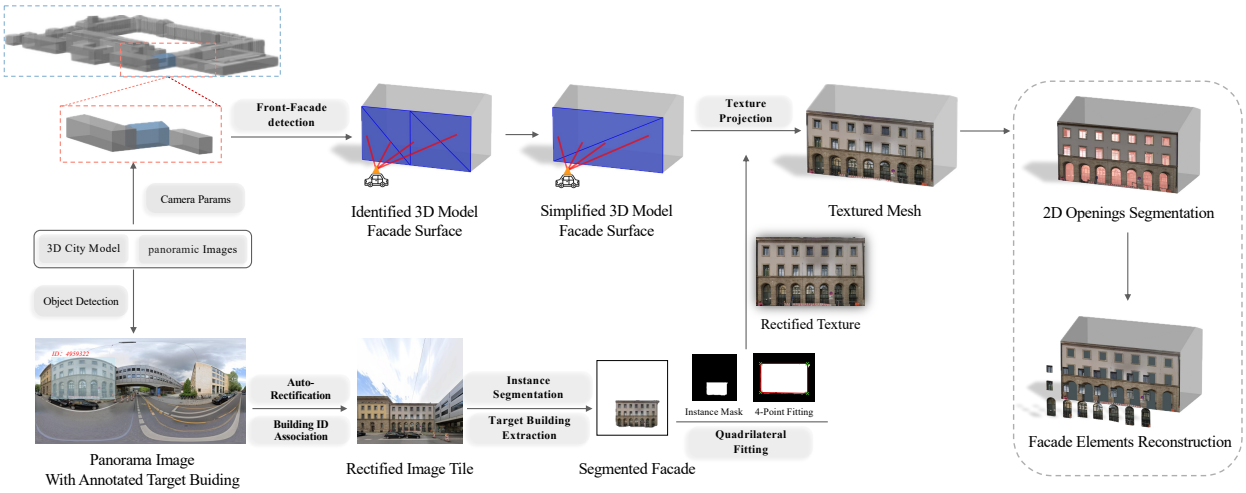


Figure 2. Overview of the proposed Texture2LoD3 method: The method commences with global matching of georeferenced panorama images and low-level 3D models. In the top branch, 3D target facade surfaces are simplified, while in the bottom branch panoramic images are rectified and building facade instances are extracted. Subsequently, fine object-to-object matching and projection is performed to the simplified 3D model surface. Quadrilateral fitting and image-to-plane ray casting ensure accurate ortho-rectified 3D texture, enabling accurate facade elements segmentation and LoD3 reconstruction.

The fine quadrilateral fitting of the facade instance shall ensure complete facade depiction (Sec. 3.4), followed by ray-casting-based projection onto the simplified 3D model planar surface (Sec. 3.5); Enabling accurate facade elements segmentation and LoD3 reconstruction.

3.1. Matching Panoramic Image to 3D Model

In this work, "matching" refers to aligning geo-referenced ground-level panoramic images with corresponding 3D building models. Specifically, the goal is to associate the facade observed in a ground-level image with its counterpart in the 3D model, thus establishing a coherent mapping between image pixels and 3D geometry.

Camera Parameters We assume that each panoramic image is accompanied by a set of camera parameters that are essential for the matching process. In particular, the camera parameters include: a) *Position*: The geographic coordinates (latitude and longitude) of the camera; b) *Heading*: The azimuth angle indicating the direction the camera faces, measured in degrees clockwise from North; c) *Field-of-view (FOV)*: The angular extent of the scene captured by the camera in degrees; d) *Generic parameters*: Any extra available parameters, e.g., the camera's height above ground level.

Due to the imprecision of geo-referenced data, the available 2D sensor positions and 3D model vertices in the B-Rep only provide a coarse association. Moreover, semantic 3D building models often subdivide a single facade into multiple small triangular faces—a phenomenon we refer to as facade subdivision. This subdivision complicates texture mapping because it prevents a straightforward correspondence between image features and continuous facade regions. To overcome these issues, we propose a unified ray-casting-based approach that leverages camera parameters to detect facade regions and simplify the 3D model, thereby facilitating a robust matching between the panoramic image and the 3D building model.

3D B-Rep Model and Camera Integration We first extract the camera parameters (position, heading, FOV, and the manually set camera height) from the geo-referenced panoramic images and project them into the global building coordinate reference system. In our framework, the 3D building model is represented as a boundary representation (B-Rep), i.e., a collection of vertices, edges, and faces that define the surfaces of the building; We assume the following information is available: a) *Ground surface definition*: The model explicitly delineates the building's base, from which the building height can be extracted, e.g., via the minimum and maximum Z-coordinates adhering to the CityGML GroundSurface definition [16]; b) *Altitude and orientation*: The model's global orientation (altitude) is inherently defined within a global coordinate reference system, ensuring that facade orientations are consistent; c) *Height*: The vertical extent of the building is provided or can be computed from the B-Rep, enabling precise placement of the camera.

Ray-Casting-Based Facade Detection For each camera, multiple rays with varying horizontal and vertical angles are cast against the 3D model's triangular mesh. The ray-casting process records the intersected faces and their spatial distribution. We then select the camera view that yields

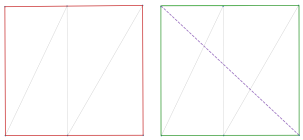


Figure 3. (Left) Original surface with multiple triangular faces. (Right) Fitted quadrilateral representation with re-triangulation along the diagonal (dashed purple), preserving facade shape.

the highest number of valid intersections and best aligns the camera position with the centroid of the hit points. Note that here, ray-casting is used to robustly detect the facade region by identifying the contiguous set of faces corresponding to the building's facade, even in the presence of fragmentation. This detection step is crucial for the subsequent matching process, as it determines which part of the 3D model corresponds to the observed image.

Local Plane Fitting and 3D B-Rep Model Simplification The set of intersected triangular faces from the optimal view is aggregated and fitted to a local plane via principal component analysis (PCA), which yields a centroid $\mathbf{c}$ and two in-plane basis vectors $\mathbf{u}$ and $\mathbf{v}$. Each vertex p on the detected facade is then projected onto this plane:

$$x = \langle p - \mathbf{c}, \mathbf{u} \rangle, \quad y = \langle p - \mathbf{c}, \mathbf{v} \rangle \tag{1}$$

From the 2D projections, a minimum area bounding rectangle is computed, resulting in four corner points $\{(x_i, y_i)\}_{i=1}^{4}$. These corners are mapped back into 3D space:

$$\mathbf{q}_i = \mathbf{c} + x_i\,\mathbf{u} + y_i\,\mathbf{v}, \quad i = 1, \ldots, 4 \tag{2}$$

The quadrilateral defined by $\{\mathbf{q}_i\}$ is subsequently re-triangulated into two triangles, thereby replacing the fragmented original representation with a simplified mesh that preserves critical geometric features while reducing computational complexity (Fig. 3).

It is worth noting that the literature offers a wide variety of methods for geometric simplification and for converting between triangular and quadrilateral representations, such as those available in the CGAL library [10]. In contrast, our approach relies solely on the consistency of plane normals computed via PCA, which is robust under the assumption that the facade region is locally planar—a reasonable assumption for most urban building facades and semantic 3D city models that shall adhere to this assumption.

FOV Calculation Based on Building Geometry To compute the effective camera field-of-view (FOV) for each building, we define a buffer region around each camera observation point to identify nearby structures. The exterior boundaries of the building are sampled to determine the angular directions (bearings) from the camera. By evaluating occlusion effects—ensuring that each building vertex is visible without interference from adjacent structures—we determine the effective angular extent of the building facade. These FOV metrics aim to exclude occlusion-induced noise and the target building's facade view.

3.2. Panoramic Image Auto-rectification

We utilize an automatic rectification approach for panoramic images consisting of three stages inspired by Zhu et al. [67]: a) tile extraction and local rectification; b) consensus estimation of zenith and horizontal vanishing points; and c) global re-projection. This part of the method aims to effectively rectify panoramic images by combining local tile analysis, a robust SVD-based consensus, and global re-projection. It shall provide a consistent geometric basis for subsequent facade segmentation and texturing.

Tile Extraction and Local Rectification We partition the input panorama image into multiple overlapping tiles via a ray-casting strategy. Local features and edges within each tile yield estimates of the horizon line $\mathbf{h}$, horizontal vanishing points $\{\mathbf{v}_i\}$, and a local zenith vector $\mathbf{z}$. Importantly, the local zenith vector $\mathbf{z}$ is computed independently from the horizontal vanishing points. Specifically, while both are derived from the same set of local edge features, the zenith vector is estimated via a robust SVD-based process on the normalized edge directions, which directly captures the predominant vertical direction in each tile. These local parameters serve as geometric cues for subsequent global alignment. Although the image is already rectified, semantic information does not drive the rectification process. Consequently, when a building's facade is particularly wide, individual tiles may only capture a portion of the facade (even if that portion is rectified). In such cases, subsequent image tile stitching (Section 3.2) is necessary to produce a more complete representation of the facade.

Consensus Estimation We aggregate all normalized zenith vectors $\{\mathbf{z}_i\}$ and compute a consensus zenith $\mathbf{z}^*$ via SVD:

$$\mathbf{z}^* = \mathrm{SVD}(\{\mathbf{z}_i\}) \tag{3}$$

From $\mathbf{z}^* = (z_x, z_y, z_z)^\top$, the pitch ϕ and roll θ angles are:

$$\phi = \arctan\Big(\frac{z_z}{z_y}\Big), \quad \theta = -\arctan\Big(\frac{z_x}{\mathrm{sgn}(z_y)\sqrt{z_y^2 + z_z^2}}\Big) \tag{4}$$

We define standard rotation matrices $R_{\text{roll}}(\theta)$, $R_{\text{pitch}}(\phi)$ (and optionally $R_{\text{heading}}(\psi)$) to align the vanishing points. A histogram of horizontal angles can further refine these estimates if necessary.

Global Re-projection With the consensus rotation determined, we re-project the entire panorama image into a rectified view. For a pixel with spherical coordinates (θ, ϕ), its 3D direction vector $\mathbf{v}(\theta, \phi)$ is rotated back by $R_{\text{roll}}(-\theta)$ and

$R_{\text{pitch}}(-\phi)$. The result is then mapped to image coordinates via an inverse equirectangular projection:

$$x = \left(\frac{\theta'}{360^\circ} + \frac{1}{2}\right)W, \quad y = \left(\frac{\phi'}{180^\circ} + \frac{1}{2}\right)H \tag{5}$$

Image Tile Stitching In cases where a single rectified tile cannot capture the entire building facade, we stitch multiple overlapping tiles into one image. We detect SIFT keypoints in each tile, match them across overlaps, and estimate a robust homography via RANSAC [31]. The source tile is then warped accordingly, and a smooth blending operation mitigates seam artifacts.

3.3. Building Facade Segmentation

To accurately isolate and extract building facades from complex urban scenes, we adopt the pipeline illustrated in Fig. 4. Our approach integrates an automatic instance-level segmentation (Semantic-SAM [30]) with semantic filtering via CLIP [41], thus allowing building facades to be selectively retained while discarding irrelevant objects (e.g., cars, trees, people). We choose Semantic-SAM owing to its outstanding performance in instance segmentation tasks [30]. Given that many of our input images feature multiple adjacent building facades, Semantic-SAM's robust segmentation capability is essential for reliably distinguishing individual facade instances.

Instance Generation via Semantic-SAM Given a rectified panoramic image I, we employ the Semantic-SAM automatic mask generator to produce a set of unlabeled instance masks $\{M_i\}$. These masks aim to cover all salient regions in the scene, ranging from building surfaces to smaller objects like cars or trees. Although the mask generator provides instance-level segmentation, no semantic labels are assigned.

CLIP-Based Label Filtering To determine which instance masks correspond to building facades, we process each masked image region using a CLIP [41] encoder (ViT-L/14). Specifically, we compute an image embedding and compare it via cosine similarity to text embeddings derived from a predefined set of text prompts (typically 2–3 prompts, e.g., *"building facade"*, *"vehicle"*, and *"pedestrian on the street"*). An instance is retained if its highest-confidence label is *"building facade"* and its similarity score exceeds a chosen threshold; otherwise, it is discarded. Additionally, masks identified as *"building eave"* are subtracted to ensure that only the primary vertical surfaces of the building remain. This process also filters out instances classified as *"vehicle"* or *"pedestrian on the street"* to exclude dynamic and non-architectural elements from further processing.

Mask Combination and Noise Removal As multiple facade masks may be produced for a single building or portions thereof, we unify them via logical OR: $M_{\text{facade}} = \bigvee_{i \in \mathcal{I}} M_i$, where each M_i is "building facade". Likewise, all eave masks are aggregated via logical OR and then subtracted from M_{facade}. We further remove small connected components whose area is below a minimum threshold, estimated by $A_{\min}$, to eliminate spurious detections. Morphological opening and closing are then performed using a kernel of size $k \times k$ (with k chosen according to the image resolution) to fill small holes and smooth the boundaries of the combined mask.

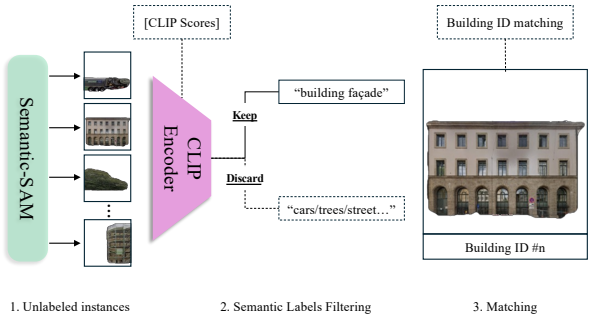


Figure 4. Semantic-SAM generates unlabeled instance masks, which are then passed to a CLIP encoder for semantic filtering. We retain masks classified as *building facade*.

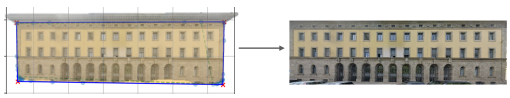


Figure 5. Building facade after filtering out extraneous parts, e.g., eave masks, and a schematic view quadrilateral fitting extracting the four corner points based on the refined mask.

Final Facade Extraction After noise removal, the resulting binary mask accurately outlines the dominant building facades. As a final step, we align the mask size with the original panorama image and multiply it element-wise with the original image $I_{\text{masked}}(x, y) = I(x, y) \times M_{\text{facade}}(x, y)$, yielding a facade-only color image that is preserved for subsequent morphological adaptation (Sec. 3.4) and texturing (Sec. 3.5).

3.4. Facade Mask Quadrilateral Fitting

In this step, we refine the facade segmentation mask to produce a clean, noise-free representation that accurately outlines the facade (Fig. 5). The process consists of three stages: a) mask smoothing via morphological operations; b) robust quadrilateral fitting to the facade contour; and c) perspective rectification.

Smoothing via Morphological Operations Given an input binary mask I, we first smooth the mask by applying a Gaussian blur:

$$I_{\text{blur}}(x, y) = \sum_{(u,v) \in \Omega} G(u, v, \sigma)\, I(x - u, y - v), \tag{6}$$

where $G(u, v, \sigma)$ is a Gaussian kernel and Ω is the kernel support. This smoothing reduces high-frequency noise. Next, we perform morphological closing followed by opening to fill small holes and remove spurious regions:

$$I_{\text{close}} = (I_{\text{blur}} \oplus B) \ominus B, \quad I_{\text{open}} = (I_{\text{close}} \ominus B) \oplus B, \quad (7)$$

with $\oplus$ and $\ominus$ denoting dilation and erosion, respectively, and B being a rectangular structuring element of size (15×15). From the resulting mask, contours are extracted and the largest contour, $C_{\max}$, is selected: $C_{\max} = \arg\max_{C \in \mathcal{C}} \text{Area}(C)$. Its convex hull, $H = \text{convexHull}(C_{\max})$, provides a robust boundary for the facade.

Quadrilateral Fitting To obtain a compact facade representation, we fit a quadrilateral to the points of the convex hull. Let $\mathcal{P} = \{p_1, p_2, \ldots, p_n\}$ denote the set of points in H. We seek a quadrilateral Q with vertices $\{q_1, q_2, q_3, q_4\}$ that maximizes the Intersection over Union (IoU) with H, where:

$$\text{IoU}(H, Q) = \frac{\text{Area}(H \cap Q)}{\text{Area}(H \cup Q)} \quad (8)$$

Perspective Rectification With the scaled quadrilateral Q^{scaled}, we compute a homography that maps its vertices to the corners of a target rectangle. Assuming that the target image has width W and height H, we define:

$$T = \{(0,0),\ (W-1,0),\ (W-1,H-1),\ (0,H-1)\} \quad (9)$$

The homography matrix P satisfies:

$$\begin{bmatrix} x_i' \\ y_i' \\ 1 \end{bmatrix} \sim P \begin{bmatrix} x_i \\ y_i \\ 1 \end{bmatrix}, \quad i = 1, \ldots, 4 \quad (10)$$

where (x_i, y_i) are the coordinates of q_i^{scaled} and (x_i', y_i') are the corresponding target coordinates. This perspective transformation matrix is computed and applied to the original image: $I_{\text{warped}} = \text{warpPerspective}(I_{\text{orig}}, P)$.

3.5. Facade Texturing by Ray-Casting

In this stage, we accurately map the texture from the panoramic image onto the simplified facade geometry. Our approach uses a ray-casting method that projects rays from the camera center and computes their intersections with the facade surface, thus determining the texture coordinates for each sample.

Ray Generation and Direction Determination Using the simplified facade (Sec. 3.1), we generate a 3D ray for each sampling point on the target texture grid. Each pixel in the panoramic image is first associated with spherical coordinates (θ, ϕ), from which its 3D direction vector is computed as:

$$\mathbf{v}(\theta, \phi) = \begin{bmatrix} \cos\phi \sin\theta \\ \sin\phi \\ \cos\phi \cos\theta \end{bmatrix} \quad (11)$$

Subsequently, the direction is adjusted using the inverse of the rotation matrices derived during the panoramic image auto-rectification stage for pitch, roll, and heading (Sec. 3.2). This step aligns the rays with the actual orientation of the facade.

Ray-Facade Intersection Each ray, cast from the camera center $\mathbf{o}$, is tested for intersection with the facade surface. Since the facade is approximated as a quadrilateral (typically decomposed into two triangles), the intersection point is calculated using the standard ray-plane intersection formula:

$$t = \frac{(\mathbf{p}_0 - \mathbf{o}) \cdot \mathbf{n}}{\mathbf{v} \cdot \mathbf{n}} \quad (12)$$

where $\mathbf{p}_0$ is an arbitrary point on the facade plane, and $\mathbf{n}$ is the unit normal vector of the plane. The intersection point is then given by $\mathbf{p} = \mathbf{o} + t\,\mathbf{v}$.

While a homography warp from rectified images could be used for texture mapping, it assumes that the facade is perfectly planar and that the rectification is flawless. In practice, residual geometric distortions and local deviations from planarity often persist. Our ray-casting method directly computes the intersection of rays with the actual 3D facade, thereby accommodating these imperfections and ensuring a more robust and accurate texture mapping. Moreover, a simple homography warp cannot account for non-planarities or slight misalignments due to calibration errors, which our ray-casting approach inherently corrects by leveraging the true 3D geometry.

Texture Coordinate Mapping Once the intersection point $\mathbf{p}$ is determined, it is projected onto the local 2D coordinate system of the facade using the plane parameters obtained from PCA (centroid $\mathbf{c}$ and in-plane basis vectors $\mathbf{u}$ and $\mathbf{v}$):

$$x = \langle \mathbf{p} - \mathbf{c}, \mathbf{u} \rangle, \quad y = \langle \mathbf{p} - \mathbf{c}, \mathbf{v} \rangle \quad (13)$$

After normalization, the (x, y) coordinates correspond directly to the texture coordinates in the original panoramic image.

Texture Sampling and Synthesis The texture coordinates are used to sample pixel values from the panoramic image, employing bilinear interpolation to ensure pixel reprojection. These sampled values are then mapped onto the simplified facade mesh, thereby generating a high-detail, geometrically consistent texture.

4. Experiments

Our ReLoD3 Texture Dataset Benchmark In the absence of datasets comprising accurate LoD3 reference data aligned with extracted opening masks, manual textures, and street-level images, we introduce the ReLoD3 dataset. The ReLoD3 comprises 27 unique LoD3 models modeled according to the CityGML standard [16] including windows, doors, and eaves modeled based on high-accuracy MLS

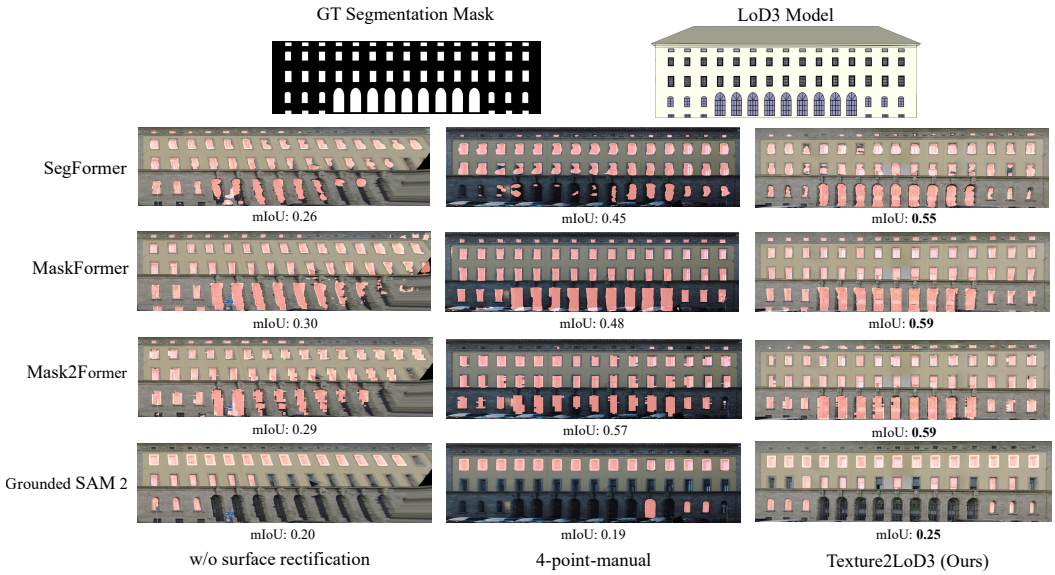


Figure 6. Tested facade segmentation baselines on a selected building from the introduced ReLoD3 benchmark dataset across various texture projection methods. Our Texture2LoD3 is less prone to distortions, hence yielding more accurate segmentation across the baselines.

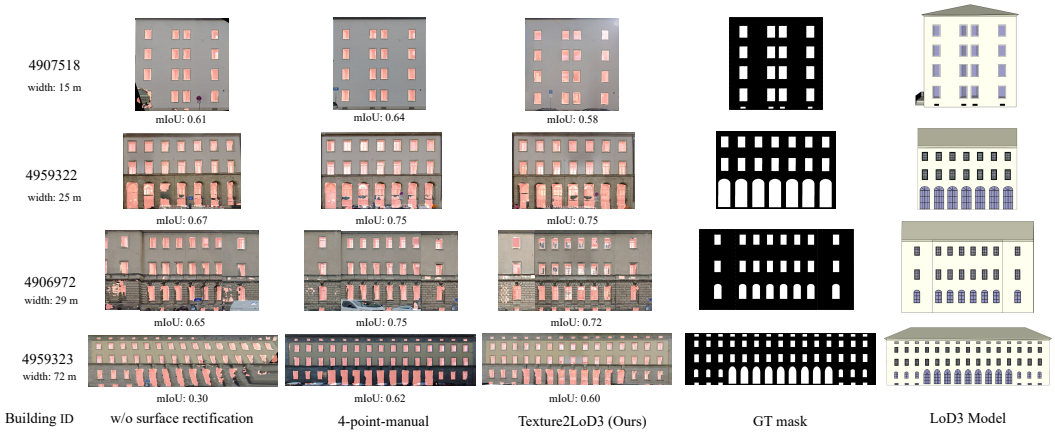


Figure 7. Texture2LoD3 maintains on-par accuracy with manual texturing even in the presence of increasing facade width, unlike the method without rectification. Shown on MaskFormer [8] on four width-different facades of the introduced ReLoD3 benchmark dataset.

Table 1. Quantitative comparison of semantic segmentation models on facade opening detection across two datasets. Performance is measured using SSIM (↑), IoU (↑), and LPIPS (↓).

	w/o surface rectification			4-point-manual			Texture2LoD3 (Ours)		
Methods	SSIM	IoU	LPIPS	SSIM	IoU	LPIPS	SSIM	IoU	LPIPS
SF[63]	0.84	0.43	0.38	0.86	0.51	0.35	**0.87**	**0.53**	**0.34**
MF[8]	0.83	0.49	0.39	**0.86**	0.59	0.34	0.84	**0.60**	**0.33**
M2F[9]	0.84	0.45	0.37	0.85	**0.48**	**0.35**	**0.86**	**0.48**	0.36
GS2[30]	0.83	0.40	0.39	**0.84**	**0.44**	**0.37**	**0.84**	0.42	**0.37**

point clouds of relative accuracy 1-3 cm [1], manually 4-point projected perspective terrestrial optical images of the digital camera (Sony α7), and corresponding GSV Images [13], located in Munich, Germany. This dataset is part of the TUM2TWIN initiative [54]. We deem LoD3 opening masks as ground-truth owing to their superior accuracy and no distortions present, unlike manually rectified perspective images. In this experiment, we used 238 windows and 38 door instances captured from various building facades. The data is available under the project page: https://wenzhaotang.github.io/Texture2LoD3/.

4.1. Results and Discussion

3D Facade Segmentation as Texture Quality Measure We evaluate 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). For the supervised methods, we leverage the pre-trained on ADE20K [65], fine-tuned on the CMP dataset [56]; For the open-set experiments, we use the text prompt *"window"* and *"door"*.

The quality of facade segmentation serves as an effective proxy for evaluating texture quality in 3D building models. We employ three distinct metrics to comprehensively assess segmentation performance: Structural Similarity Index Measure (SSIM) [58], mean Intersection over Union (mIoU), and Learned Perceptual Image Patch Similarity (LPIPS) [64]. SSIM measures the perceived quality between images and correlates with human visual perception, IoU quantifies the spatial overlap accuracy between predicted and ground truth segments, and LPIPS captures perceptual similarities using deep feature representations that align with human judgments of visual similarity.

We compared three texture processing methods: unrectified imagery (*w/o surface rectification*), manual 4-point rectification (*ReLoD3*), and our automatic approach (*Texture2LoD3*). While unrectified imagery projection is the standard projection procedure, manual 4-point rectification represents the current standard in many practical workflows and relies on manual corner selection, our Texture2LoD3 uses geometric data from LoD1/2 models for automatic alignment. All images were captured at a consistent height (approx. 1.7m) to reduce alignment errors. To ensure fair segmentation comparison, we apply test-time adaptation for mask evaluation across all methods (see supplementary material for details).

As shown in Tab. 1, all segmentation models benefit significantly from accurate texture rectification, with consistent performance improvements visible across all metrics. The baseline approach without rectification achieves the lowest scores due to perspective distortions complicating the segmentation task. The 4-point-manual method delivers noticeable improvements, particularly in IoU scores, demonstrating the value of perspective correction in facade analysis. Our Texture2LoD3 approach consistently outperforms the w/o surface rectification baseline across all models and metrics and can replace manual projections. SegFormer exhibits the most substantial gains, achieving an SSIM of 0.87, IoU of 0.53, and LPIPS of 0.34 when combined with our method. This result represents improvements of 3% in SSIM and 10% in IoU compared to the unrectified baseline and 1-2% improvement over the manual rectification approach.

The qualitative results in Fig. 7 and Fig. 6 visually confirm these quantitative findings. Fig. 7 demonstrates how our Texture2LoD3 method produces cleaner segmentation boundaries and more consistent element detection across various building facades. The improvement is particularly evident in buildings with complex architectural features and elongated facades. Fig. 6 highlights performance differences through visual comparisons of a facade segmented by various methods. Texture2LoD3 produces results that align more closely with both the ground truth and geometric model, as shown by higher mIoU scores. This demonstrates that geometry-aware texture processing improves segmentation, with Texture2LoD3 outperforming manual methods without requiring labor-intensive intervention.

Limitations and Future Work The Texture2LoD3 method leverages the worldwide ubiquity of both semantic 3D models and panoramic street-view images, which shall open worldwide availability of so-far scarce LoD3 models. Yet, caution must be exercised as our rectified images are obtained from GSV images; there still may be some occlusions present concealing facades; the image quality is also highly dependent on the lighting conditions at the time the GSV images were captured. The identified hyper-parameters were consistently applied to our ReLoD3 dataset, yet further experiments must be undertaken to prove their computational efficiency and scalability, e.g., in architecturally different scenes of Asia.

5. Conclusion

In this paper, we introduce Texture2LoD3, a method enabling LoD3 building reconstruction by accurately projecting widely available street-level panoramic images onto surfaces of low-detail semantic 3D building models. Our work has led us to the conclusion that such a method can unlock worldwide availability of LoD3 models, as our automatic results outperform standard projections (by 11% IoU) and can replace manual texture projections (positive 1% IoU difference). Crucially, we also observe the qualitative advantage of our method, as it is less prone to perspective distortions when compared to manual perspective image projection or projecting without any surface rectification. Moreover, by employing prior low-detail semantic 3D building models as projection targets, we maintain the essential requirements of georeferencing, watertightness, and low-poly representation, extended by texture semantics. Owing to the absence of datasets allowing for such developments, we present the ReLoD3 texturing benchmark dataset, which will facilitate further research on LoD3 building reconstruction from images.

Acknowledgments The work was conducted within the framework of the Leonhard Obermeyer Center at TUM and the ReLoD3 project of TUM and NUS. We are grateful for the diligent work of the TUM2TWIN members, especially to Franz Hanke who meticulously collected images and manually projected onto building models.

References

[1] 3D Mapping Solutions. MoSES mobile mapping platform - technical details. https://www.3d-mapping.de/ueber-uns/unternehmensbereiche/data-acquisition/unser-vermessungssystem/, 2023. Accessed: 2023-01-30. 7

[2] 3DIS. CityEditor. https://www.3dis.de/cityeditor/, 2025. Accessed: 2025-03-22. 3

[3] Filip Biljecki, Hugo Ledoux, Jantien Stoter, and Junqiao Zhao. Formalisation of the level of detail in 3D city modelling. *Computers, Environment and Urban Systems*, 48:1–15, 2014. 2

[4] Filip Biljecki, Jantien Stoter, Hugo Ledoux, Sisi Zlatanova, and Arzu Çöltekin. Applications of 3D city models: State of the art review. *ISPRS International Journal of Geo-Information*, 4(4):2842–2889, 2015. 1

[5] Claus Brenner and Nora Ripperda. Extraction of facades using RJMCMC and constraint equations. *Photogrammetric Computer Vision*, 36:155–160, 2006. 2

[6] M. Buyukdemircioglu and S. Oude Elberink. Automated texture mapping cityjson 3d city models from oblique and nadir aerial imagery. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, X-4/W5-2024:87–93, 2024. 2

[7] Manuela F Cerón-Viveros, Wolfgang Maass, and Jiaojiao Tian. Oa-winseg: Occlusion-aware window segmentation with conditional adversarial training guided by structural prior information. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2025. 2

[8] Bowen Cheng, Alexander G. Schwing, and Alexander Kirillov. Per-pixel classification is not all you need for semantic segmentation. 2021. 7, 8, 1

[9] Bowen Cheng, Ishan Misra, Alexander G. Schwing, Alexander Kirillov, and Rohit Girdhar. Masked-attention mask transformer for universal image segmentation. 2022. 7, 8, 1

[10] Andreas Fabri and Sylvain Pion. Cgal: The computational geometry algorithms library. In *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, pages 538–539, 2009. 4

[11] Raghudeep Gadde, Renaud Marlet, and Nikos Paragios. Learning grammars for architecture-specific facade parsing. *International Journal of Computer Vision*, 117(3):290–316, 2016. 2

[12] Sean Gillies, Casper van der Wel, Joris Van den Bossche, Mike W. Taves, Joshua Arnott, Brendan C. Ward, et al. Shapely, 2022. 3

[13] Google. Google Street View. https://www.google.com/maps, 2023. Accessed: 2025-03-22. 7

[14] Eleonora Grilli and Fabio Remondino. Machine learning generalisation across different 3D architectural heritage. *ISPRS International Journal of Geo-Information*, 9(6):379, 2020. 2

[15] Eleonora Grilli, Elisa Mariarosaria Farella, Alessandro Torresani, and Fabio Remondino. Geometric features analysis for the classification of cultural heritage point clouds. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W15:541–548, 2019. 1

[16] Gerhard Gröger, Thomas H Kolbe, Claus Nagel, and Karl-Heinz Häfele. OGC City Geography Markup Language CityGML Encoding Standard, 2012. Open Geospatial Consortium: Wayland, MA, USA, 2012. 2, 3, 6

[17] Norbert Haala and Martin Kada. An update on automatic 3D building reconstruction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(6):570 – 580, 2010. 1

[18] Harshit, Pallavi Chaurasia, Sisi Zlatanova, and Kamal Jain. Low-cost data, high-quality models: A semi-automated approach to lod3 creation. *ISPRS International Journal of Geo-Information*, 13(4):119, 2024. 2

[19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2961–2969, 2017. 2

[20] Simon Hensel, Steffen Goebbels, and Martin Kada. 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*, IV-2/W5:37–44, 2019. 2

[21] Ludwig Hoegner and Georg Gleixner. 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*, XLIII-B2-2022:387–394, 2022. 1

[22] Ludwig Hoegner and Uwe Stilla. Texture extraction for building models from ir sequences of urban areas. In *2007 Urban Remote Sensing Joint Event*, pages 1–6. IEEE, 2007. 2

[23] Yujun Hou, Matias Quintana, Maxim Khomiakov, Winston Yap, Jiani Ouyang, Koichi Ito, Zeyu Wang, Tianhong Zhao, and Filip Biljecki. Global streetscapes—a comprehensive dataset of 10 million street-level images across 688 cities for urban science and analytics. *ISPRS Journal of Photogrammetry and Remote Sensing*, 215:216–238, 2024. 1

[24] Hai Huang, Mario Michelini, Matthias Schmitz, Lukas Roth, and Helmut Mayer. LOD3 building reconstruction from multi-source images. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B2-2020:427–434, 2020. 1, 2

[25] Dorota Iwaszczuk, Ludwig Hoegner, and Uwe Stilla. Matching of 3d building models with ir images for texture extraction. In *2011 Joint Urban Remote Sensing Event*, pages 25–28. IEEE, 2011. 2

[26] Martin Kada, Darko Klinec, and Norbert Haala. Facade texturing for rendering 3d city models. In *ASPRS Conference*, pages 78–85, 2005. 2

[27] Tom Kelly, John Femiani, and Peter Wonka. Winsyn:: A high resolution testbed for synthetic data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 22456–22465, 2024. 2

[28] Thomas H. Kolbe and Andreas Donaubauer. Semantic 3D city modeling and BIM. In *Urban Informatics*, pages 609–636, Singapore, 2021. Springer Singapore. 2

[29] Filip Korc and Wolfgang Förstner. eTRIMS image database for interpreting images of man-made scenes. *Deptartment of*

Photogrammetry, University of Bonn, Technical Report TR-IGG-P-2009-01, pages 1–12, 2009. 1, 2

[30] Feng Li, Hao Zhang, Peize Sun, Xueyan Zou, Shilong Liu, Jianwei Yang, Chunyuan Li, Lei Zhang, and Jianfeng Gao. Semantic-sam: Segment and recognize anything at any granularity. *arXiv preprint arXiv:2307.04767*, 2023. 5, 7

[31] Rui-Lin Lin. Image-stitching-opencv. https://github.com/linrl3/Image-Stitching-OpenCV, 2024. Accessed: 2025-03-23. 5

[32] Hantang Liu, Yinghao Xu, Jialiang Zhang, Jianke Zhu, Yang Li, and Steven CH Hoi. DeepFacade: A deep learning approach to facade parsing with symmetric loss. *IEEE Transactions on Multimedia*, 22(12):3153–3165, 2020. 2

[33] Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, et al. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *arXiv preprint arXiv:2303.05499*, 2023. 2

[34] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60:91–110, 2004. 2

[35] Francesca Matrone, Eleonora Grilli, Massimo Martini, Marina Paolanti, Roberto Pierdicca, and Fabio Remondino. Comparing machine and deep learning methods for large 3D heritage semantic segmentation. *ISPRS International Journal of Geo-Information*, 9(9):535, 2020. 2

[36] Helmut Mayer and Sergiy Reznik. Building facade interpretation from uncalibrated wide-baseline image sequences. *ISPRS Journal of Photogrammetry and Remote Sensing*, 61 (6):371–380, 2007. 2

[37] Przemyslaw Musialski, Peter Wonka, Daniel G Aliaga, Michael Wimmer, Luc Van Gool, and Werner Purgathofer. A survey of urban reconstruction. *Computer graphics forum*, 32(6):146–177, 2013. 2

[38] Hui En Pang and Filip Biljecki. 3d building reconstruction from single street view images using deep learning. *International Journal of Applied Earth Observation and Geoinformation*, 112:102859, 2022. 2

[39] Bryan G Pantoja-Rosero, Radhakrishna Achanta, Mateusz Kozinski, Pascal Fua, Fernando Perez-Cruz, and Katrin Beyer. Generating LoD3 building models from structure-from-motion and semantic segmentation. *Automation in Construction*, 141:104430, 2022. 1, 2

[40] Roberto Pierdicca, Marina Paolanti, Francesca Matrone, Massimo Martini, Christian Morbidoni, Eva Savina Malinverni, Emanuele Frontoni, and Andrea Maria Lingua. Point cloud semantic segmentation using a deep learning framework for cultural heritage. *Remote Sensing*, 12(6):1005, 2020. 2

[41] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PmLR, 2021. 5

[42] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *Advances in Neural Information Processing System (NeurIPS)*, 28, 2015. 1

[43] Tianhe Ren, Shilong Liu, Ailing Zeng, Jing Lin, Kunchang Li, He Cao, Jiayu Chen, Xinyu Huang, Yukang Chen, Feng Yan, et al. Grounded sam: Assembling open-world models for diverse visual tasks. *arXiv preprint arXiv:2401.14159*, 2024. 8, 1

[44] Hayko Riemenschneider, Ulrich Krispel, Wolfgang Thaller, Michael Donoser, Sven Havemann, Dieter Fellner, and Horst Bischof. Irregular lattices for complex shape grammar facade parsing. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1640–1647, 2012. 2

[45] Chiara Romanengo, Bianca Falcidieno, and Silvia Biasotti. Discretisation of the hough parameter space for fitting and recognising geometric primitives in 3d point clouds. *Mathematics and Computers in Simulation*, 228:73–86, 2025. 2

[46] Robert Roschlaub and Joachim Batscheider. An INSPIRE-conform 3D building model of Bavaria using cadastre information, LiDAR and image matching. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLI-B4:747–754, 2016. 1

[47] Abbas Salehitangrizi, Shabnam Jabari, Michael Sheng, and Yun Zhang. 3D modeling of façade elements using multi-view images from mobile scanning systems. *Canadian Journal of Remote Sensing*, 50(1):2309895, 2024. 2

[48] sk-zk. streetlevel: A street-level dataset and benchmark. https://github.com/sk-zk/streetlevel, 2024. Accessed: 2025-03-22. 3

[49] Yanfei Su, Weiquan Liu, Zhimin Yuan, Ming Cheng, Zhihong Zhang, Xuelun Shen, and Cheng Wang. Dla-net: Learning dual local attention features for semantic segmentation of large-scale building facade point clouds. *Pattern Recognition*, 123:108372, 2022. 2

[50] Richard Szeliski. *Computer vision: Algorithms and applications*. Springer Science & Business Media, 2010. 1, 2

[51] YKA Tan, LK Kwoh, and SH Ong. Large scale texture mapping of building facades. *IAPRS B37*, pages 687–691, 2008. 2

[52] Yue Tan, Olaf Wysocki, Ludwig Hoegner, and Uwe Stilla. Classifying point clouds at the facade-level using geometric features and deep learning networks. *International 3D GeoInfo Conference 2023, Recent Advances in 3D Geoinformation Science*, pages 391–404, 2023. 2

[53] Trimble. SketchUp Pro. https://www.sketchup.com/en/benefits-of-sketchup, 2025. Accessed: 2025-03-22. 3

[54] TUM2TWIN team. TUM2TWIN. https://tum2t.win/, 2023. [Accessed 27-11-2024]. 7

[55] Sebastian Tuttas and Uwe Stilla. Reconstruction of façades in point clouds from multi aspect oblique ALS. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, II-3/W3:91–96, 2013. 2

[56] Radim Tyleček and Radim Šára. Spatial pattern templates for recognition of objects with regular structure. In *German Conference on Pattern Recognition, Saarbrücken, Germany, September 3-6, 2013*, pages 364–374. Springer, 2013. 1, 2, 8

[57] Yuefeng Wang, Wei Jiao, Hongchao Fan, and Guoqing Zhou. A framework for fully automated reconstruction of semantic building model at urban-scale using textured lod2 data. *ISPRS Journal of Photogrammetry and Remote Sensing*, 216: 90–108, 2024. 1, 2

[58] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004. 8

[59] Olaf Wysocki, Yan Xia, Magdalena Wysocki, Eleonora Grilli, Ludwig Hoegner, Daniel Cremers, and Uwe Stilla. Scan2LoD3: Reconstructing semantic 3D building models at LoD3 using ray casting and Bayesian networks. *IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 6547–6557, 2023. 1, 2, 5

[60] Olaf Wysocki, Benedikt Schwab, Christof Beil, Christoph Holst, and Thomas H Kolbe. Reviewing open data semantic 3D city models to develop novel 3D reconstruction methods. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 48:493–500, 2024. 1, 2

[61] Olaf Wysocki, Benedikt Schwab, Bruno Willenborg, and Marija Knezevic. Awesome CityGML. https://github.com/OloOcki/awesome-citygml, 2024. Accessed: 2024-01-30. 1

[62] Olaf Wysocki, Yue Tan, Thomas Froech, Yan Xia, Magdalena Wysocki, Ludwig Hoegner, Daniel Cremers, and Christoph Holst. ZAHA: Introducing the Level of Facade Generalization and the Large-Scale Point Cloud Facade Semantic Segmentation Benchmark Dataset. In *Proceedings of the Winter Conference on Applications of Computer Vision (WACV)*, pages 7637–7647, 2025. 2

[63] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M Alvarez, and Ping Luo. Segformer: Simple and efficient design for semantic segmentation with transformers. *Advances in neural information processing systems*, 34: 12077–12090, 2021. 7, 8, 1

[64] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 8

[65] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 633–641, 2017. 8, 1

[66] P. Zhu, W. R. Para, A. Fruehstueck, J. Femiani, and P. Wonka. Large scale architectural asset extraction from panoramic imagery. *IEEE Transactions on Visualization and Computer Graphics*, pages 1–1, 2020. 1

[67] Peihao Zhu, Wamiq Reyaz Para, Anna Frühstück, John Femiani, and Peter Wonka. Large-scale architectural asset extraction from panoramic imagery. *IEEE Transactions on Visualization and Computer Graphics*, 28(2):1301–1316, 2020. 4

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-

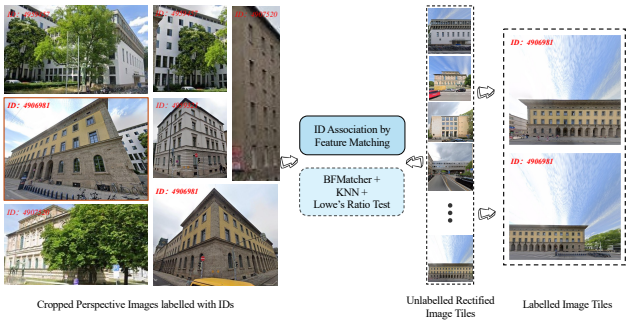


Figure 8. Pipeline of building ID association. The left side illustrates labeled building images obtained from CityGML data, while the right side presents rectified unlabelled facade tiles. The association is performed using feature matching (BFMatcher + KNN + Lowe's Ratio Test) to automatically establish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

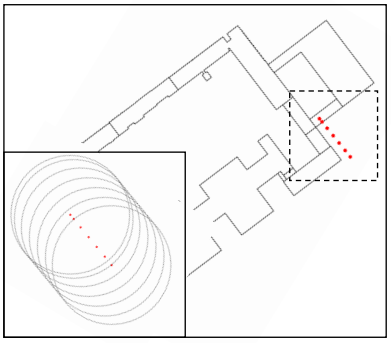


Figure 9. Schematic illustration of building footprint (black), sampling points (red), and the buffer area (gray dashed circles). The buffer defines a maximum distance from each sampling point within which building facades can be captured or considered visible. This ensures coverage of the building's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

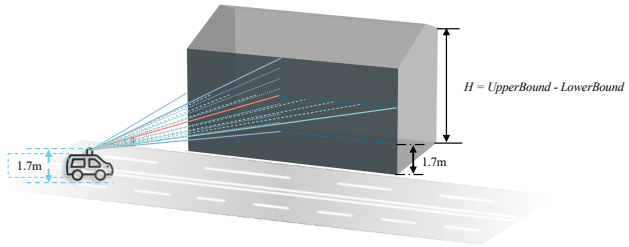


Figure 10. An illustration of our raycasting-based texturing setup. The camera (e.g., mounted on a vehicle at 1.7 m height) casts multiple rays toward the building's facade, which extends from the lower bound to the upper bound obtained from the GML data. We sample horizontal angles between the left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

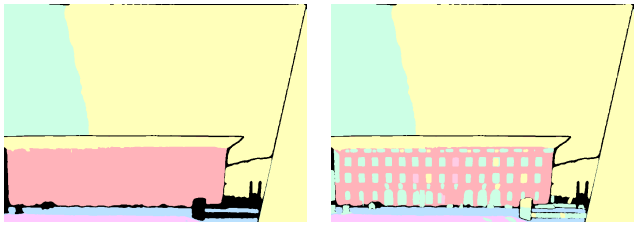
the distortion-free and cm-accurate LoD3 masks shall be treated as the ground truth.

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

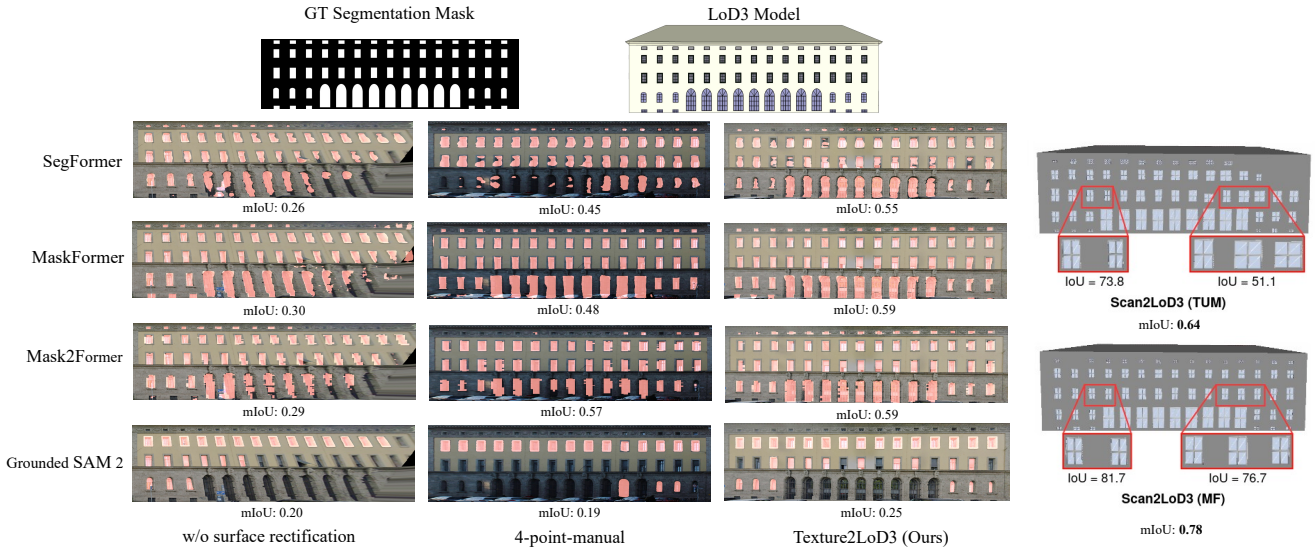


Figure 12. Comparison of the baselines and the Texture2LoD3 method to the Scan2LoD3 method leveraging multi-modal fusion of laser scanning, 3D model priors, and street-level images. Such an approach clearly outperforms only image and model combinations. Yet such a multi-modal setup is scarcely available 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.