Deep learning-based Scalable Image-to-3D Facade Parser for generating thermal 3D building models

Citation for the original published paper (version of record):
Yu, Y., Gonzalez Caceres, A., Scheidegger, S. et al (2025). Deep learning-based Scalable Image-to-3D Facade Parser for generating thermal 3D building models. Automation in Construction, 179. http://dx.doi.org/10.1016/j.autcon.2025.106449

Automation in Construction 179 (2025) 106449

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Deep learning-based Scalable Image-to-3D Facade Parser for generating thermal 3D building models

Yinan Yu [a],*, Alex Gonzalez-Caceres [b], Samuel Scheidegger [c], Sanjay Somanath [b], Alexander Hollberg [b]

[a] Department of Computer Science and Engineering, Chalmers University of Technology, Gothenburg, 412 96, Sweden
[b] Department of Architecture and Civil Engineering, Chalmers University of Technology, Gothenburg, 412 96, Sweden
[c] Asymptotic AI, Gothenburg, Sweden

ARTICLE INFO

Keywords:
Thermal 3D models
Building renovation
Orthographic images
Neural Radiance Fields (NeRF)
Ensemble learning
Window detection
LoD3
Window-to-wall ratio

ABSTRACT

Renovating existing buildings is essential for climate impact. Early-phase renovation planning requires simulations based on thermal 3D models at Level of Detail (LoD) 3, which include features like windows. However, scalable and accurate identification of such features remains a challenge. This paper presents the Scalable Image-to-3D Facade Parser (SI3FP), a pipeline that generates LoD3 thermal models by extracting geometries from images using both computer vision and deep learning. Unlike existing methods relying on segmentation and projection, SI3FP directly models geometric primitives in the orthographic image plane, providing a unified interface while reducing perspective distortions. SI3FP supports both sparse (e.g., Google Street View) and dense (e.g., hand-held camera) data sources. Tested on typical Swedish residential buildings, SI3FP achieved approximately 5% error in window-to-wall ratio estimates, demonstrating sufficient accuracy for early-stage renovation analysis. The pipeline facilitates large-scale energy renovation planning and has broader applications in urban development and planning.

1. Introduction

In the pursuit of mitigating climate change, the renovation of existing buildings plays a key role. Bringing buildings up to modern energy standards is one of the key strategies to reduce energy consumption and greenhouse gas emissions [1]. This is especially true for European apartment buildings built in the 1970s and earlier, before energy performance regulations were introduced [2]. Energy simulation is most commonly used to evaluate potential renovation alternatives before implementation and to support decision-makers in picking the right one. As decision-makers are often building owners of large portfolios, including hundreds of buildings, there is a need for efficient and scalable solutions for evaluating this building stock. Central to any building energy simulation is a thermal 3D model of the building that includes its most important heat transfer elements, such as walls, roofs, and windows. Unfortunately, as many older buildings lack up-to-date floor plans or CAD files [3], energy modelers often turn to municipal or cadastral data, which rarely exceed Level of Detail (LoD2)- which represent buildings as simple block models without facade details like windows. LoD3.0 models, on the other hand, capture roof details with higher accuracy while other features remain at a lower level of detail. Although LoD3.0 models already contain computational information for various applications, the taxonomy and refined structures of windows are often ignored [4]. However, including windows in thermal 3D modeling is crucial for calculating solar gains and inaccuracies in estimating the buildings envelope can result in incorrect evaluations of retrofit measures. [5] showed that halving the Window-to-Wall Ratio (WWR), from 0.28 to 0.14, cuts heating loads by 9.4%–13.3%, highlighting the strong linear sensitivity of energy demand to WWR. The required level of detail that includes windows is defined as LoD3.1 according to [6].

Existing solutions fall into two categories: top-down and bottom-up approaches. Top-down approaches, such as the use of archetype libraries like IEE-TABULA [7], model cities using average buildings but underestimate real-world variety [8], making them unsuitable for informing building-level renovations [9]. Bottom-up approaches, such as the physical modeling method [10], rely on detailed thermodynamic modeling of individual buildings [11,12]. LiDAR scanning is a key technology here, offering high-precision 3D data. However, LiDAR has notable drawbacks: high equipment and processing costs, lack of semantic information, and reliance on labor-intensive manual annotation [13]. These barriers limit its scalability for large-scale building portfolio analysis.

* Corresponding author.
E-mail address: yinan@chalmers.se (Y. Yu).

https://doi.org/10.1016/j.autcon.2025.106449
Received 9 October 2024; Received in revised form 31 July 2025; Accepted 1 August 2025
Available online 15 August 2025

To address these challenges, researchers have explored cameras as primary 3D reconstruction sensors. Cameras are low-cost, flexible, and benefit from advances in computer vision and deep learning (see Section 2.2). They enable semantic analysis through object detection and instance segmentation. However, camera-based methods also face limitations, including perspective distortion, occlusions, and the need for dense image collections for reliable 3D reconstruction.

Research gap. Despite progress, to our knowledge, there is no end-to-end, fully automated pipeline that generates LoD3.1 building models from images with semantically accurate window and facade details. Top-down archetypes are too generalized for individual building renovations, and LiDAR-based approaches, while accurate, are costly and require extensive manual annotation and post-processing. Hence, there remains a critical need for a scalable, cost-effective method to generate LoD3.1 models with sufficient geometric and semantic detail for early-stage energy renovation analysis.

The aim of this paper is to develop a pipeline that generates thermal LoD3.1 models using scalable image-based data sources. Scalability considers not only equipment cost and data collection effort but also the complexity of data processing. Our goal is to support building portfolio owners in selecting optimal renovation solutions efficiently.

The overarching research question we address is:

Which data sources and algorithms can be applied to generate thermal 3D models for building energy renovation with LoD3-level detail and sufficient accuracy?

We propose a camera-based pipeline demonstrated on three residential buildings from the 1961–1975 period, a representative case for energy renovation in Europe. Our design minimizes the need for annotated training data and model fine-tuning across different built environments.

Our main contributions are as follows:

1. We propose a unified framework that accommodates both sparse and dense data collections, enabling flexible and scalable 3D facade modeling under varying data availability conditions.
2. For sparse street-level imagery (e.g., Google Street View), we introduce an ensemble-based fusion method (Algorithm 6) that aggregates multiple partial orthographic views to improve robustness against occlusions, viewpoint variations, and localization noise.
3. For dense image collections, we leverage Neural Radiance Fields (NeRF) not only for 3D reconstruction but specifically for generating detailed facade renderings, allowing direct computation of true orthographic images via surface-based parallel projection.
4. Across both sparse and dense workflows, we apply orthographic transformations to standardize the data representation, simplify feature detection and geometry parameterization, and improve dimensional accuracy by mitigating perspective distortion (illustrated in Fig. 1). The orthographic transformation minimizes perspective distortion and simplifies the parameterization of facade features.

Our pipeline effectively bridges the gap between scalability and accuracy in generating LoD3.1 building models required for thermal energy simulation, aiming to support decision-makers in evaluating renovation options and contribute to the broader goal of reducing energy consumption and mitigating climate change.

2. Background and related work

To select the most suitable sensors to enable scalable and efficient data collection, we first describe the relevant background on different sensors. We then review related works using these sensors for facade analysis and the generation of 3D models.

2.1. Sensor types

We categorize the sensor choices into two types: primary sensor and complementary information. The primary sensor is the main system used for data collection, while the complementary information includes additional data sources that enhance the accuracy and detail of the 3D models.

2.1.1. Primary sensor

We differentiate between three commonly used sensor types: (1) LiDAR, (2) mono cameras, and (3) stereo cameras.

LiDAR (Light detection and ranging). LiDAR technology [14,15] stands out for its high precision in mapping the 3D structure of buildings, offering relatively quick data acquisition over large areas, which makes it a suitable choice for both small and large-scale projects. Further, it performs well under various lighting conditions. However, it is challenging to extract semantic information from 3D data due to the inherently complexity of three-dimensional pattern recognition. Moreover, the nature of 3D point clouds, often characterized by their sparsity, further complicates the task of semantic analysis. This is especially problematic when measuring transparent objects such as windows or dark objects. In addition, the high cost of LiDAR systems may limit its accessibility for smaller projects.

(Mono) camera. Using cameras for 3D reconstruction offers an accessible and cost-effective method for creating detailed 3D models of buildings, leveraging the widespread availability of consumer-grade devices. Cameras excel at capturing facade textures and colors, enabling semantic analysis such as object detection [16,17] and instance segmentation [18]. Flexible deployment options, including handheld and drone-mounted setups, further enhance their utility across diverse project scales.

- **Perspective images:** 3D reconstruction from video sequences is well-established in computer vision and photogrammetry [19]. Recent advances in Artificial Intelligence (AI) and Deep Learning (DL) have significantly improved both visual quality and geometric accuracy. Dense video capture around a structure allows each frame to serve as a perspective image from a distinct camera pose.
- **Panoramic images:** Panoramic images [20,21], captured using wide-angle lenses, multi-camera rigs, or rotating systems, collect rays from a broad field of view, enabling efficient coverage of large facades with fewer captures.
- **Orthographic images:** Orthographic images [22,23] use parallel projection to represent scenes without perspective distortion, preserving true scale and geometry. They are particularly useful for accurate geometric measurements and facade element parameterization. However, standard cameras inherently capture perspective images, where light rays converge at a focal point, causing distortion with distance. As a result, true orthographic views are not captured directly and are typically generated through post-processing of reconstructed 3D models.

Despite their advantages, camera-based methods face challenges. High-quality capture depends heavily on lighting conditions, and post-processing is often required to produce accurate 3D models. Artifacts such as rolling shutter effects and motion blur further complicate reconstruction. Additionally, orthographic projections are sensitive to the quality of the underlying 3D model, making robust reconstruction critical when indirect computation is required.

Stereo camera. These cameras emulate human binocular vision to capture 3D data, providing an accessible and cost-effective alternative to LiDAR [24–26]. They are valued for their affordability, simplicity, and versatility, supporting both indoor and outdoor mapping with relatively straightforward data processing. However, stereo cameras offer lower accuracy than LiDAR, particularly over long distances, and their performance can degrade under poor lighting conditions or on reflective or smooth surfaces.

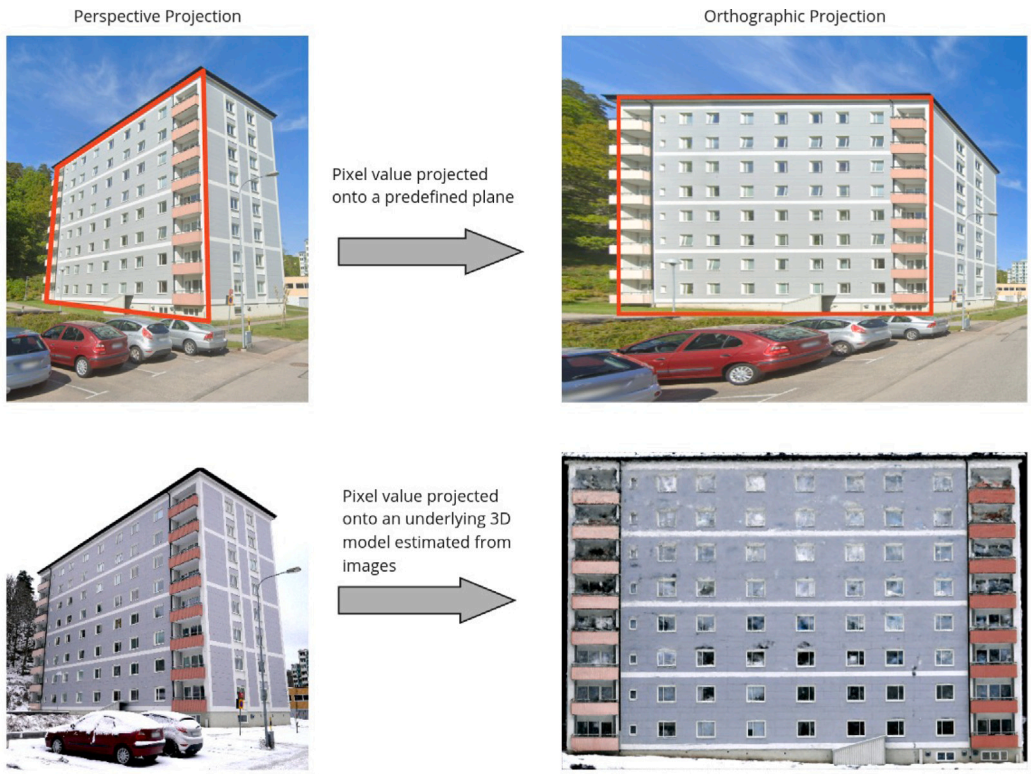
Perspective Projection
Orthographic Projection
Pixel value projected onto a predefined plane
Pixel value projected onto an underlying 3D model estimated from images

Fig. 1. Comparison of perspective images with proposed orthographic images for both sparse images (first row; from Google Street View) and dense images (second row; using a perspective camera).

Summary. LiDAR excels at capturing structural dimensions but struggles to differentiate scene elements, such as windows versus walls, without additional data layers, and its high cost can be prohibitive. Stereo cameras provide direct scale but require a more complex setup and can suffer from reduced accuracy, especially over long distances. In contrast, monocular cameras, combined with scale estimation techniques, offer a flexible and cost-effective solution for capturing detailed spatial data. For thermal 3D modeling, monocular cameras provide the necessary resolution, detail, and operational flexibility, making them a preferred choice for efficient, low-cost model generation.

2.1.2. Complementary 3D information

Densely collected monocular images can generate 3D point clouds via photogrammetry, but these lack absolute scale, which is crucial for 3D modeling. To address this, external scale information must be integrated. In this section, we introduce common sources for obtaining such scale data.

Ground control points (GCP). GCPs are specific, accurately surveyed points on the Earth's surface used to georeference aerial or satellite imagery to real-world coordinates [27]. They are critical for ensuring positional accuracy in mapping projects and are typically established through GNSS or traditional surveying methods.

Local coordinate reference points. Real-world spatial dimensions in the image plane can be estimated by measuring distances and angles between camera centers of registered images. These measurements enable a 3D similarity transformation to align the reconstructed model with a coordinate system, using known relative distances and angles as local reference points.

Known 3D structures. This approach uses known 3D structures, such as flat surfaces (planes), within a model's 3D space. Examples include photographed objects like building walls or ground surfaces with known real-world dimensions. Accurate real-world measurements and spatial relationships can be extracted from images using this information. Although this method assumes local flatness, which may not always hold for curved or uneven surfaces, it provides a simplified and effective way to represent large-scale urban environments.

Camera position and pose. When a localization sensor (e.g. a GNSS receiver) is available during the image capturing process, each image frame taken from the video sequence has an associated position in space, and the images can be anchored in a world coordinate system in 3D. This anchoring allows for the alignment and integration of multiple images into a coherent and unified 3D model, where the scale is consistent with the real-world. However, for data collection with low cost, an accurate localization sensor is often unavailable, making the scale estimation challenging. In such cases, a manual localization process can be employed.

2.2. Review of related work

There exist a few review papers on the topic of feature extraction using cameras and LiDAR sensors [28–31]. In this section, we briefly give an overview of this field.

2.2.1. Facade analysis in the camera image plane

Computer vision and image analysis are fundamental building blocks for facade parsing, as images contain rich semantic information through their pixel values. While analysis conducted solely in the image plane is insufficient for thermal 3D modeling, it remains essential in most facade parsing pipelines. In this section, we introduce articles that focus only on the image plane for facade analysis. In particular, we divide these techniques into three categories based on the geometric primitives they use to model facade features: points, rectangles, and polygons.

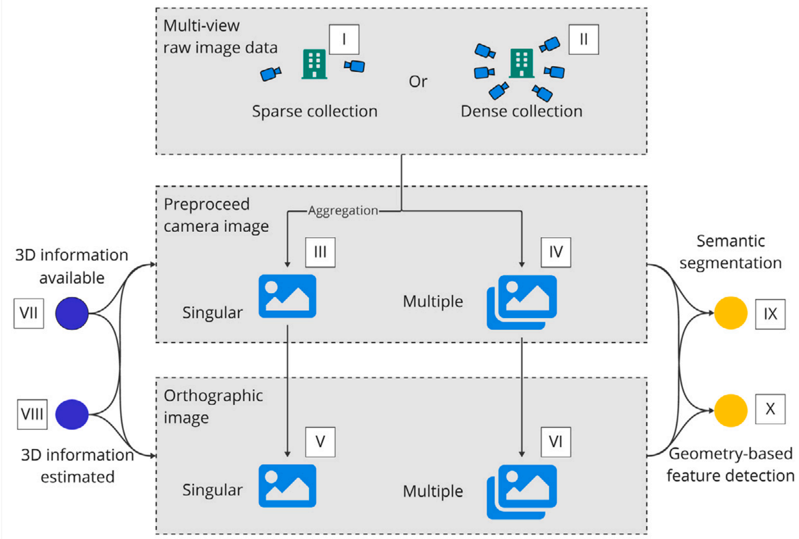

Fig. 2. Typical pipelines with camera images as primary input.

Pixel-wise segmentation. Several works have focused on pixel-wise segmentation for facade analysis. For instance, [32] used cut-out images of facades to perform segmentation of windows and doors using convolutional neural networks (CNN). Similarly, [33] proposed a method for semantic segmentation of windows and doors, introducing a symmetric loss function, enforcing most window predictions to be rectangular. [34] proposed a hierarchical deep learning framework that integrates several deep neural networks (PSPNet [35], DANet [36], and DETR [37]) for facade element detection to automatically detect various facade elements. Additionally, [38] used ResNet and BiFPN for instance segmentation, focusing on window detection and WWR calculation. [39] developed a Vision Transformer (ViT)-based semantic segmentation method combined with line detection to improve facade element detection, capturing the shapes of buildings, windows, doors, roofs, and other elements.

Bounding box detection. Despite providing valuable information, pixel-level segmentation is often an intermediate step because it lacks the geometric definition of facade elements typically required for subsequent analysis. In facade analysis, elements are commonly parameterized by rectangular bounding boxes. [40] utilized reinforcement learning and Markov decision processes for bounding box detection from orthographic images. Similarly, [41] evaluated various methods for segmentation and object detection, using Conditional Random Fields (CRF) for classification and applying weak architectural principles for optimization to produce bounding boxes. [42] proposed a method for converting perspective images to orthographic views, followed by detection using a soft cascaded classifier and post-processing to refine detections. This method assumed that windows are usually uniformly distributed within their rows or columns. [43] focused on wall segmentation and window bounding box detection, converting wall segmentations into bounding boxes across multiple datasets. This approach highlighted challenges with bounding box accuracy due to perspective camera distortions. [44] utilized Structured Random Forest (SRF) and a Region Proposal Network (RPN) based on a CNN for asset bounding boxes in orthographic images, while [45] used computer vision techniques and GSV images to detect windows and calculate the WWR.

Polygon detection. Polygon detection methods are designed to capture the exact shape of facade elements. [46] developed a method for detecting keypoints of windows from perspective images, learning keypoint relationships to group them into window polygons. [47] worked on detecting windows in aerial images using Hough forest classification and a proposed refinement of Hough voting for multiple object instances detection. Although this method is primarily designed for aerial images, its refinement approach is noteworthy. [48] proposed a method to detect windows and other elements in CAD drawings. Furthermore, [33, 49] explored orthorectification and facade segmentation and polygon detection from street view and satellite images, respectively.

Choosing a parameterization for downstream 3D modeling. Selecting a suitable parameterization is critical for 3D modeling, balancing expressiveness and ease of use for downstream applications. Pixel-wise segmentation offers the highest flexibility, capturing detailed and complex shapes. However, it is difficult to use for downstream tasks because each pixel must be processed individually, leading to increased complexity in handling and storing the data. Polygon-based approaches provide a middle ground, capable of representing various shapes with fewer parameters than pixel-wise methods, but still requiring more complex handling. Rectangular bounding boxes, while the least flexible, offer the simplest and most user-friendly parameterization. They are typically sufficient for modeling most facade features and are easier to manage and integrate into downstream applications. It is worth noting that bounding box detection is more accurate in orthographic images, as this eliminates distortions present in perspective images, thereby enhancing the accuracy of 3D projections.

2.2.2. 3D modeling using camera images as the primary sensor

Typical pipelines for facade analysis using camera images as the primary input are illustrated in Fig. 2. The process begins with raw data collection, which can vary in density. Sparse collection methods (I) include large-scale drive-by street-level image acquisition, such as GSV, while dense collection (II) methods involve dense image captures of specific areas or buildings. These multi-view images can be used for estimating unstructured 3D information (such as point clouds) or parameterized 3D models (such as planes) on LoD2.

After the initial data collection, preprocessing steps are employed, including correcting for camera lens distortion and other image quality adjustments. Following preprocessing, the images can either be aggregated into a single composite image by fusing multiple views or selecting the best images based on certain criteria or treated as individual images. These two approaches correspond to components III and IV in the pipeline, respectively. One additional step can be applied to correct perspective distortions in the images before semantic analysis on the facade. Perspective distortions refer to the phenomenon where parallel lines appear convergent and distant objects appear smaller. These distortions can be problematic where the geometries in the image plane need to be maintained when transformed into 3D. To address this, orthographic transformation (also referred to as orthorectification) can be applied to the images (V and VI). This transformation corrects perspective distortions, ensuring that parallel lines remain parallel and objects retain their consistent size regardless of their distance from the camera or their position in the image. To establish the correspondence between pixels in the image plane and their 3D location, additional information is necessary such as 3D models or building blueprints. If 3D information is available (VII), it can be directly applied to project each pixel to a 3D location. Otherwise, this information must be estimated (VIII) from the images.

Street-level imagery is commonly used for facade analysis and 3D modeling. These images capture static built environment features, pedestrians, cyclists, and vehicles, making them indispensable for various applications [50]. Several web-based providers offer street imagery, with GSV and Bing StreetSide being among the most prominent [51]. GSV, in particular, stands out as a vast online browsable dataset consisting of billions of georeferenced street-level panoramic images from around the world. Since its inception in 2007, GSV has continuously updated its global image database, capturing panoramas every 5–10 meters in urban environments [52]. Due to their large scale, these datasets are typically sparse around each individual building facade.

Some services, such as GSV API, also allows access to depth maps for specific panoramas, which can be decoded and visualized to reconstruct 3D planes. When these planes are not available, they can be estimated from the images using Structure-from-Motion (SfM) [19]. SfM creates 3D models from images by matching keypoints across multiple images, resulting in a 3D point cloud where each point is assigned a color based on the corresponding pixel value. When the images are not georeferenced, meaning that the camera poses are not available, SfM can estimate the poses. One can fit planes to these point clouds to obtain parameterized geometries subsequent analysis. While SfM is an effective method for 3D reconstruction, it requires dense image collection for effective keypoint matching.

When the data collection is dense, Neural Radiance Fields (NeRF) [53] is a relevant technique for 3D facade rendering. NeRF takes multi-view images and camera poses (measured or estimated from SfM) as input and achieves highly realistic rendering and novel viewpoint synthesis by training a neural network to predict color and density values along rays passing through a scene, effectively reconstructing complex lighting effects and details. To make the algorithm more accessible, [54] provided tools with enhanced hyperparameter selection and optimization through real-time visualization of the rendering process. Furthermore, [55] demonstrated that NeRF could be applied to large-scale city rendering using street-level multi-view images.

Given these underlying image-to −3D techniques, there are several notable studies in the literature. [56] introduced a method that uses multiple images for SfM to create a 3D mesh, focusing on segmenting and labeling the mesh from a single chosen image. [57] utilized multiple images to generate a mesh via SfM. They further applied a segmentation model to project segmentations onto the mesh, using overlapping images to determine the most frequent label for more accurate segmentation. [58] leveraged drone (UAV) images to create a point cloud via SfM, which was georeferenced using ground control points. They implemented a color-based two-step selection process to detect window patches in the point cloud. [59] transitioned from stereo [60] to single mono camera images for their 3D reconstruction. They utilized deep learning for depth estimation to create a 3D point cloud, followed by segmentation of architectural features like windows and doors which were then assigned to the 3D point cloud. Further, [61] monitor construction activity progress, generating orthographic views via single-view projection when a suitable camera view exists, or falling back to NeRF rendering when single views are inadequate (e.g., due to face size, proximity, or occlusion). Semantic segmentation on these views yields area-based completion percentages. Compared to the orthographic representation based on virtual camera traversal presented in [61], our approach leverages NeRF to compute true orthographic projections directly from surface geometry. While camera traversal could potentially provide high-quality renders, our design is particularly suited for residential facades with long continuous sides, where traversal approaches may fail due to limited field of view, increased risk of occlusions, and inconsistencies in scale and alignment across multiple renderings.

[62] took a different approach by using a single image with known building silhouettes in the image as inputs. Their method employs Convolutional Neural Networks (CNNs) to select a "building mass grammar" and estimate parameters, thereby determining the 3D structure of the building. This structure is used to create orthorectified facade images and to generate detailed 3D models including facade and window grammars. [63] focused on using multiple images for SfM to estimate planes. They combined segmentation with projecting segmentations to the closest planes, refining these masks to generate bounding boxes on 3D planes for enhanced accuracy in 3D modeling. [64] employed images captured from a moving vehicle equipped with GNSS/IMU for precise camera localization and measuring poses. Their process involved SfM to build a 3D mesh, masking buildings, performing semantic segmentation, and projecting these masks onto the mesh. The mesh was rotated to fit the x–y plane, and bounding boxes were used for facade features, also estimating the building age from the images. [65] proposed a dual-track method using multi-view images and building footprints. They utilized Faster R-CNN and the Segment Anything model [68] to project 2D borders into 3D using pinhole camera models and collinearity equations. LiDAR scanner data can also integrated into this workflow.

[66] utilized a 3D model as input to generate orthogonal facade images. They employed Faster R-CNN for bounding box detection and clustering-based window alignment to ensure consistency in window dimensions and positions. This method also included glass plane detection to enhance window detail and model floors and ceilings. [67] uses a grammar-based edge detection framework and a learning-based method utilizing CNNs and compares the result in New York and Lisbon. The paper finds that the learning-based method generally performs better, and proposes a hybrid approach to leverage strengths of both methods. In addition, [69] focused on creating an extensive instance segmentation dataset for interior and exterior scenes, aimed at 3D reconstruction. The dataset, aligned with standardized data schemas like Industry Foundation Classes (IFC), ensures accurate geometric representations and topological relationships in segmentation results.

In summary, these studies collectively highlight the evolution of 3D reconstruction techniques from 2D images, incorporating advanced segmentation methods, deep learning, and multi-view geometry to achieve increasingly accurate and detailed models. A summary of the aforementioned techniques can be found in Table 1.

2.2.3. *3D modeling using both camera and LiDAR as the primary sensors*

Although less relevant to our paper, as we primarily use cameras, point processing is still worth mentioning since SfM produces point clouds, making some techniques transferable. We include these existing works for completeness.

[70] explore the learning of weighted attributed context-free grammar rules for 3D building reconstruction. They employ an Support Vector Machine (SVM) for the classification of facade structures and an Multi-Layer Neural Network (MLN) for estimating parameters of facade parts. [71] present a hierarchical approach to facade point cloud analysis. Their method does not incorporate color information. Instead, they segment the point cloud into "principal facade planes" and "2.5D segments" (images with depth). The BieS algorithm extends the samples, and the ScSPM algorithm extracts features. A linear SVM classifier then categorizes each superpixel into semantic classes such as window, wall, roof, shop, or door based on the learned features. [72] focus on the creation of 3D models of cultural heritage buildings using RGB point clouds. Their work emphasizes the generation of

Table 1
Summary of related work (camera image as primary input for 3D modeling) and their respective processes and outcomes.

Method	Pipeline	I	II	III	IV	V	VI	VII	VIII	IX	X
[56]	II → I → VIII (point cloud, mesh) + IV → IX (semantic segmentation)	✓	✓		✓				✓	✓	
[57]	II → VIII (point cloud, mesh) + IV → IX (semantic segmentation)		✓		✓				✓	✓	
[62]	III + building silhouette → VIII (building mass grammar) + V → X (facade and window grammar)			✓		✓			✓		✓
[60]	Georeferenced stereoscopic images I → VIII + V → X	✓				✓			✓		✓
[58]	II → VIII → IX		✓						✓	✓	
[59]	III → VIII (depth estimation) → IX (3D rendering, semantic segmentation)			✓					✓	✓	
[63]	II → VIII (estimate planes → LoD2) + IV → IX (in images) + X (in 3D geometries)		✓		✓				✓	✓	✓
[64]	II → VIII (mesh, RGB from images) + IV → IX (in images) → X (in point clouds)		✓		✓				✓	✓	✓
[65]	II → VIII → III → IX → X		✓	✓					✓	✓	✓
[66]	VII + V → X (window detection and details for defining grammar)					✓		✓			✓
[61]	II + VII → VIII (SfM cloud/poses, NeRF model) + IV → {V (Proj. Transf.) | VI (NeRF Trav.)} → IX (Semantic Segmentation)		✓		✓	✓	✓	✓	✓	✓	
[67]	I + VII → V → X	✓				✓		✓			✓
SI3FP (ours) StreetView	I + VII (planes) → IV → VI → X (window detection)	✓			✓		✓	✓			✓
SI3FP (ours) Camera2D	II → VIII + IV → V → X (window detection)		✓		✓	✓			✓		✓

3D models without segmenting the point cloud. The models are created using Autodesk Revit software, with comparisons made between manual and automatic methods and the point cloud. [73] use both ortho and projection images or point clouds as input for segmentation. They apply boosting decision trees to segment the images and point clouds separately and evaluate their performance both individually and in combination. However, the paper does not address the challenges of combining point clouds with camera data. [74] focus on merging airborne and terrestrial point clouds, filtering out ground points, and extracting planes using RANSAC. They utilize oblique images taken from the sky to find the best angle for each plane, projecting the images onto the planes. The plane edges are aligned with line detections in the images to create a colored 3D model, which is manually edited to segment windows and other features. [75] use point clouds as input and apply RANSAC to segment balconies and windows. They construct a hierarchical graph representing the facade, with levels for the facade, floor, and window. Repeating objects are clustered in the graph based on their similar size and spacing.

2.2.4. Research gap and differences to our approach

Our methodology differs from existing techniques by avoiding the conventional process of "pixel-wise segmentation in image → projecting these labels to 3D points (derived from images) → defining geometry in 3D from these points". This traditional approach is challenging for downstream thermal modeling due to the difficulty in estimating window parameters in 3D when the point cloud derived from images is sparse and the segmentation results are erroneous. Instead, we estimate geometric primitives in the image plane. Specifically, we generate an orthographic image where perspective distortions are corrected to enhance accurate shapes and sizes. We then perform bounding box detection directly on this orthographic image. This method leverages the capability of semantic analysis and geometry parameterization in images (as opposed to sparse point clouds), leading to a more robust estimation of window geometry.

Note that orthographic transformation is crucial for estimating accurate geometry on the facade plane. Additionally, we utilize orthographic images as a unified interface for both sparse and dense data collection, enhancing usability by accommodating varying levels of data availability. SI3FP performs best when the feature surfaces parallel to the facade planes are flat, which is a realistic assumption for our use case.

For sparse data collection, most literature focuses on selecting the best single perspective image for detecting building facades. However, this approach is insufficient due to severe occlusion and perspective distortion that can occur even in the best images. Instead, we developed an ensemble method that aggregates all available images to enhance robustness and accuracy. This aggregation helps mitigate the impact of occlusions by combining information from multiple viewpoints.

Moreover, for dense data collection, there is a noticeable gap in research concerning the application of NeRF to building facade parsing. [76] conducted a case study on semantic segmentation of building structures using 3D point clouds. In contrast, our work applies NeRF to create a detailed photorealistic 3D render of buildings, followed by an orthographic transformation of the facade. By using NeRF, we achieve higher-quality reconstructions compared to traditional Multi-View Stereo (MVS) methods [77], particularly in handling complex lighting conditions and reflections common in urban environments. These images are subsequently used for window detection, employing pre-trained deep learning models. This approach simplifies the facade parsing process and enhances the reliability of the geometric estimation of building facades.

3. Methods

The SI3FP pipeline consists of two alternative paths offering their respective trade-offs: (1) the *StreetView* path for scalable inspection, and (2) the *Camera2D* path for targeted inspection. An overview of these two paths can be found in Fig. 3. Each path has its individual

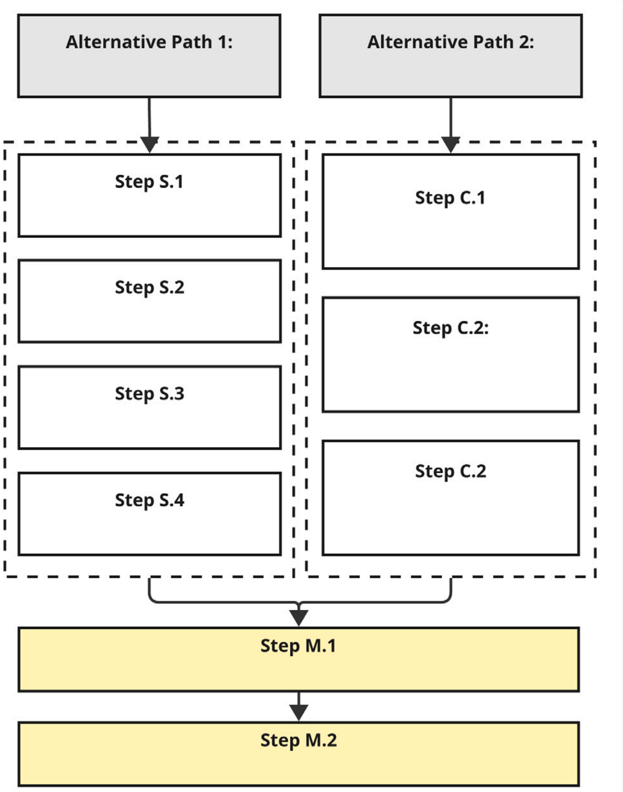

Fig. 3. Overview of SI3FP, which uses camera images as input to generate true-to-scale orthographic images for semantic facade parsing and 3D thermal modeling.

data collection and processing steps: four steps (S.1-S.4) for StreetView and three steps (C.1-C.3) for Camera2D. They then converge into two merged steps, denoted as M.1 and M.2 for semantic facade parsing and thermal modeling.

3.1. StreetView (S) for scalable inspection

Street level imagery can be obtained through multiple methods. By employing a mobile data collection platform, such as a vehicle or drone, equipped with video cameras, a gyroscope, distance sensor (such as a LiDAR or a radar system), and a location sensor (e.g. GNSS receivers), data can be captured across diverse environments at scale. Alternatively, access to large-scale street view datasets is possible through specialized services dedicated to offering extensive street view imagery. Although our focus is on data collection via Google's Street View (GSV) API, the outlined process can be adapted for use with other data sources or collection methods. An overview can be found in Fig. 4. The StreetView path consists of five steps described below (Step S.1-S.4).

Step S.1 data collection and filtering: Select panoramic images and meta data.

The complete input data for StreetView can be visualized in Fig. 6. The initial step involves selecting panoramic images. While an important use case is to collect these images continuously across various locations, for the sake of simplicity, we focus on illustrating the data collection method within a specific neighborhood that is identified by a central geographic coordinate (marked by its latitude and longitude) and a radius. Utilizing GSV API, we can pinpoint nearby panoramic views identified by a unique PanoID for the given geographical coordinates. Each PanoID represents a 360-degree panoramic image, providing detailed meta data such as latitude, longitude, altitude, heading (direction), pitch (angle of elevation), roll (axial tilt), capture date, and connections to adjacent views. The selection of the *origin view*, or starting point, is achieved by opting for the most recently captured image among the search results for the given location. This enables the exploration of various street view locations throughout the targeted neighborhood by traversing the links to neighboring views. When there are sufficiently many images available for one building, a filtering criterion is applied to include only those images captured on the same date to maintain consistency across the collected images (e.g. uniform lighting, contrast, and scene composition, among other environmental conditions). The outcome of this step is a set of panoramic images[1](cf. Fig. 5) with their respective meta data.

In addition to panoramic images, GSV provides plane definitions. More specifically, walls and ground surfaces are represented as planes in the world coordinate system. The information is typically gathered by a sensor that provides 3D information (e.g. a stereo camera or a LiDAR scanner), and it can be refined by retrieving the building locations and from databases of building locations and outlines offered by various official or unofficial organizations (sometimes referred to as the land registry).

Each panoramic image is associated with a camera pose. Each pixel in the panoramic image can be associated with a plane, allowing for transformation of the pixel value between the camera coordinate system and the world coordinate system in 3D. This association, if not readily given, can be estimated by standard computer vision techniques [19]. More precisely, to map a pixel from a panoramic image directly to a 3D point on a specified plane, we first convert pixel coordinates (x, y) to spherical angles θ and ϕ, and then to a unit direction vector $\mathbf{V}$. The intersection of $\mathbf{V}$ with the plane defined by normal vector $[a, b, c]$ and distance d is determined by scaling $\mathbf{V}$ by $\frac{d}{aV_x+bV_y+cV_z}$. This scaled $\mathbf{V}$ gives us the 3D coordinates.

Step S.2 plane clustering: Identify planes associated with the building of interest.

To extract 3D information and enable detailed analysis, each RGB panoramic image captured must be connected to a 3D framework. Particularly within GSV imagery, every panoramic image is linked to a set of 3D planes. In detail, for each panoramic image, there exists a plane association matrix sized 512 × 256. Each element within this matrix corresponds to a specific plane. The alignment of this matrix is in sync with the panoramic image from which it is derived, although it is a downsampled version relative to the original image's resolution (16384 × 8192). This can be visualized in Fig. 7. This is specific to how GSV defines the association between the panoramic image and their corresponding 3D structure. It is noteworthy that some pixels may lack a plane association. This can be caused by the limitation of the data collection equipment and process.

With the plane association matrix, in combination with the planes, the RGB data from the panoramic images can be mapped onto these planes, infusing the 3D structure with color and texture. This step results in a colored point cloud in 3D (cf. Fig. 8).

To enhance the robustness of the system, we repeat the aforementioned process for all nearby, potentially overlapping panoramic images collected in Step S.1 (cf. Fig. 9). Every panoramic image contributes its unique planes to the collective model. We then adjust these planes into a common coordinate framework through translation and rotation.

The next critical step is to cluster these transformed 3D planes from different panoramas to identify those corresponding to the same physical facade surface. We use the Agglomerative Clustering algorithm with average linkage. The distance $D(i, j)$ between any two plane segments i (unit normal $\mathbf{n}_i$, origin distance d_i) and j ($\mathbf{n}_j, d_j$) is computed using a custom metric:

$$D(i, j) = (1 - \mathbf{n}_i \cdot \mathbf{n}_j) + 0.01 \cdot |d_i - d_j| \tag{1}$$

[1] The size of each panoramic image provided by the current version is 16384 × 8192 pixels.

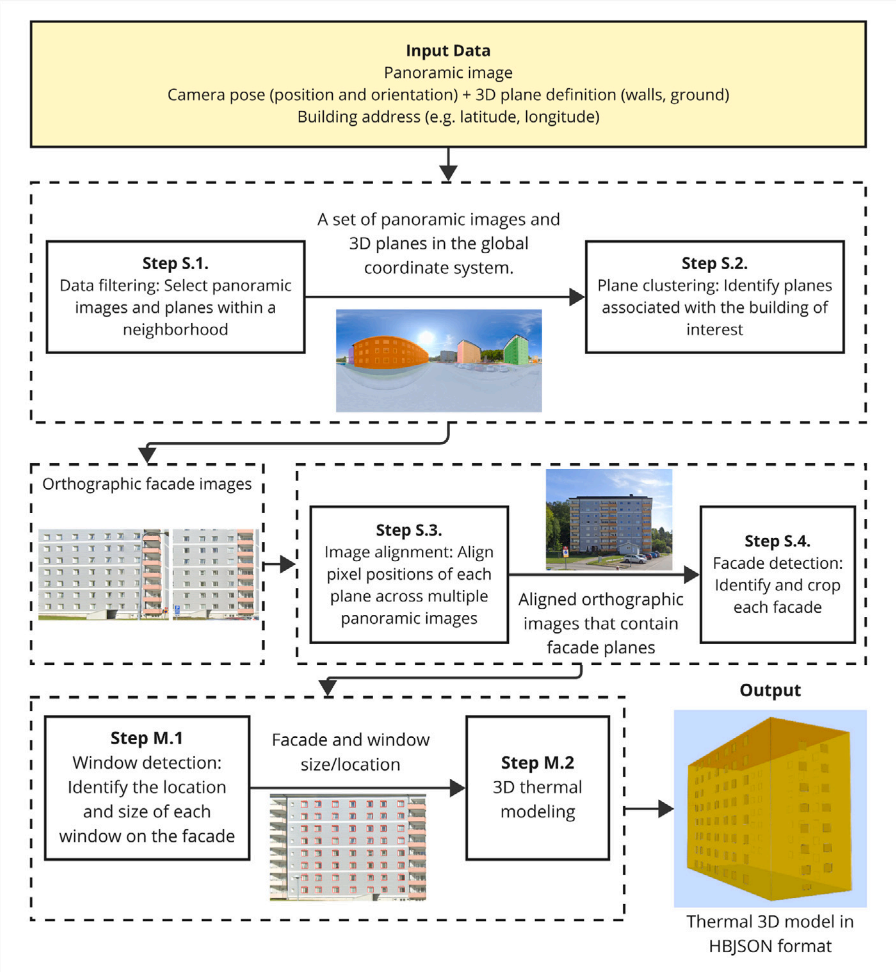
Input Data
Panoramic image
Camera pose (position and orientation) + 3D plane definition (walls, ground)
Building address (e.g. latitude, longitude)
Step S.1.
Data filtering: Select panoramic images and planes within a neighborhood
A set of panoramic images and 3D planes in the global coordinate system.
Step S.2.
Plane clustering: Identify planes associated with the building of interest
Orthographic facade images
Step S.3.
Image alignment: Align pixel positions of each plane across multiple panoramic images
Aligned orthographic images that contain facade planes
Step S.4.
Facade detection: Identify and crop each facade
Step M.1
Window detection: Identify the location and size of each window on the facade
Facade and window size/location
Step M.2
3D thermal modeling
Output
Thermal 3D model in HBJSON format

Fig. 4. Workflow for the StreetView path.

Fig. 5. Example of a 16384 × 8192 panoramic image captured at a given location.

This metric combines the cosine distance between normals $(1 - \mathbf{n}_i \cdot \mathbf{n}_j)$, weighting orientation similarity highly ($w_n = 1.0$), with the weighted absolute difference in origin distances $|d_i - d_j|$ ($w_d = 0.01$). A strict distance threshold of 1×10^{-5} is used for clustering.

Following clustering, orthographic views are generated from each contributing Street View panorama in a cluster. A common 3D plane and coordinate system are established for the cluster. This process is summarized in Algorithm 3.

An orthographic image grid is defined on this plane at a desired real-world resolution (e.g., pixels per meter). For each grid pixel, the corresponding 3D point on the common plane is projected back into the source panoramic image using its known pose and geometry to determine the source pixel coordinates. The color is then sampled from the panorama and assigned to the orthographic grid pixel. This is the orthographic transformation step, described in Algorithm 1.

This per-panorama process yields multiple, consistently scaled orthographic views of the same facade area. These views effectively correct the perspective distortion of the original panoramas, ensuring parallel lines remain parallel, preserving scale, and reducing parallax errors. They serve as the input for the image alignment in Step S.3.

Step S.3 image alignment: Align pixel positions of each plane across multiple panoramic images.

Challenges such as incorrect plane definitions, camera poses, and misalignments can arise. Therefore, we would like to ensemble results from different orthographic images to enhance the robustness of the subsequent semantic analysis. To achieve this, SIFT key points detection [78] and image registration [19] are employed between each pair of panoramic images. It is worth noting that this process, aimed at aligning facades and windows, can be time-consuming due to its pairwise complexity. The outcome of this step is a collection of aligned images as shown in Fig. 10.

Step S.4 facade detection: Detect and crop each facade.

The next objective is to extract the facades of interest from these orthographic images (cf. 10). In theory, these facades can be extracted in the image based on the plane definition. However, in reality, the plane definition is often not well aligned or complete when provided at scale. This causes issues such as planes do not cover the full facade, incorrect facade size, etc. Therefore, we extract the facade algorithmically with a bounding box.

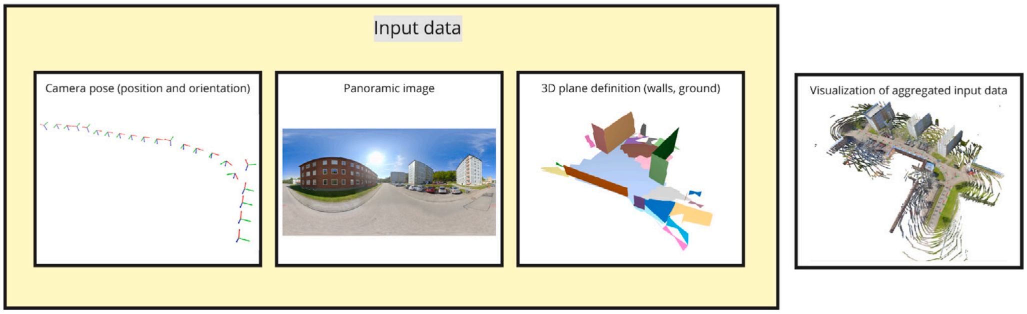
Input data
Camera pose (position and orientation)
Panoramic image
3D plane definition (walls, ground)
Visualization of aggregated input data

Fig. 6. Input data for StreetView.

Algorithm 1 StreetView-based Orthographic Projection

```
Input: Panoramic image $\mathbf{I}_{pano}$, Plane equation $\mathrm{Eq}_{plane}$ $(\mathbf{n}_{plane}, d)$, 3D points $\mathcal{P}_{plane3D}$, Pixel size $s_{pixel}$, Panorama pose $\mathbf{T}_{pano}$ (implicit)
Output: Orthographic image $\mathbf{I}_{ortho}$
Stage 1: Define Plane Geometry & Coordinate System
$\mathbf{n}_{plane} \leftarrow \mathrm{ExtractNormal}(\mathrm{Eq}_{plane})$
$\mathbf{u}_{plane}, \mathbf{v}_{plane} \leftarrow \mathrm{CalculatePlaneBasisVectors}(\mathbf{n}_{plane})$
$\mathbf{R}_{w2p} \leftarrow \mathrm{RotationMatrixFromBasis}(\mathbf{u}_{plane}, \mathbf{v}_{plane}, \mathbf{n}_{plane})$
Stage 2: Determine Extent and Output Grid
$\mathcal{P}_{plane2D} \leftarrow \mathrm{TransformToPlaneCoords}(\mathcal{P}_{plane3D}, \mathbf{R}_{w2p}, \mathrm{Eq}_{plane})$
$\mathbf{p}_{min2D}, \mathbf{p}_{max2D} \leftarrow \mathrm{CalculateExtent}(\mathcal{P}_{plane2D})$
$W_{out} \leftarrow \mathrm{Round}((\mathbf{p}_{max2D}.x - \mathbf{p}_{min2D}.x)/s_{pixel})$
$H_{out} \leftarrow \mathrm{Round}((\mathbf{p}_{max2D}.y - \mathbf{p}_{min2D}.y)/s_{pixel})$
$\mathbf{I}_{ortho} \leftarrow \mathrm{CreateImageBuffer}(W_{out}, H_{out})$
$\mathbf{p}_{gridOrigin2D} \leftarrow \mathbf{p}_{min2D}$
Stage 3: Generate Orthographic Image Pixels
for $y \leftarrow 0$ to $H_{out} - 1$ do
    for $x \leftarrow 0$ to $W_{out} - 1$ do
        $\mathbf{p}_{plane2D} \leftarrow \mathbf{p}_{gridOrigin2D} + \mathrm{Vector2D}(x \cdot s_{pixel}, y \cdot s_{pixel})$
        $\mathbf{R}_{p2w} \leftarrow \mathrm{Inverse}(\mathbf{R}_{w2p})$
        $\mathbf{p}_{world} \leftarrow \mathrm{TransformToWorldCoords}(\mathbf{p}_{plane2D}, \mathbf{R}_{p2w}, \mathrm{Eq}_{plane})$
                                        ▷ Core step: Reproject 3D point to 2D panorama
        $u_{pano}, v_{pano} \leftarrow \mathrm{ProjectWorldToPano}(\mathbf{p}_{world}, \mathbf{T}_{pano}, \mathbf{I}_{pano}.\mathrm{shape})$
        if IsValidCoord($u_{pano}, v_{pano}, \mathbf{I}_{pano}$.shape) then
            $C_{pixel} \leftarrow \mathrm{SampleColor}(\mathbf{I}_{pano}, u_{pano}, v_{pano})$
            $\mathbf{I}_{ortho}[y, x] \leftarrow C_{pixel}$
        else
            $\mathbf{I}_{ortho}[y, x] \leftarrow \mathrm{BackgroundColor}$
return $\mathbf{I}_{ortho}$
```

Algorithm 2 StreetView Geometric Plane Clustering

```
Input: Dictionary $\mathcal{P}$ containing panoramic data (incl. local planes and world transform $\mathbf{T}$), Clustering distance threshold $\delta_{cluster}$ $(= 1 \times 10^{-5})$
Output: Dictionary $C_{geom}$ mapping cluster IDs to lists of $(pano\_id, plane\_idx)$ tuples for geometrically similar plane segments.
Stage 1: Transform Candidate Planes to World Coordinates
$\mathcal{P}_{world} \leftarrow$ empty list          ▷ List to store world-frame plane equations $p_{world} = (\mathbf{n}, d)$
$\mathcal{L}_{idx} \leftarrow$ empty list                    ▷ List to store corresponding $(pano\_id, plane\_idx)$
for $pano\_id \in \mathrm{Keys}(\mathcal{P})$ do
    for $plane\_idx \leftarrow 0$ to $\mathrm{NumPlanes}(\mathcal{P}[pano\_id]) - 1$ do
        $p_{local} \leftarrow \mathcal{P}[pano\_id].\mathrm{depth.planes}[plane\_idx]$
        if IsHorizontal($p_{local}$) or IsZero($p_{local}$) then          ▷ Filter non-facade planes
            continue
        $\mathbf{T} \leftarrow \mathcal{P}[pano\_id].T$                          ▷ Get panorama's transform to world
        $p_{world} \leftarrow \mathrm{TransformPlane}(p_{local}, \mathbf{T})$     ▷ Calculate plane equation in world frame
        $\mathcal{P}_{world}.\mathrm{append}(p_{world})$
        $\mathcal{L}_{idx}.\mathrm{append}((pano\_id, plane\_idx))$
Stage 2: Cluster Transformed Planes by Geometric Similarity
                              ▷ Calculate pairwise distances using metric from Eq. (1)
$\mathbf{D} \leftarrow \mathrm{CalculatePairwiseDistances}(\mathcal{P}_{world}, \mathrm{CustomMetric})$
                                                  ▷ Apply Agglomerative Clustering
$l_{geom} \leftarrow \mathrm{AgglomerativeClustering}(\mathbf{D}, \mathrm{linkage}=\text{'average'}, \mathrm{threshold} = \delta_{cluster})$
Stage 3: Collect Cluster Members
$C_{geom} \leftarrow$ empty dictionary
for $label \in \mathrm{UniqueLabels}(l_{geom})$ do
    if $label = -1$ then                                    ▷ Skip potential noise label
        continue
    $idx_{members} \leftarrow \mathrm{IndicesWhere}(l_{geom} == label)$
    if $\mathrm{Length}(idx_{members}) \geq 1$ then               ▷ Keep clusters with at least one member
        $\mathcal{M} \leftarrow [\mathcal{L}_{idx}[i]$ for $i \in idx_{members}]$              ▷ Get list of $(pano\_id, plane\_idx)$
        $C_{geom}[label] \leftarrow \mathcal{M}$
                  ▷ Output $C_{geom}$ identifies groups of geometrically similar plane segments.
return $C_{geom}$
```

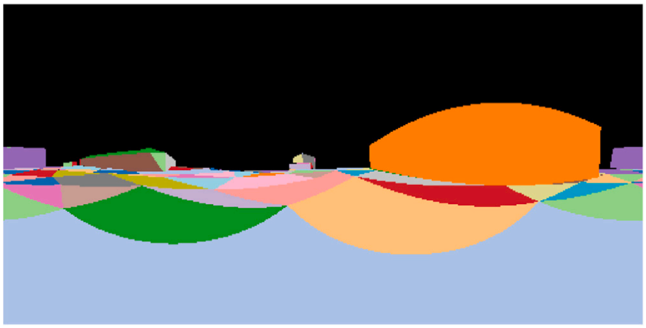

Fig. 7. A 512 × 256 matrix showing the plane index (plane_{id}) associated with the pixel values in the panoramic image.

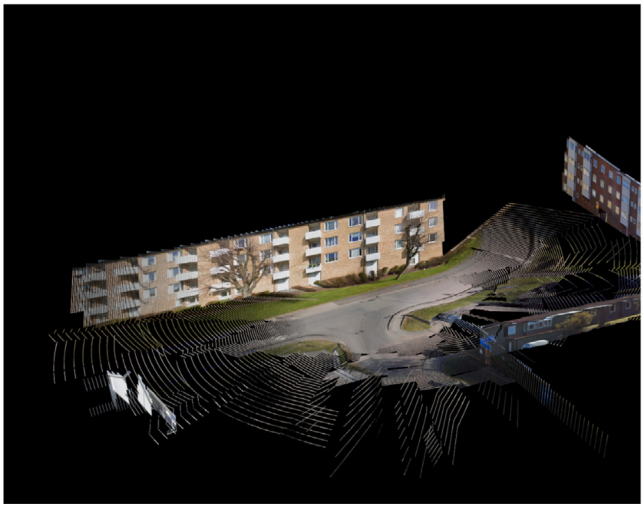

Fig. 8. Point cloud generated by projecting the panoramic image pixel values onto their corresponding 3D location.

We start by detecting lines and identify the vertical and horizontal ones by estimating their angles with a predefined tolerance of 10 degrees. Among these lines, those that appear consistently across multiple images are considered reliable and categorized as *relevant lines*, whereas all others are considered *irrelevant lines*. For each orthographic image, a RANSAC-inspired method [79] proposes candidate facade boundaries using lines from the reliable set (derived from consistent lines across views) and scores these boundaries based on how well all detected lines in the current image fit, implicitly favoring structurally sound alignments found within the reliable lines. The algorithm is described in Algorithm 4. Following this identification, we crop the facade from the image.

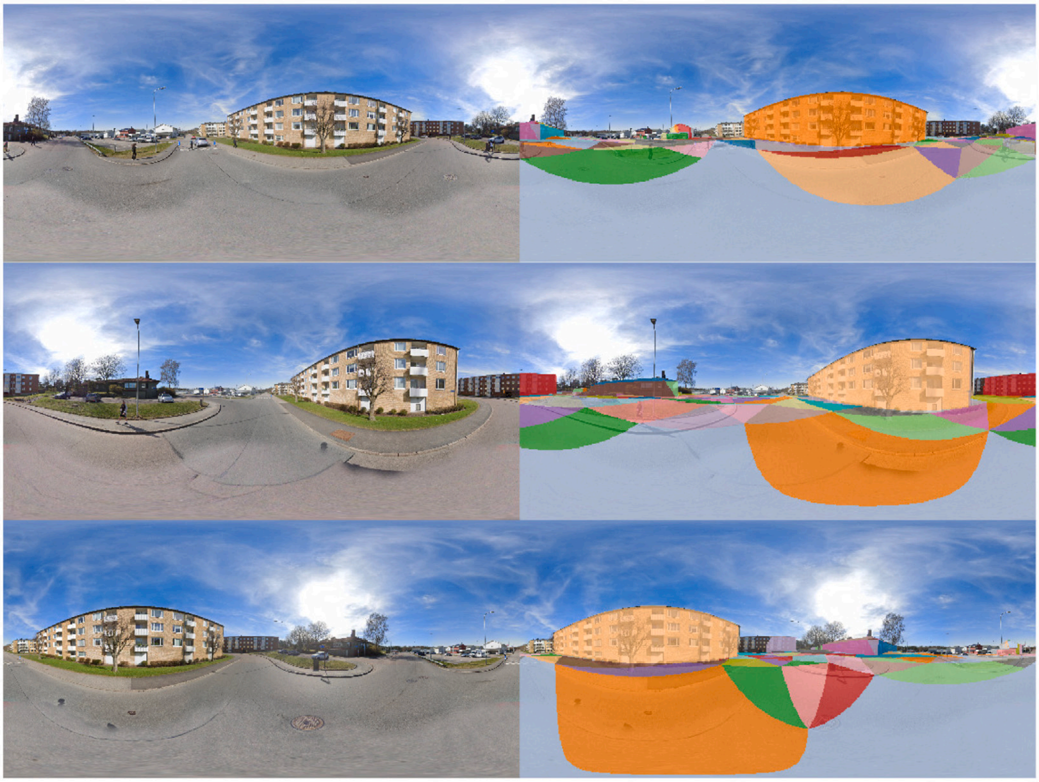

Fig. 9. Multiple, nearly potentially overlapping panoramic images along with their plane association matrices.

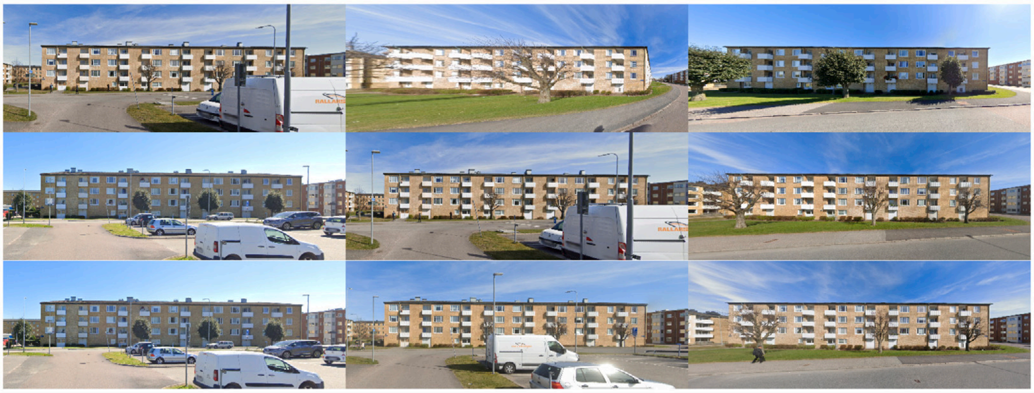

Fig. 10. Aligning orthographic images for improving robustness. We employ keypoint detection and alignment techniques to align the orthographic images. This alignment ensures that identical pixel locations across different images correspond to the same physical point in 3D space.

Note that the effectiveness of this process relies on the presence of dominant horizontal and vertical lines on the facade, which allows the scoring mechanism to identify correct boundaries, though this characteristic might be less prevalent on irregular buildings.

Facade detection can be achieved by deep learning techniques — we can opt for either Segment Anything [68] to segment any part of the image, deep neural networks trained for architectural features segmentation [33] or object detection networks [80]. Given the complex scenarios encountered in large-scale datasets, such as obstructive trees or viewing angles that capture the side of a building, segmentation methods are not sufficient when precision is required not just at the pixel level but for comprehensively capturing entire objects, which is essential for subsequent 3D modeling. Bounding box detection, on the other hand, aligns with our needs. However, given the architectural diversity of buildings, a tailored dataset might be necessary for optimal results for each geographic locations. The geometry-based iterative RANSAC method proves to be sufficiently effective in managing such challenges.

Limitations The proposed path offers a systematic approach to achieve robust 3D semantic analysis for thermal modeling at scale. However, large-scale data collection poses inevitable challenges. In areas where panoramic images are sparse, missing, or outdated, the accuracy of the 3D reconstruction may be compromised. Additionally, the process of aligning images and detecting facades can be hindered by environmental factors such as obstructive vegetation, poor lighting conditions, and suboptimal angles of capture. These factors can lead to inaccuracies in the plane definitions and alignments, which may affect the final 3D models and semantic analyses. Moreover, our facade detection algorithm assumes that lines on the facade are either vertical or horizontal — its effectiveness can vary based on the architectural diversity of the buildings being analyzed. Lastly, collecting and storing data at a large scale poses its unique challenges. To manage and make extensive

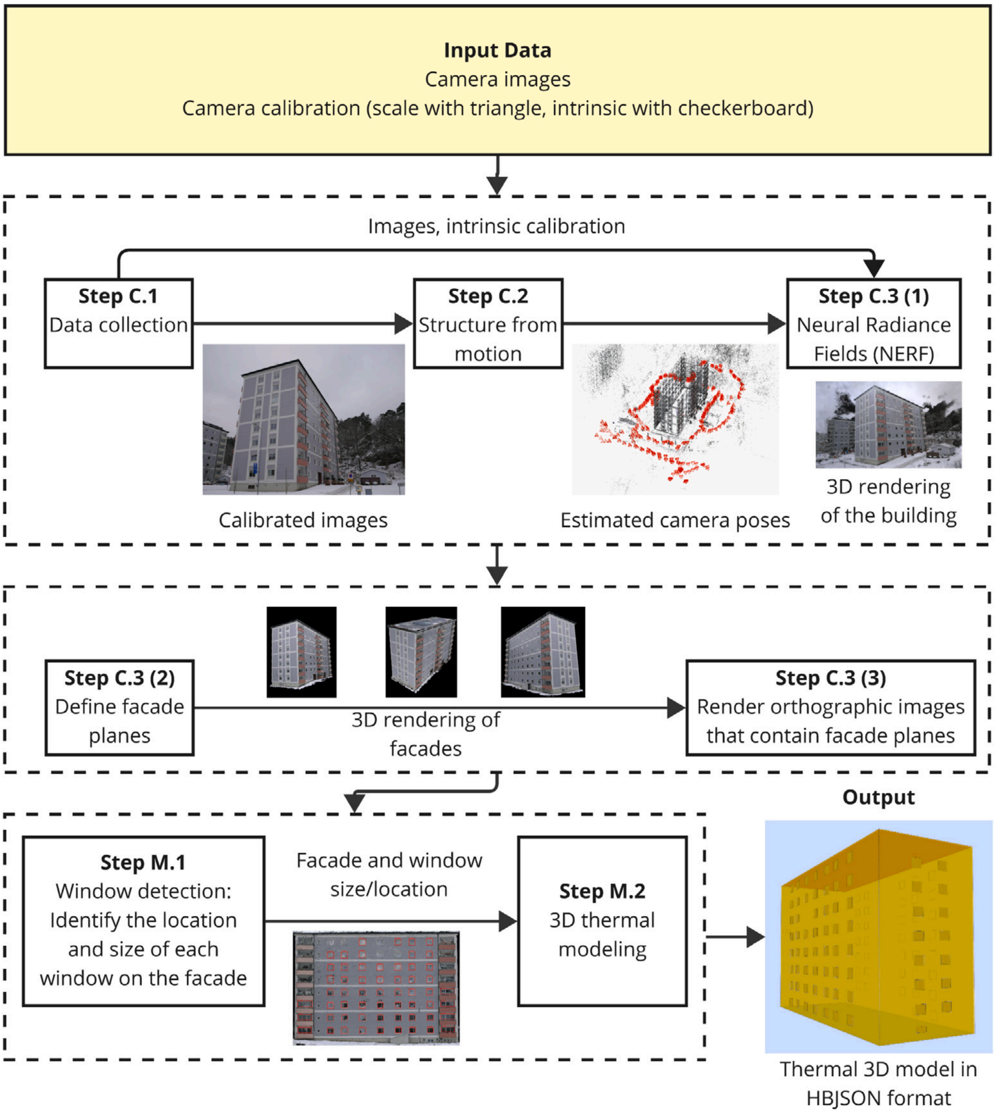
Input Data
Camera images
Camera calibration (scale with triangle, intrinsic with checkerboard)
Images, intrinsic calibration
Step C.1
Data collection
Step C.2
Structure from motion
Step C.3 (1)
Neural Radiance Fields (NERF)
Calibrated images
Estimated camera poses
3D rendering of the building
Step C.3 (2)
Define facade planes
3D rendering of facades
Step C.3 (3)
Render orthographic images that contain facade planes
Output
Step M.1
Window detection: Identify the location and size of each window on the facade
Facade and window size/location
Step M.2
3D thermal modeling
Thermal 3D model in HBJSON format

Fig. 11. Workflow for the Camera2D pipeline.

Algorithm 3 StreetView Per-Segment Orthographic Image Generation

```
Input: Clustered plane segments $C_{geom}$ (from Algorithm 2), Panorama data $\mathcal{P}$, Pixel size $s_{pixel}$
Output: Dictionary $\mathcal{I}_{ortho}$ mapping each $(pano\_id, plane\_idx)$ to its orthographic image $\mathbf{I}_{ortho\_seg}$.
$\mathcal{I}_{ortho} \leftarrow$ empty dictionary
for $clusterID \in$ Keys($\mathcal{C}_{geom}$) do
    $\mathcal{M} \leftarrow \mathcal{C}_{geom}[clusterID]$        ▷ List of members $(pano\_id, plane\_idx)$
    if Length($\mathcal{M}$) == 0 then continue
    // Generate orthographic image for each segment in the cluster
    for $(pano\_id, plane\_idx) \in \mathcal{M}$ do
        $\mathbf{I}_{pano} \leftarrow$ LoadPanoImage($\mathcal{P}[pano\_id]$)
        $p_{local} \leftarrow \mathcal{P}[pano\_id]$.depth.planes$[plane\_idx]$        ▷ Segment's local plane equation
        $\mathcal{P}_{seg3D} \leftarrow$ ExtractPointsForSegment($\mathcal{P}[pano\_id], plane\_idx$)   ▷ Segment's 3D points
        $\mathbf{T}_{pano} \leftarrow \mathcal{P}[pano\_id].T$        ▷ Segment's panorama pose
        if $\mathcal{P}_{seg3D}$ is empty then continue
        // Generate ortho view using segment's data and panorama
                                        ▷ Applies the principle from Algorithm 1
        $\mathbf{I}_{ortho\_seg} \leftarrow$ GenerateSingleOrthoStreetView($\mathbf{I}_{pano}, p_{local}, \mathbf{T}_{pano}, \mathcal{P}_{seg3D}, s_{pixel}$)
        $\mathcal{I}_{ortho}[(pano\_id, plane\_idx)] \leftarrow \mathbf{I}_{ortho\_seg}$
                ▷ Output $\mathcal{I}_{ortho}$ contains individual segment views, ready for alignment.
return $\mathcal{I}_{ortho}$
```

datasets like GSV searchable, it is standard practice to parametrize the data and store only the parameters. For example, 3D objects (originally collected as LiDAR point clouds) in GSV are often parameterized and stored as flat planes. This method greatly enhances data management efficiency but introduces certain limitations to our modeling capabilities. One significant artifact of this simplification is the occurrence of parallax errors. Parallax arises when objects are viewed from different angles, leading to apparent shifts in their positions relative to each other. In the context of facade modeling, using flat planes means that these positional shifts cannot be accurately captured, especially at larger viewing angles, where the effect is more significant. This simplification can distort the spatial relationships and dimensions in our models and affects the accuracy of the outputs.

Instead of depending on specific data provider such as GSV, alternative data collection methods can be considered. One approach is the use of mobile mapping systems (MMS), or unmanned aerial vehicles (UAVs), or drones equipped with location sensors, image cameras, and LiDAR. These equipment can capture high-resolution images from multiple angles and elevations as an alternative or complementary data collection method for the StreetView path. Note that when collecting in-house data, post-processing and parameterization steps need to be carried out.

Algorithm 4 RANSAC-Inspired Facade Detection Algorithm

```
Input: A set of $N$ aligned orthographic images $\mathcal{I} = \{I_1, \cdots, I_N\}$
Output: Optimal consensus bounding box $b^*$ and Cropped facades $\mathcal{F} = \{F_1, \cdots, F_N\}$
Stage 1: Detect Lines and Create Proposal Set
Detect lines $\mathcal{L}_k$ in each image $I_k \in \mathcal{I}$ using the Line Segment Detector (LSD). Keep track of all detected lines $\{\mathcal{L}_k\}_{k=1}^{N}$.
Process all detected lines by filtering and merging them (both within each image and across images) to create a reliable line proposal set $\mathcal{L}_r$.    ▷ This set represents line segments consistently identified across multiple views.
Stage 2: RANSAC to Find Candidate Box per Image
$\mathcal{B}_{candidates} \leftarrow$ empty list    ▷ Store best box found for each image
for $k \leftarrow 1$ to $N$ do    ▷ Run RANSAC independently for each image $I_k$
    Initialize best score $S_{best\_k} \leftarrow -\infty$ and best box $b_k^* \leftarrow$ null for image $I_k$.
    Let $NumIterations$ be the number of RANSAC trials.    ▷ Define RANSAC parameter
    for $i \leftarrow 1$ to $NumIterations$ do    ▷ Inner RANSAC loop
        Propose a candidate bounding box $b_{i,k}$ using two randomly selected lines from the proposal set $\mathcal{L}_r$.
        Calculate a score $S(b_{i,k})$ for the candidate box based on how well lines detected within the current image $I_k$ ($\mathcal{L}_k$) fit within $b_{i,k}$.    ▷ Lines fully inside increase score, lines crossing decrease score.
        if $S(b_{i,k}) > S_{best\_k}$ then
            Update $S_{best\_k} \leftarrow S(b_{i,k})$ and $b_k^* \leftarrow b_{i,k}$.
    $\mathcal{B}_{candidates}$.append($b_k^*$)    ▷ Store the best box found for image $I_k$
Stage 3: Determine Final Consensus Bounding Box
Analyze the geometric properties (corner coordinates) of all candidate boxes stored in $\mathcal{B}_{candidates}$.
Determine the final consensus bounding box $b^*$ by clustering the candidate corners (using DBSCAN) and selecting the median coordinates of the dominant cluster.
Stage 4: Crop Facades
$\mathcal{F} \leftarrow$ empty list    ▷ Initialize list for cropped facade images
for $k \leftarrow 1$ to $N$ do
    $F_k \leftarrow$ CropImage($I_k, b^*$)    ▷ Crop image $I_k$ using the final consensus box $b^*$
    $\mathcal{F}$.add($F_k$)
return $b^*$, $\mathcal{F}$
```

3.2. *Camera2D (C) for targeted inspection*

Despite the enhanced robustness and scalability of the StreetView path, challenges persist due to the broad approach to data collection. For detailed examinations of a particular building of interest, the StreetView path might prove inadequate due to occlusion and missing facades. To cover such use cases, we developed a second alternative path that targets the estimation of specific buildings to ensure a dense and complete data collection. This methodology reconstructs 3D information via Neural Radiance Fields (NeRF), SfM, and real-world local reference points, aiming to mitigate common photogrammetric distortions for accurate digital twin generation. By analyzing multiple photographs of a scene from different viewpoints, such techniques identify common keypoints and matches between images, which are then used to triangulate the 3D position of these points, forming a sparse 3D cloud. Following this, it can perform dense reconstruction to get a more detailed 3D model. Note that in addition to camera position, accurate camera orientation information can enhance the accuracy of the outcome, otherwise camera orientation can be estimated from the image data. An overview can be found in Fig. 11. There are three main steps (C.1-C.3) described as follows.

Step C.1: Data collection protocol Data collection for the Camera2D path involved capturing a dense set of photographs (typically 300 to 500 images) for each building, focusing on achieving sufficient facade coverage from multiple angles, aiming for 20% overlap between sequential images, and incorporating loop closure where practical to enable robust 3D reconstruction via SfM and NeRF. The input data is illustrated in Fig. 12. Specifically, in our image acquisition protocol, we collect high-resolution still photographs due to their immunity to the rolling shutter effect and motion blur – artifacts commonly associated with video capture [81,82]. These artifacts can significantly degrade the quality of data used in SfM processes, leading to less accurate 3D models. In particular, we capture a dense array of photographs from multiple angles, which is critical for creating an overlapping dataset that enables a robust input with adequate redundancy for the SfM analysis. The redundancy not only enhances the precision of the resulting 3D model but also provides a safeguard against potential data loss or corruption.

Further, the photographs are taken in a way that they envelop the entire object, capturing every aspect and detail necessary for a complete reconstruction. The shooting angles and positions are guided by both the geometrical considerations of NeRF and the practical constraints of on-site conditions.

Moreover, as described in Section 2.1.2, images are taken from three local coordinate reference points (a real-world triangle on the ground) in proximity to the object. The sides of the triangle are measured with an 8-meter measuring tape. This triangle acts as a reference to anchor the scale for the entire sequence of the reconstruction process. While an 8-meter tape is sufficient for our current requirements, alternative measurement tools, such as laser distance meters, could be used to potentially enhance the signal-to-noise ratio and measurement precision.

Step C.2: Camera pose estimation using COLMAP After image acquisition, we apply COLMAP [83] to reconstruct the 3D structure of the building from overlapping images and determining the intrinsic and extrinsic camera parameters. COLMAP is a commonly used photogrammetry tool that automates 3D reconstruction from unordered image sets by integrating SfM and MVS techniques. It performs camera calibration, image matching, 3D model generation, and texture mapping. The process begins with detecting keypoints and extracting descriptors for each image, followed by matching features across images using appearance-based methods combined with geometric verification. Verified matches are used to construct a scene graph, where nodes represent images and edges represent shared keypoints. COLMAP then initiates an incremental reconstruction, starting from an image pair with sufficient matches and geometric diversity, progressively registering new images and triangulating additional points. Finally, a global bundle adjustment is performed to optimize camera poses and 3D point coordinates by minimizing the overall reprojection error. Through this pipeline, COLMAP produces detailed and accurate 3D models from large, unordered image collections. Throughout this procedure, the intrinsic camera model parameters, which define the projection characteristics of the camera, are refined alongside the reconstruction. In addition, during the SfM process, the camera poses corresponding to the triangle's corners are identified. The known lengths of the triangle's sides are then applied to triangulate and determine the relative positions of these poses in metric units. This step translates the relative scale derived from the SfM process into an absolute scale applicable to the real world, establishing a consistent and accurate scale across our SfM reconstruction.

Step C.3: NeRF modeling and orthographic projection To generate orthographic views from the inherently perspective NeRF model (Instant-NGP), we first define the target facade plane using corner vertices (e.g., $\mathbf{v}_0, \mathbf{v}_1, \mathbf{v}_2$) of the relevant bounding box face identified within the scaled 3D reconstruction (derived from Step C.2). These vertices define the plane's origin $\mathbf{v}_0$ and its basis vectors within the plane (e.g., $\mathbf{u} = \mathbf{v}_1 - \mathbf{v}_0$, $\mathbf{v} = \mathbf{v}_2 - \mathbf{v}_0$). The orthographic image is then generated by densely sampling the NeRF volume. For each pixel (x, y) in the target orthographic image (of pixel dimensions $W \times H$), the corresponding 3D point $\mathbf{p}(x, y)$ on the facade plane is calculated:

$$\mathbf{p}(x, y) = \mathbf{v}_0 + \frac{x}{W}\mathbf{u} + \frac{y}{H}\mathbf{v} \tag{2}$$

The NeRF model is then queried to render the color and depth information along a **single ray originating at $\mathbf{p}(x, y)$ and directed perpendicularly outwards** from the facade plane. Assembling these rendered samples pixel-by-pixel simulates parallel projection, creating the orthographic image. Key configuration parameters included setting the axis-aligned bounding box scale (to 4, in our case). Training proceeded until convergence based on visual inspection of rendering quality and stabilization of the training loss. The algorithm is described in Algorithm 5. This method directly renders the full extent of the defined facade plane at the desired real-world scale determined in

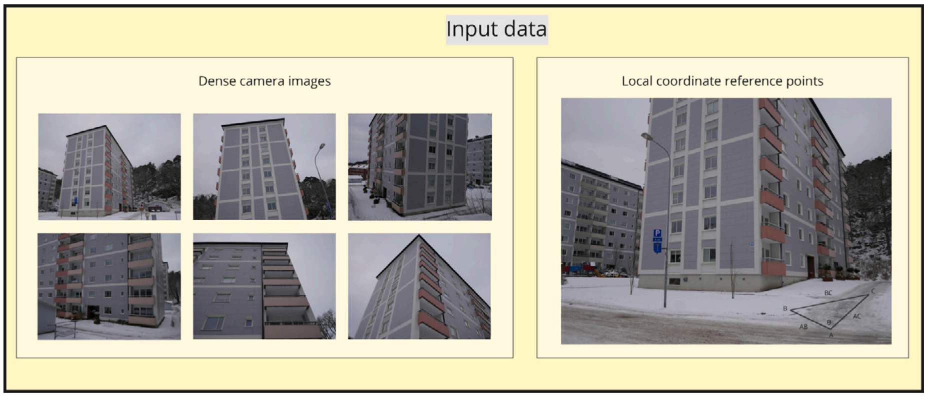

Fig. 12. Input data for Camera2D.

Algorithm 5 NeRF-based Orthographic Projection

```
Input: Trained NeRF model $\mathcal{M}_{NeRF}$, Facade corners $\mathcal{P}_{corners}$, Pixel size $s_{pixel}$, Samples per pixel $N_{spp}$
Output: Orthographic image $\mathbf{I}_{ortho}$
Stage 1: Define Plane Geometry & Coordinate System
$\mathbf{p}_{origin} \leftarrow \mathcal{P}_{corners}[0]$
$\mathbf{u}, \mathbf{v}, \mathbf{w} \leftarrow$ CalculatePlaneBasis($\mathcal{P}_{corners}$)        ▷ $\mathbf{w}$ is normal
$\mathbf{R}_{ortho} \leftarrow$ RotationMatrixFromBasis($\mathbf{u}, \mathbf{v}, \mathbf{w}$)
Stage 2: Determine Output Resolution
$W_{facade} \leftarrow$ Distance($\mathcal{P}_{corners}[1], \mathbf{p}_{origin}$)
$H_{facade} \leftarrow$ Distance($\mathcal{P}_{corners}[2], \mathbf{p}_{origin}$)
$W_{out} \leftarrow$ Round($W_{facade}/s_{pixel}$)
$H_{out} \leftarrow$ Round($H_{facade}/s_{pixel}$)
$\mathbf{I}_{ortho} \leftarrow$ CreateImageBuffer($W_{out}, H_{out}$)
Stage 3: Generate Orthographic Image Pixels
for $y \leftarrow 0$ to $H_{out} - 1$ do
    for $x \leftarrow 0$ to $W_{out} - 1$ do
        $\mathbf{p}_{world} \leftarrow \mathbf{p}_{origin} + (y \cdot s_{pixel} \cdot \mathbf{v}) + (x \cdot s_{pixel} \cdot \mathbf{u})$
        $\mathbf{T}_{c2w} \leftarrow$ SetupOrthoCameraAtPoint($\mathbf{p}_{world}, \mathbf{R}_{ortho}$)
        SetNeRFCamera($\mathcal{M}_{NeRF}, \mathbf{T}_{c2w}$, FOV $\approx 0$)
                                ▷ Core step: Synthesize pixel color from 3D model
        $C_{pixel} \leftarrow$ RenderFromNeRF($\mathcal{M}_{NeRF}, N_{spp}$)
        $\mathbf{I}_{ortho}[y, x] \leftarrow C_{pixel}$
return $\mathbf{I}_{ortho}$
```

Step C.2. It renders the scene including any foreground occlusions reconstructed by NeRF but avoids occlusion issues associated with distant virtual cameras.

Limitations The ground-based camera angle limits the visibility of roof structures. Further, achieving a successful SfM reconstruction requires a significant number of images. Since images are typically taken from ground level, this can introduce perspective distortion in higher parts of buildings, making the rendering noisier in orthographic projections. An alternative might involve using drones for capturing images, which can directly provide camera poses and potentially 3D information through LiDAR, though at a higher cost.

Furthermore, while we render NeRF images as orthographic projections to standardize the image representation, we acknowledge that their underlying pixel distributions can still differ substantially from real-world images. This difference may pose challenges for object detection and classification, as models trained only on real-world data may not generalize effectively to NeRF-rendered inputs. We did not apply any domain adaptation or special training strategies to mitigate this gap in our paper. Addressing this limitation remains an important direction for future work.

3.3. Merged steps

The aforementioned steps for both alternative paths, StreetView and Camera2D, produce true-to-scale orthographic facades. To complete the 3D thermal modeling, two remaining merged steps are described as follows.

Step M.1: Semantic facade parsing Identify the location and size of each window on the facade. Once the facade is identified, a pretrained ResNet-50 RetinaNet, trained on the LSAA dataset [84], is used for window detection. More specifically, a ResNet-50 RetinaNet model as the base model, initialized with weights pre-trained on the COCO 2017 dataset, was fine-tuned for 64,000 iterations specifically for this task using the LSAA dataset (Zhu et al. 2020) of architectural facade elements. The training is conducted using an SGD optimizer with a learning rate of 0.0005 and data augmentation incorporating color jitter and random resizing/scale jittering (based on 1440px max resize, 800-1600px scale range). For inference, orthographic images were resized (maintaining aspect ratio, shortest side to 1024px, longest side max 1333px) and normalized using standard ImageNet statistics. If multiple images are available for the same facade, detections from individual views are fused using a specialized ensemble method (Algorithm 6) for enhanced robustness. Initially, only detections exceeding a confidence score of 0.2 are considered. Viewpoints are not explicitly weighted. This involves grouping these filtered detections from different views based on high spatial overlap (IoU > 0.3) to identify potential matches corresponding to the same physical window. To ensure reliability and consistency, providing robustness against partial occlusions and spurious detections, only groups representing windows detected across at least two different source images ($N \geq 2$) are considered valid. For each valid group, geometric information and confidence scores are merged to generate a single, representative detection. This merged detection must also meet a final score threshold τ_{score2}, set to 0.4, to be accepted. This algorithm is described in Algorithm 6.

This method enhances the reliability of the detection process by ensuring consistency across different images. Once windows are detected and their positions within the facade are determined, the dimensions in these elements in the image can be translated into real-world measurements by leveraging the respective scale information (i.e. plane definition for StreetView and local coordinate reference points for Camera2D; cf. Section 2.1.2). These outcomes collectively provide users with the essential data needed to reconstruct the facade in 3D.

Step M.2: 3D thermal modeling Once the windows are detected, their locations and scales can be used to render them in 3D. When combined with available footprint information, a 3D model of the complete building can be reconstructed using HoneybeeJSON,[2] a standardized JSON schema to encode geometric information about the building's envelope, including the coordinates of the facades and the location of windows. Once the geometric information is encoded into the HoneybeeJSON schema, thermal properties of the building, such as material properties and the u-value of the windows. The input of these properties could be automated using databases such as TABULA [7] or Energy Performance

[2] https://github.com/ladybug-tools/honeybee-schema

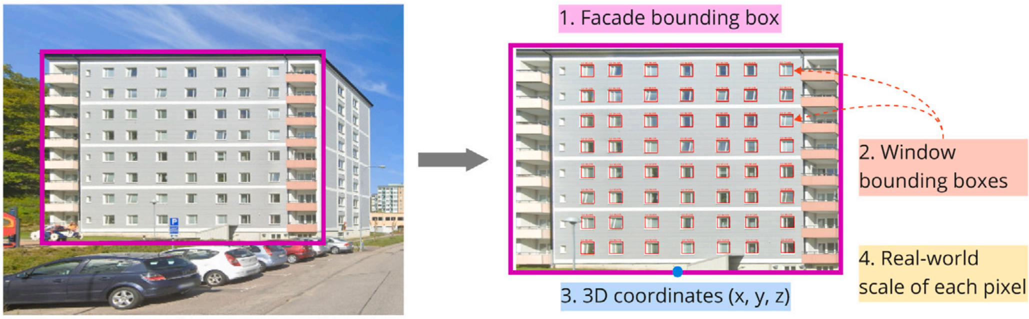
1. Facade bounding box
2. Window bounding boxes
3. 3D coordinates (x, y, z)
4. Real-world scale of each pixel

Fig. 13. Final unified outputs of the two alternative paths, Camera2D and StreetView, in the SI3FP pipeline.

Algorithm 6 StreetView Multi-View Detection Fusion

```
Input: Dictionary $\mathcal{D}_{raw}$ mapping source $k = (pano\_id, plane\_idx)$ to lists of raw detections
  $d = \{\text{bbox}, \text{score}, \text{category\_id}\}$, Confidence threshold $\tau_{conf}$, IoU threshold $\tau_{iou}$, Minimum
  detections per cluster $N_{min}$, Merged score threshold $\tau_{score2}$
Output: List $\mathcal{D}_{final}$ of merged and validated detections for the facade.
Stage 1: Filter Raw Detections by Confidence
$\mathcal{L}_{det} \leftarrow$ empty list            ▷ Store filtered detections meeting initial confidence
for $k \in \text{Keys}(\mathcal{D}_{raw})$ do            ▷ Iterate through each source view
    for $d \in \mathcal{D}_{raw}[k]$ do            ▷ Iterate through detections in the view
        if $d.\text{score} \geq \tau_{conf}$ then
            $\mathcal{L}_{det}.\text{append}(d)$            ▷ Keep detection if score is high enough
if $\text{Length}(\mathcal{L}_{det}) == 0$ then return empty list
                                    ▷ Exit if no detections pass initial filter
Stage 2: Group by Category and Cluster Spatially
$\mathcal{D}_{final} \leftarrow$ empty list            ▷ Initialize final output list
for $cat\_id \in \text{UniqueCategories}(\mathcal{L}_{det})$ do            ▷ Process one object category at a time
    $\mathcal{D}_{cat} \leftarrow \text{DetectionsInCategory}(\mathcal{L}_{det}, cat\_id)$            ▷ Get all filtered detections of this
  category
    if $\text{Length}(\mathcal{D}_{cat}) < 2$ then continue
                                    ▷ Need at least two detections to potentially form a cluster
    // Cluster detections based on bounding box overlap (IoU)
    $Clusters_{idx} \leftarrow \text{ClusterByHighIoU}(\mathcal{D}_{cat}, \tau_{iou})$ ▷ Groups indices of detections that likely
  represent the same object instance
Stage 3: Validate and Merge Spatial Clusters
    for $idx_{cluster} \in Clusters_{idx}$ do            ▷ Process each spatial cluster
        // — Validation 1: Check Minimum Detections in Cluster —
        if $\text{Length}(idx_{cluster}) < N_{min}$ then ▷ Check if cluster has enough total detections
            continue            ▷ Reject cluster: Too few detections grouped together
        // — Merge Cluster —
        $\mathcal{M}_{det} \leftarrow [\mathcal{D}_{cat}[i] \text{ for } i \in idx_{cluster}]$            ▷ Get the actual detection objects in this
  cluster
        $d_{final} \leftarrow \text{MergeDetectionsSpecific}(\mathcal{M}_{det})$ ▷ Compute representative bbox (median
  coords) and score (mean of sqrt scores)
        $d_{final}.\text{category\_id} \leftarrow cat\_id$            ▷ Assign the category ID
        // — Validation 2: Check Merged Score Threshold —
        if $d_{final}.\text{score} < \tau_{score2}$ then            ▷ Check if the merged detection is confident
  enough
            continue            ▷ Reject cluster: Merged score too low
        // — Store Validated and Merged Detection —
        $\mathcal{D}_{final}.\text{append}(d_{final})$            ▷ Keep the final detection for this cluster
return $\mathcal{D}_{final}$            ▷ Return the list of high-confidence, merged detections
```

Algorithm 7 SI3FP Pipeline

```
Input: Data Source Type ‘DataSourceType‘ (‘Camera2D‘ or ‘StreetView‘), ‘InputData‘,
  ‘BuildingFootprint‘, Target pixel size $s_{pixel}$, Detection thresholds $(\tau_{conf}, \tau_{iou}, N_{min}, \tau_{score2})$
Output: Thermal 3D model ‘ThermalModel‘ (HBJSON format)
if ‘DataSourceType‘ == ‘Camera2D‘ then
  — Camera2D Path —
    // C1: Data Collection
    $\mathcal{I}_{dense} \leftarrow \text{InputData}$            ▷ Assume dense image data collected
    // C2: Structure-from-Motion
    $\mathcal{T}_{cam}, \mathcal{P}_{corners} \leftarrow \text{EstimateScaledPosesAndFacadeCorners}(\mathcal{I}_{dense})$            ▷ Using COLMAP
    // C3a: NeRF Training
    $\mathcal{M}_{NeRF} \leftarrow \text{TrainNeuralRadianceField}(\mathcal{I}_{dense}, \mathcal{T}_{cam})$            ▷ Using Instant-NGP
    // C3b: Orthographic Projection
    $\mathbf{I}_{ortho} \leftarrow \text{GenerateOrthoNeRF}(\mathcal{M}_{NeRF}, \mathcal{P}_{corners}, s_{pixel})$            ▷ Using Algorithm 5
    $G_{facade} \leftarrow \text{GetFacadeGeometryFromCorners}(\mathcal{P}_{corners})$
    $\mathcal{F} \leftarrow \{\mathbf{I}_{ortho}\}$            ▷ Define Facade image set
else if ‘DataSourceType‘ == ‘StreetView‘ then
  — StreetView Path —
    // S1: Data Collection
    $\mathcal{P}_{anos} \leftarrow \text{InputData}$            ▷ Assume panorama data collected
    // S2: Plane Clustering
    $\mathcal{C}_{geom} \leftarrow \text{ClusterStreetViewPlanes}(\mathcal{P}_{anos})$            ▷ Using Algorithm 2
    // Generate Per-Pano Ortho Images
    $\mathcal{I}_{ortho\_segments} \leftarrow \text{GeneratePerPanoOrthographics}(\mathcal{C}_{geom}, \mathcal{P}_{anos}, s_{pixel})$   ▷ Algorithm 3;
  uses Algorithm 1 logic
    // S3: Image Alignment
    $\mathcal{I}_{aligned} \leftarrow \text{AlignOrthographicImages}(\mathcal{I}_{ortho\_segments})$            ▷ Using SIFT/Registration
    // S4: Facade Detection
    $b^*, \mathcal{F} \leftarrow \text{DetectFacadeBoundsAndCrop}(\mathcal{I}_{aligned})$            ▷ Using Algorithm 4 logic
    $G_{facade} \leftarrow \text{GetFacadeGeometryFromBbox}(b^*)$
  — Merged Steps —
// M1: Semantic Facade Parsing
$\mathcal{D}_{raw} \leftarrow \text{DetectWindowsInitial}(\mathcal{F})$            ▷ Apply detector (RetinaNet) to image(s) in $\mathcal{F}$
if ‘DataSourceType‘ == ‘StreetView‘ then
    $\mathcal{D}_{final} \leftarrow \text{FuseMultiViewDetections}(\mathcal{D}_{raw}, \tau_{conf}, \tau_{iou}, N_{min}, \tau_{score2})$ ▷ Using Algorithm
  6
else
    $\mathcal{D}_{final} \leftarrow \text{FilterSingleViewDetections}(\mathcal{D}_{raw}, \tau_{conf})$            ▷ Camera2D case
$G_{windows} \leftarrow \text{CalculateWindowGeometries3D}(\mathcal{D}_{final}, G_{facade}, s_{pixel})$
// M2: 3D Thermal Model Generation
$ThermalModel \leftarrow \text{AssembleHBJSONModel}(G_{facade}, G_{windows}, BuildingFootprint)$
return $ThermalModel$
```

Certificates, which is beyond the scope of this paper. The Honeybee model can be simulated in EnergyPlus using the built-in translation tools to provide the energy demand of the building.

The complete SI3FP pipeline The complete pipeline can be found in Algorithm 7.

Final output The final output is a thermal 3D model in HBJSON format. More specifically, for each facade, the outcome consists of four key components illustrated in Fig. 13.

1. Facade bounding box: Each path generates an orthographic image of the facade, with a bounding box describing its position and dimension within the image.
2. Window bounding boxes: Surrounding all windows within the facade, bounding boxes are detected to detail their positions and dimensions.
3. 3D coordinates of the facade center: The central point of the facade's bounding box is pinpointed in 3D space, specified by its latitude, longitude, and altitude.
4. Real-world scale of each pixel: The real-world scale of each pixel within the image is calculated from the scale to enable measurements from the image to actual dimensions.

4. Experiments and results

This section outlines the experimental setup for the evaluation of the alternative paths in the pipeline SI3FP.

4.1. Experimental setup

Data collection To test and validate our pipeline with real-world data, we collected ground truth from three buildings in Sweden. We use three

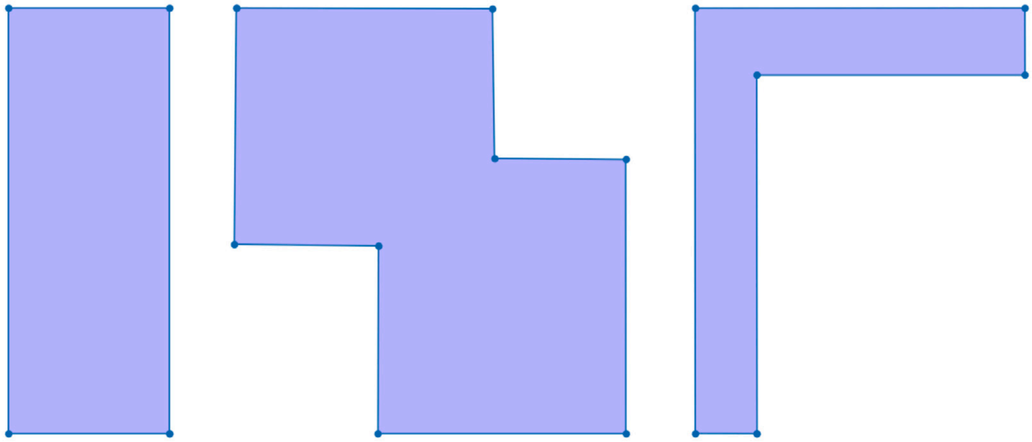

Fig. 14. Footprints of the buildings in our case study.

Table 2
Dataset overview for each building.

Building	Building information			Camera2D		StreetView		
	Height	Longest side	Facades	Images	Resolution (px)	Panoramas	Resolution (px)	Missing facades
B1	23.2	14.9	8	322	2296 × 1724	47	16384 × 8192	3 (37.5%)
B2	22.7	35.2	4	439	2250 × 1680	22	16384 × 8192	1 (25.0%)
B3	11.0	71.6	6	274	2288 × 1708	58	16384 × 8192	2 (33.3%)

multi-family residential buildings from 1961 to 1975, because buildings from this period urgently need energy renovation [85]. This era is known as the Million Homes Program — a national initiative to rapidly build one million dwellings to deal with the housing shortage [86]. Multi-family buildings represent 54% of the dwellings in the Swedish residential building stock [87]. These typologies are common not only in Sweden but all over Europe and represent 36% of the European residential building stock by floor area.[3] The footprints can be found in Fig. 14. For the StreetView path, approximately 15–40 relevant panoramic views per building facade cluster, obtained via the GSV API and filtered as described in Section 3.1. For the Camera2D path, the dense sets of 4 MP photographs captured for each building, with the detailed capture protocol described in Step C.1, Section 3.2. The information and the total number of images collected for each building is detailed in Table 2.

Evaluation In this paper, we focus on using the pipeline SI3FP for thermal 3D modeling, and hence we primary discuss around four evaluation aspects relevant to our application.

- Window detection: First of all, we are interested in evaluating the window detection rate. This is a critical intermediate step in facade parsing. A window prediction was considered a TP if its Intersection over Union (IoU) with the corresponding ground truth bounding box exceeded a threshold of 0.5, following the standard object detection practice [88]. To evaluate the performance of window detection, we use the F1-Score, which is the harmonic mean of Precision ($P = TP/(TP + FP)$) and Recall ($R = TP/(TP + FN)$).[4] Note: 'StreetView (Total)' considers all ground truth windows, while 'StreetView' without 'Total' only considers facades where the method produced predictions.
- Area of the windows and facades: The total area covered by windows is an important parameter in thermal modeling, as it directly influences the building's heat gain and loss. Accurate measurement of window areas enables calculations of thermal load and energy requirements. The area is evaluated by the Mean Absolute Relative Area Error (the average magnitude of the relative area error over matched window pairs ($A(w)$ is window area):

$$\text{Mean Abs Rel Area Err} = \frac{1}{|M|} \sum_{(w_p,w_{gt})\in M} \left| \frac{A(w_p) - A(w_{gt})}{A(w_{gt})} \right|$$

- Location of the windows: Spatially accurate window location helps in assessing natural light distribution within interiors, which affects both energy consumption for lighting and heating. The estimated location is evaluated by the Mean Absolute Position Error (m): (the average Euclidean distance between centers of matched window pairs, in meters):

$$\text{Mean Abs Pos Err (m)} = \frac{1}{|M|} \sum_{(w_p,w_{gt})\in M} D(C(w_p), C(w_{gt}))$$

Further, the location and area are jointly evaluated by the **Mean (IoU)**, which is defined as the average IoU over correctly matched window pairs $(w_p, w_{gt}) \in M$.

$$\text{Mean IoU} = \frac{1}{|M|} \sum_{(w_p,w_{gt})\in M} IoU(w_p, w_{gt})$$

- WWR (window-to-wall ratio): Based on an expert workshop with three representatives of large residential building portfolio owners in West Sweden, we established that a 5% error in WWR is accurate enough in the early planning phase. [38] confirmed this assumption and showed that experts have higher deviations in their estimations in this phase. The impact a 5% error in WWR has on the simulated energy demand depends on many factors. Assuming a WWR of 0.25, an increase of 5% would correspond to an increase of 0.75% in heating demand according to a study in Sweden [89], which is acceptable in this phase according to the expert workshop.
More specifically, we evaluate the WWR in the following manner:
  - WWR Error without missing facades (e.g. Camera2D and StreetView Standard): The mean of per-facade errors, $e(j) =$

[3] https://www.bpie.eu/wp-content/uploads/2015/10/HR_EU_B_under_microscope_study.pdf

[4] TP:True Positive; FP: False Positive; TN: True Negative; FN: False Negative.

Table 3
Evaluation results of SI3FP on three buildings.

Building	Method	F1-score	F1-score (No Balc)	Mean IoU	Mean Abs Rel Area Err (%)	Mean Abs Pos Err (m)	WWR error	WWR error (Imputed)
B1	Camera2D	0.75	0.90	0.80	17.1%	0.065	−0.080	–
B1	StreetView	0.75	0.97	0.70	15.2%	0.189	−0.090	–
B1	StreetView (Total)	0.71	0.65	–	–	–	−0.095	−0.070
B2	Camera2D	0.83	0.87	0.78	21.9%	0.077	−0.009	–
B2	StreetView	0.80	0.81	0.73	23.2%	0.105	−0.052	–
B2	StreetView (Total)	0.61	0.63	–	–	–	−0.092	−0.042
B3	Camera2D	0.72	0.84	0.77	15.2%	0.129	−0.048	–
B3	StreetView	0.28	0.46	0.69	22.3%	0.216	−0.040	–
B3	StreetView (Total)	0.19	0.04	–	–	–	−0.099	−0.048
Total	Camera2D	0.76	0.86	0.78	17.8%	0.095	−0.038	–
Total	StreetView	0.60	0.82	0.71	19.2%	0.165	−0.058	–
Total	StreetView (Total)	0.47	0.42	–	–	–	−0.095	−0.050

$W\hat{W}R(j) - WWR_{gt}(j)$, averaged over the set of facades F_{pred} where the method produced predictions. $W\hat{W}R(j)$ is the actual predicted WWR for facade j.

- WWR Error with missing facade (e.g. StreetView Total): The mean of per-facade errors, $e(j) = W\hat{W}R(j) - WWR_{gt}(j)$, averaged over *all* ground truth facades F_{gt}. If a facade j is missing, its estimated $W\hat{W}R(j)$ is treated as 0.0.
- Imputed WWR Error with missing facades (e.g. StreetView Total imputed): The mean of per-facade errors, $e(j) = W\hat{W}R_{imp}(j) - WWR_{gt}(j)$, averaged over *all* ground truth facades F_{gt}. If facade j lacks a StreetView prediction, its estimated WWR $W\hat{W}R_{imp}(j)$ is imputed using the average WWR calculated from facades that *did* have predictions ($\overline{WWR}_{pred}$).

Components like COLMAP and NeRF are applied or fitted to the data from these three buildings directly. The detection algorithm was fine-tuned on a separate, custom dataset of architectural elements (starting from COCO pre-trained weights). The three case study buildings were held out from this fine-tuning process, thus serving as an independent test set to evaluate the detector's generalization performance within the overall pipeline framework.

LiDAR scan as the ground truth We used the Topcon GLS-2000 scanner as the reference sensor to create the ground truth. The scanner offers a 360-degree scanning range. With a distance accuracy of 3.5 mm up to 150 meters and a horizontal and vertical angle accuracy of 6 arc-seconds, it produces dense point clouds, reaching up to 120,000 points per second. This collected data is stored as .pts file, a simple text file and loaded into the software Rhinoceros 3D as .xyz file. The geometry was manually modeled over the point cloud. In case of unclear areas, e.g. sparce points due to obstructions, architectural drawings and GSV images were referenced. Then the surface of the geometry was extracted and labeled with descriptions such as exterior wall, window, roof, etc., by using Grasshopper and the plugin Honeybee. The final model was exported as a JSON file and checked for correct interpretation in OpenStudio.

Computational infrastructure All data processing, model training (NeRF), and inference tasks reported in this paper were conducted on a workstation equipped with an AMD Ryzen 7 5800X3D CPU, 64 GB of RAM, and an NVIDIA RTX 3090 GPU with 24 GB of VRAM.

4.2. Results

The overall results are presented in Table 3, evaluated across three buildings using Camera2D and StreetView data sources. We discuss the evaluation metrics below.

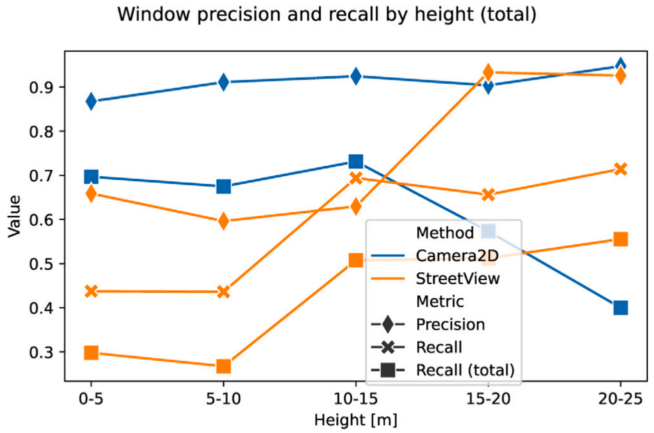

Fig. 15. Precision and recall of window detection versus the height of the building.

Window detection The detection result is summarized by the F1-Score in Table 3. Overall, Camera2D consistently achieves better performance than StreetView across almost all evaluation metrics. This difference is mainly due to the higher control over data acquisition in Camera2D: images are captured specifically for the task with careful coverage and appropriate resolution. In contrast, StreetView imagery is captured opportunistically, often from a distance or obstructed viewpoints, leading to missing facades and less optimal angles.

A clear trend emerges when comparing F1-Scores with and without balconies. Across all buildings, excluding facades with balconies significantly improves detection performance, particularly for StreetView imagery. Balconies introduce heavy occlusions, causing windows to be partially or entirely hidden. This makes reliable window detection extremely challenging, especially for automated methods trained on unobstructed data assumptions.

Building geometry influences performance for StreetView. B3, with its long facades, shows the worst results. Long facades require capturing windows across wide angles and distances in StreetView, which increases distortion and reduces the effective resolution for distant windows.

As noted in Section 3.2, the NeRF algorithm used in Camera2D requires high quality images to produce sharp renders. Cameras positioned at ground level and angled upwards often introduce significant perspective distortion. This results in blurring in both the reconstructed 3D renderings and the orthographic images, particularly impacting the detection of windows at higher elevations (decrease in recall) especially when the windows are small.

One potential solution to this source of error is to employ drones equipped with video cameras, which can capture imagery from elevated and varied angles, reducing perspective issues.

Fig. 15 illustrates how precision and recall vary with the height of the window position. Note that StreetView experiences an increase in

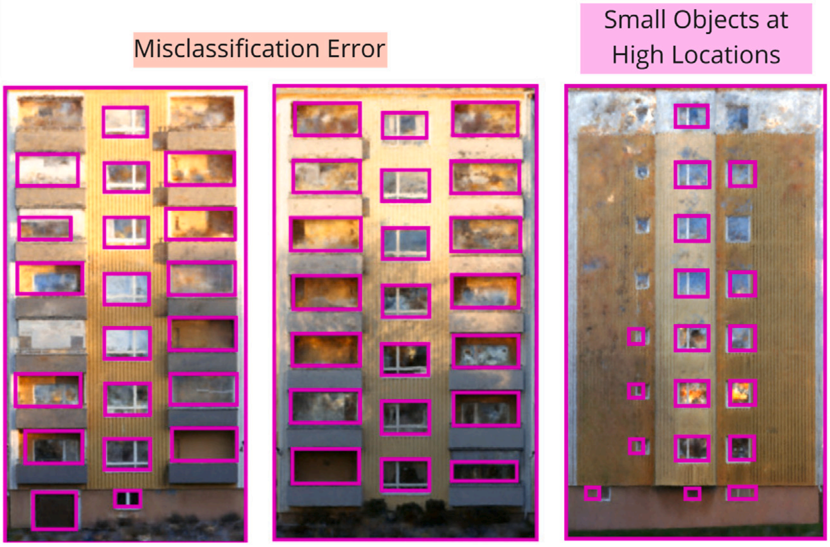

Fig. 16. Two types of errors in Camera2D.

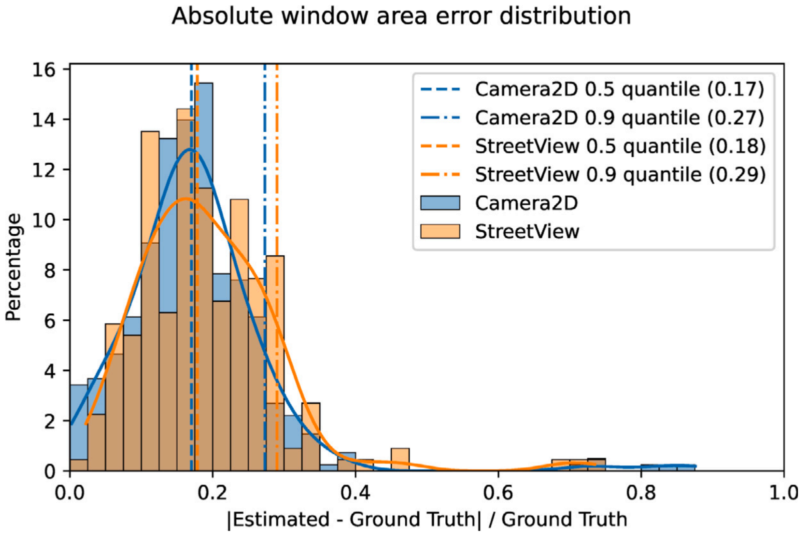

Fig. 17. Distribution of absolute errors in window area estimations.

both recall and precision at higher heights, because the lower parts of facades are more frequently obscured by obstacles such as vegetation and vehicles.

Another type of error is classification errors, where architectural features such as balconies with window-like structures are mistakenly identified as windows. To address this, we can fine-tune our deep learning models on more specialized datasets. While this approach demands additional resources – being both time-consuming and potentially costly – it could lead to improved accuracy tailored to specific use cases where such distinctions are critical.

Examples for qualitative visual inspection of these errors can be found in Fig. 16.

Areas of the windows and facades We evaluate the estimated area of the windows and facade by the normalized absolute error as shown in Figs. 17 and 18, respectively. The median (50% quantile) of the error is 9% for StreetView and 4% for Camera2D. It is expressed as a ratio of the absolute error to the actual value. As we can see that the performances for window area estimation are similar between Camera2D and StreetView, whereas Camera2D outperforms StreetView for facade area estimation due to the targeted data collection process, which reduces issues such as occlusion.

Location of the windows As we can see from Fig. 19, compared to the StreetView pipeline, Camera2D offers an advantage in terms of spatial accuracy due to its use of a fully 3D model for buildings. By constructing and utilizing 3D models, this pipeline more effectively captures the 3D structure of building facades. This results in a more faithful representation of the physical world and hence minimizes parallax errors.

The StreetView pipeline relies on the assumption that all planes are flat, which can lead to significant parallax issues when these planes are viewed from oblique angles. The larger the angle of observation relative to the plane, the more severe the parallax error. This limitation is particularly visible in urban settings where the angle of data capture can vary widely.

The area and position errors, though higher for StreetView, remain moderate overall, particularly for buildings without extreme occlusion or long facades. Most of the inaccuracies come from slight shifts or scaling issues of detected windows rather than gross misdetections. In other words, the methods are generally able to identify the approximate window locations and sizes, but fine-grained geometric precision suffers under sparse or occluded conditions.

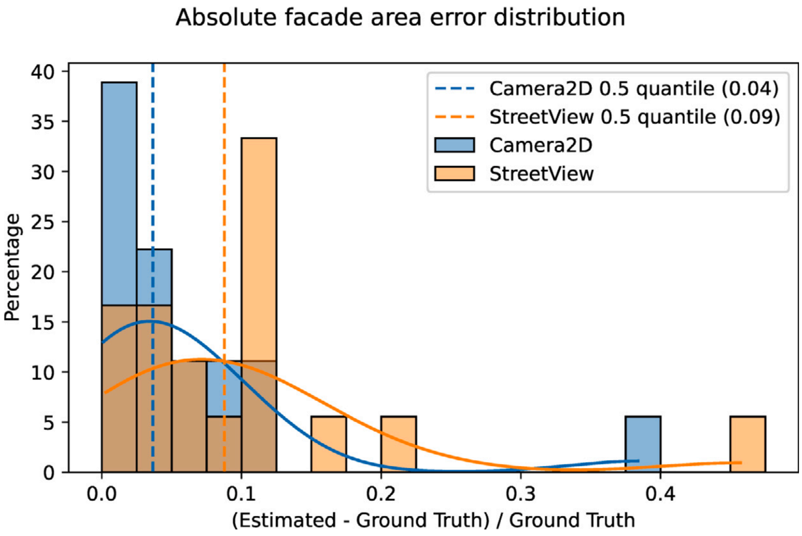

Fig. 18. Distribution of absolute errors in facade area estimations.

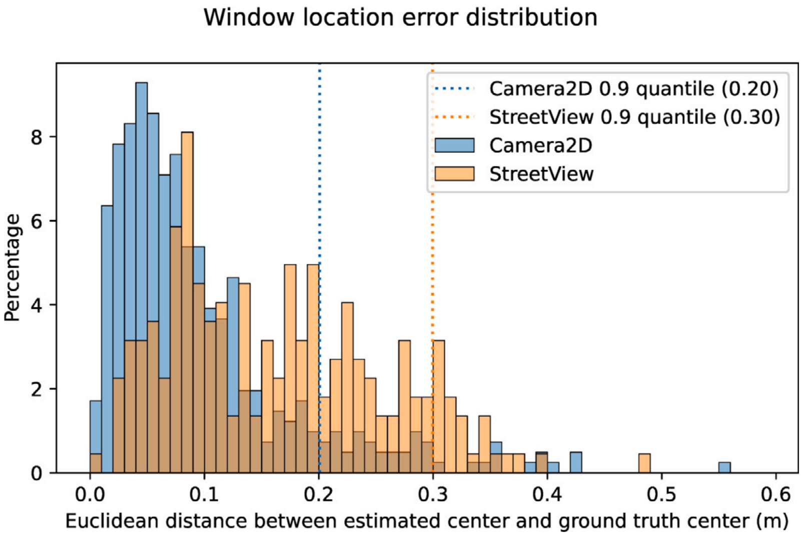

Fig. 19. Distribution of positional errors for window detections across different building facades.

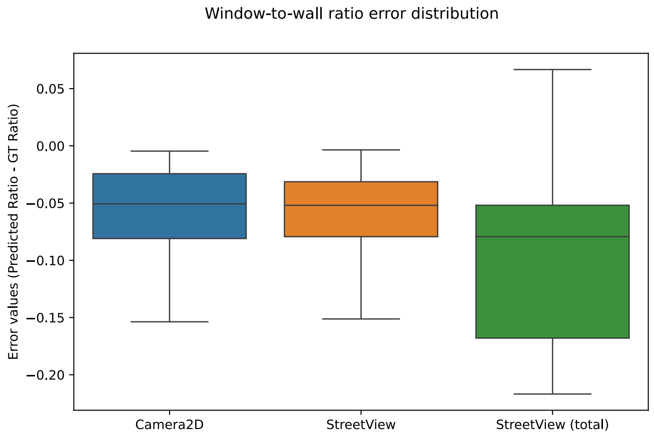

Fig. 20. Box plot of the aggregated window-to-wall ratio error (cf. 4.1) across various buildings.

WWR As shown in Fig. 20 and Table 3, we present the error of the estimated WWR. Camera2D and StreetView both yield comparable results in estimating the WWR with Camera2D performs slightly better.

The difference between StreetView and StreetView (Total) results highlights the critical impact of data completeness. Missing facades penalize StreetView Total F1-Scores and WWR errors significantly. This is because missing facade data is treated as having zero detected windows, dragging down global performance measures. However, when missing facades are accounted for through simple imputation (averaging WWR from detected facades), the WWR estimation improves substantially, illustrating that lightweight corrective strategies can partially compensate for sparse data gaps.

Scalability To address the scalability of the data collection, we summarize the comparison of different data sources (LiDAR, StreetView, and Camera2D) in Table 4. LiDAR provides a dense point cloud and precise building coordinates through manual annotation but the data collection requires approximately 120 min and is costly. In contrast, StreetView, which captures panoramic images along with camera pose and plane definitions, offers a sparse data density but is significantly quicker and cost-effective, requiring around 2 min using a data collection vehicle. When such a data collection vehicle is not available, online services and

Table 4
Benchmark comparison of 3D modeling approaches.

Method	WWR accuracy	Time (per building)		Scalability
	(Typ. error)	Collection	Processing	(Overall)
Conventional LiDAR	~1%	~120 min	~6.0 h	Low
SI3FP StreetView	~6%	~2 min	~0.1 h	Very High
SI3FP Camera2D	~5%	~30 min	~2.5 h	High

Table 5
Estimated computational requirements and scalability for SI3FP pipeline steps. .

Step	Resource(s)	Time (per building)	Scalability (per building)
StreetView Path Steps			
S.1	Network, CPU (Low)	~2 min	$\mathcal{O}(N_{pano})$
S.2	CPU (Multi-core), RAM	< 1 min	$\mathcal{O}(N_{pano}^2 N_{pl/pano}^2)$
S.3	CPU, GPU	< 1 min	$\mathcal{O}(N_f N_{pano/f}^2 W H)$
S.4	CPU	< 10 s	$\mathcal{O}(N_{line}^2)$
Camera2D Path Steps			
C.1	Manual Effort	~30 min	$\mathcal{O}(N_{img})$
C.2	CPU, GPU, High RAM	~2 h	$\sim O(N_{img}^3)$ (worst case), depends on K and HW
C.3	GPU (High VRAM), RAM	< 10 min	Training complexity of NeRF
Merged Steps			
M.1	GPU (VRAM)	~1 s	$\mathcal{O}(N_o HW + N_{box}^2)$
M.2	CPU (Low)	~1 s	$\mathcal{O}(N_f + N_w)$

resources can be utilized to achieve the data collection, with the cost being vendor-dependent. Camera2D captures dense perspective camera images with local coordinate reference points and takes about 30 min with a relatively low cost with a simple camera setup. This comparison highlights the trade-offs between data density, equipment complexity, time efficiency, and cost for scalable thermal modeling.

4.3. Computational performance and scalability considerations

Beyond accuracy, the practical applicability of the SI3FP pipeline depends on its computational requirements. Using the hardware detailed in Section 4.1 (NVIDIA RTX 3090 GPU, AMD Ryzen 7 5800X3D CPU, 64 GB RAM), we provide indicative performance characteristics for key stages. A summary, including the computational complexity and theoretical scalability, is provided in Table 5. For convenience, we denote the number of panorama images as N_{pano}, facades as N_f, orthographic images as N_o, windows as N_w, planes as N_{pl}, and image resolution as $H \times W$. The number of detected lines is denoted by N_{line} and detected boxes is N_{box}. For the COLMAP algorithm in C.2, we use K to denote the number of adjacent images for each image in the sequential matching, and we use N_{img} to denote the total images captured. Moreover, we use $N_{\text{pano}/f}$ to denote the number of panorama images per facade. Similarly, $N_{\text{pl/pano}}$ denotes the number of planes per panorama. In general, we use $N_{x/y}$ to denote the number of x per y. We assume the detection time of ResNet-50 is constant per pixel.

The Camera2D path involves the most computationally intensive steps per building. SfM (Step C.2 using COLMAP) typically required 2 h, demanding significant RAM and benefiting from GPU acceleration for feature matching. NeRF modeling (Step C.3 using Instant-NGP) required approximately 10 min of training time on the GPU, with its VRAM usage scaling with scene complexity; subsequent orthographic rendering was fast (seconds per view).

The StreetView path generally has a lower computational cost per building after the initial data acquisition (S.1). Plane clustering (S.2) and geometry-based facade detection (S.4) were relatively fast (seconds to minutes, CPU-bound). The primary computational step is often the multi-view image alignment (S.3), which took seconds to minutes per facade cluster on our hardware, its duration scaling with the number of views (N) to align and benefiting from GPU acceleration for parts like feature matching. For both paths, the final deep learning inference for window detection (M.1) was efficient when using the GPU, processing each facade image in approximately 20–100 ms.

Given these performance profiles, large-scale deployment for analyzing hundreds or thousands of buildings hinges on parallelization. The pipeline is well-suited for building-level parallelization, where each building is processed independently on separate compute resources. Comparing the two paths, the StreetView approach, despite potential bottlenecks in alignment for facades seen from very many views, generally offers higher throughput for large building stocks due to lower per-building compute demands (assuming data availability). The Camera2D path, while providing detailed NeRF models, requires significantly more GPU time and memory per building, making it better suited for targeted analysis or requiring substantial parallel computing infrastructure if applied at a very large scale.

5. Discussion

This paper explored different methods for generating building models suitable for early-stage energy simulation. Our proposed pipeline, along with alternative approaches, proved effective at capturing the thermal envelope – including roofs, floors, and walls – with sufficient accuracy for practical applications.

However, challenges remain in accurately detecting windows using data collected from budget sensors at scale, reflecting the core difficulty of balancing scalability and accuracy. The visual comparison in Fig. 21 highlights that no data source is flawless. For example, StreetView highly depends on the availability of unobstructed views towards the building; obstructions will decrease the amount of data that can be obtained. With dense image capturing methods, data collection from ground level severely limits the amount of data for the top floors. Additionally, the presence of balconies creates obstructions, making it difficult to accurately identify the dimensions or full shape of windows.

Limitation and future work We acknowledge several limitations in our current paper. First, regarding geometric fidelity and representation, projecting facades onto a single plane, as done in both the sparse and dense paths, can distort off-plane geometries such as curves and deep ornamentation. This design directly supports the objectives of our paper: for early-stage renovation analysis, the focus is on large-scale applicability rather than fine-grained modeling. Capturing complex geometries would require pre-segmentation into multiple planes, significantly

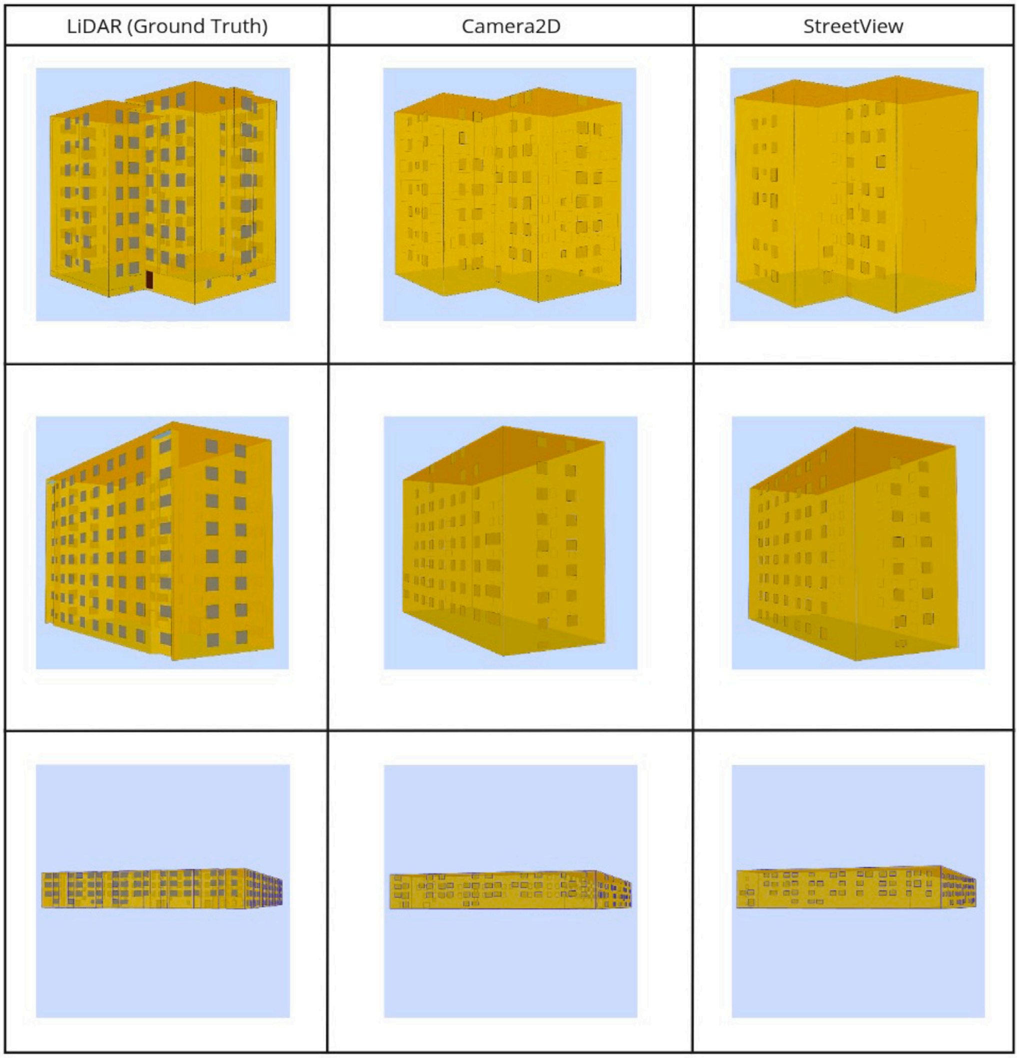

Fig. 21. Visual comparison between the different models obtained by each method, 3D LiDAR scanned (used as ground truth), StreetView, and Camera2D.

increasing computational cost, annotation burden, and system complexity. Similarly, the current bounding-box-based window detection (Step M.1) may not capture non-rectangular windows precisely, but it provides a scalable and effective solution for standard facade layouts. For applications requiring high-fidelity architectural reconstruction or conservation work, future extensions could integrate instance segmentation and multi-plane modeling. Second, challenging scene properties such as glass facades and occlusions remain difficult. Reflective and transparent materials degrade camera-based reconstruction, and dense urban clutter limits complete facade visibility. Although the StreetView ensemble strategy (Algorithm 6) partially mitigates these effects, fully addressing them would require multi-modal sensing or specialized modeling, which are promising directions for future research. Third, data acquisition constraints introduce limitations. Ground-level captures yield oblique views of upper facades, reducing resolution and orthographic clarity at height. Incorporating aerial imagery, such as UAV-based oblique views, would enhance coverage for tall buildings. Furthermore, while the incremental SfM stage (Step C.2) provides reliable reconstruction, it can be slow and prone to drift on large datasets. Future improvements include exploring global SfM approaches, such as GLOMAP [90], and integrating learned feature extractors and matchers to enhance scalability and robustness. Additionally, there is currently a lack of public benchmark datasets tailored for facade parsing and thermal modeling, limiting standardized evaluation across studies. Developing such datasets would be an important direction to advance the field.

Overall, our system was designed to balance accuracy, scalability, and computational feasibility, targeting practical early-phase renovation needs. Although we tested our approach only on Swedish buildings, their geometry and typology are representative of building stocks common throughout Europe, making the method likely transferable to broader European contexts. Future work will extend the system towards high-fidelity reconstruction as data availability and computational resources improve.

It is worth noting that while many factors influence simulation accuracy (e.g., infiltration rates, material properties, internal loads), geometric accuracy – particularly surface areas and volumes – is especially critical. Errors in modeled area or volume inevitably lead to large distortions in energy simulation results. Encouragingly, for opaque envelope elements, our method and alternatives achieve high dimensional accuracy. For transparent surfaces, underestimation of window areas remains an issue. In addition, missing facade from sparse data collection limits the detection capacity. This could be addressed in future work through pattern analysis techniques, although such synthesis lies beyond the current scope, which focuses on capturing existing building data rather than creating inferred models.

6. Conclusion

This paper presented SI3FP, a modular, scalable pipeline for generating thermal 3D models of buildings from visual data, designed to support early-stage renovation analysis. By integrating two complementary paths – StreetView for scalable, sparse data collection and Camera2D for targeted, high-resolution modeling – our system offers users flexibility in balancing cost, effort, and accuracy. In both paths, we apply orthographic transformation to create a consistent image domain for facade feature detection and geometry parameterization. This unified semantic interface reduces perspective distortion, simplifies geometry extraction, and enables modular reuse of downstream methods.

The technical contributions of SI3FP include a direct orthographic image generation approach that avoids projection artifacts, a multi-view geometric plane clustering and alignment method, and a tailored RANSAC-based facade extraction technique. These are combined with a unified interface for semantic parsing and 3D thermal modeling, resulting in an end-to-end system that maintains an approximate error of 5% in WWR estimation — well within the acceptable range for early-phase renovation planning.

Beyond its practical utility, SI3FP contributes several conceptual insights:

(1) **Multi-view fusion improves sparse data robustness:** Rather than selecting a single optimal image per facade, SI3FP's ensemble-based multi-view strategy improves resilience against occlusion and poor lighting, especially important in real-world street-level imagery.

(2) **Rendering is not sufficient for 3D modeling — parameterization is a key design choice:** Advanced image-to –3D rendering methods such as NeRF represent an important step towards automation. They enable data-rich, photorealistic reconstructions from unstructured images. However, rendering alone does not yield structured 3D geometry. To transition from rendering to actionable 3D models, *parameterization* is required, i.e., representing geometry as polylines, planes, or meshes. This is a design choice: the more granular the parameterization (e.g., arbitrarily shaped polygons, dense meshes), the more complex geometry it can represent — but also the more complexity and lower scalability it brings. Coarser primitives like rectangles or planes are easier to manage and often sufficient in early-stage modeling. The appropriate parameterization should be selected based on the use case, e.g., detailed facade conservation may justify mesh-based modeling, while large-scale energy planning may not. This is a design choice that needs to be carefully made jointly by the technical developers and end users. 3) **Task-specific modeling precision can be optimized without full 3D mesh reconstruction:** Our results show that accurate window detection and WWR estimation do not require full 3D mesh models. Instead, structured abstractions derived from image-plane primitives (e.g., bounding boxes, planes) can offer sufficient accuracy for thermal modeling at a much lower computational cost.

While SI3FP achieves satisfactory performance across diverse building geometries, several limitations remain that offer promising avenues for future work: (1) **Generalizing to more complex geometries:** Current plane-based simplification is insufficient for heavily curved facades or articulated surfaces. While not within the scope of our primary focus, it may be beneficial for other applications. Integrating multi-plane, polygonal, or spline-based representations – or even implicit surfaces such as signed distance functions – could increase expressiveness. However, as noted above, this increased expressiveness comes with added complexity. Therefore, exploring scalable algorithmic solutions is a promising direction for such use cases. (2) **From rendering to modeling:** Building on the insight above, future work could explore hybrid pipelines that integrate NeRF or other neural rendering techniques with parametric fitting algorithms to automatically extract geometry. This includes polygon fitting, mesh extraction, or rule-based facade grammar parsing. (3) **Physics-informed modeling:** Incorporating thermal priors or physically based constraints into the detection and modeling stages could better inform the placement and sizing of windows, especially in edge cases. (4) **Citizen-contributed and crowdsourced data:** To support continental-scale building assessments, future systems could integrate community-contributed photos and drone footage, extending SI3FP with scalable data collection and automated labeling and validation.

Beyond renovation potential analysis, the generated 3D models open up opportunities for broader applications, including 3D modeling for daylight analysis and construction material mapping for urban mining, reuse, and broader circular economy initiatives.

CRediT authorship contribution statement

Yinan Yu: Writing – original draft, Visualization, Validation, Methodology, Data curation, Conceptualization. **Alex Gonzalez-Caceres:** Writing – original draft, Data curation, Conceptualization, Validation. **Samuel Scheidegger:** Visualization, Data curation, Software, Writing – review & editing, Validation. **Sanjay Somanath:** Conceptualization. **Alexander Hollberg:** Writing – review & editing, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This work has been funded by Sweden's Innovation Agency Vinnova through the project DecarbonAIte [grant number 2021-02759] and it is part of the Digital Twin Cities Centre supported by Sweden's Innovation Agency Vinnova [grant number 2019-00041]. We would like to thank Dag Wästberg and Bernd Ketzler for their support in the project and Andreas Skälegård for providing data on the case study buildings. Furthermore, we would like to thank Jieming Yan for the support in the point cloud annotation. This work has also been funded by Chalmers Energy Area of Advance (AoA).

Data availability

Data will be made available on request.

References

[1] A. Galimshina, M. Moustapha, A. Hollberg, S. Lasvaux, B. Sudret, G. Habert, Strategies for robust renovation of residential buildings in Switzerland, Nat. Commun. 15 (1) (2024) 2227, http://dx.doi.org/10.1038/s41467-024-46305-9, Publisher: Nature Publishing Group.

[2] C.A. Balaras, K. Droutsa, E. Dascalaki, S. Kontoyiannidis, Deterioration of European apartment buildings, Energy Build. 37 (5) (2005) 515–527, http://dx.doi.org/10.1016/j.enbuild.2004.09.010, URL https://www.sciencedirect.com/science/article/pii/S0378778804002920.

[3] M. Bizjak, B. Žalik, N. Lukač, Parameter-free half-spaces based 3D building reconstruction using ground and segmented building points from airborne LiDAR data with 2d outlines, Remote. Sens. 13 (21) (2021) 4430, http://dx.doi.org/10.3390/rs13214430.

[4] Q. Zhu, Q. Shang, H. Hu, H. Yu, R. Zhong, Structure-aware completion of photogrammetric meshes in urban road environment, ISPRS J. Photogramm. Remote Sens. 175 (2021) 56–70, http://dx.doi.org/10.1016/j.isprsjprs.2021.02.010.

[5] R. Ma, R. Ma, E. Long, Analysis of the rule of window-to-wall ratio on energy demand of residential buildings in different locations in China, Heliyon 9 (1) (2023) e12803, http://dx.doi.org/10.1016/j.heliyon.2023.e12803, URL https://www.sciencedirect.com/science/article/pii/S2405844023000105.

[6] F. Biljecki, H. Ledoux, J. Stoter, An improved LOD specification for 3D building models, Comput. Environ. Urban Syst. 59 (2016) 25–37, http://dx.doi.org/10.1016/j.compenvurbsys.2016.04.005.

[7] T. Loga, B. Stein, N. Diefenbach, TABULA building typologies in 20 European countries—Making energy-related features of residential building stocks comparable, Energy Build. 132 (2016) 4–12, http://dx.doi.org/10.1016/j.enbuild.2016.06.094.

[8] R. Mohammadiziazi, S. Copeland, M.M. Bilec, Urban building energy model: Database development, validation, and application for commercial building stock, Energy Build. 248 (2021) 111175, http://dx.doi.org/10.1016/j.enbuild.2021.111175.

[9] M. Österbring, E. Mata, L. Thuvander, M. Mangold, F. Johnsson, H. Wallbaum, A differentiated description of building-stocks for a georeferenced urban bottom-up building-stock model, Energy Build. 120 (2016) 78–84, http://dx.doi.org/10.1016/j.enbuild.2016.03.060.

[10] Y. Li, C. Wang, S. Zhu, J. Yang, S. Wei, X. Zhang, X. Shi, A comparison of various bottom-up urban energy simulation methods using a case study in Hangzhou, China, Energies 13 (18) (2020) http://dx.doi.org/10.3390/en13184781.

[11] M. Aydinalp-Koksal, V.I. Ugursal, Comparison of neural network, conditional demand analysis, and engineering approaches for modeling end-use energy consumption in the residential sector, Appl. Energy 85 (4) (2008) 271–296, http://dx.doi.org/10.1016/j.apenergy.2006.09.012.

[12] J.A. Fonseca, A. Schlueter, Integrated model for characterization of spatiotemporal building energy consumption patterns in neighborhoods and city districts, Appl. Energy 142 (2015) 247–265, http://dx.doi.org/10.1016/j.apenergy.2014.12.068.

[13] L. Li, H.P.H. Shum, T.P. Breckon, Less is more: Reducing task and model complexity for 3D point cloud semantic segmentation, in: 2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, Vancouver, BC, Canada, 2023, pp. 9361–9371, http://dx.doi.org/10.1109/CVPR52729.2023.00903.

[14] I. Maksymova, C. Steger, N. Druml, Review of LiDAR sensor data acquisition and compression for automotive applications, 2, (13) MDPI AG, 2018, http://dx.doi.org/10.3390/proceedings2130852, Eurosensors 2018 ; Conference date: 09-09-2018 Through 12-09-2018URL https://www.eurosensors2018.eu.

[15] Y. Li, J. Ibanez-Guzman, Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems, IEEE Signal Process. Mag. 37 (4) (2020) 50–61, http://dx.doi.org/10.1109/MSP.2020.2973615.

[16] Z. Zou, K. Chen, Z. Shi, Y. Guo, J. Ye, Object detection in 20 years: A survey, Proc. IEEE 111 (3) (2023) 257–276, http://dx.doi.org/10.1109/JPROC.2023.3238524.

[17] Z.-Q. Zhao, P. Zheng, S.-t. Xu, X. Wu, Object detection with deep learning: A review, IEEE Trans. Neural Netw. Learn. Syst. 30 (11) (2019) 3212–3232, http://dx.doi.org/10.48550/arXiv.1807.05511.

[18] A.M. Hafiz, G.M. Bhat, A survey on instance segmentation: state of the art, Int. J. Multimed. Inf. Retr. 9 (3) (2020) 171–189, http://dx.doi.org/10.1007/s13735-020-00195-x.

[19] R.I. Hartley, A. Zisserman, Multiple View Geometry in Computer Vision, second ed., Cambridge University Press, Cambridge, UK, 2004, http://dx.doi.org/10.1017/CBO9780511811685.

[20] D. Gledhill, G.Y. Tian, D. Taylor, D. Clarke, Panoramic imaging—a review, Comput. Graph. 27 (3) (2003) 435–445, http://dx.doi.org/10.1016/S0097-8493(03)00038-4.

[21] T. Jokela, J. Ojala, K. Väänänen, How people use 360-degree cameras, in: Proceedings of the 18th International Conference on Mobile and Ubiquitous Multimedia, MUM '19, Association for Computing Machinery, New York, NY, USA, 2019, pp. 1–10, http://dx.doi.org/10.1145/3365610.3365645.

[22] R. Szeliski, Computer Vision: Algorithms and Applications, Springer Nature, 2022, pp. 46–48, http://dx.doi.org/10.1007/978-3-030-34372-9.

[23] P. Pesti, J. Elson, J. Howell, D. Steedly, M. Uyttendaele, Low-cost orthographic imagery, in: Proceedings of the 16th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, GIS '08, Association for Computing Machinery, New York, NY, USA, 2008, pp. 1–8, http://dx.doi.org/10.1145/1463434.1463465.

[24] S.T. Barnard, M.A. Fischler, Computational stereo, ACM Comput. Surv. 14 (4) (1982) 553–572, http://dx.doi.org/10.1145/356893.356896.

[25] Y. Furukawa, C. Hernández, et al., Multi-view stereo: A tutorial, Found. Trends® Comput. Graph. Vis. 9 (1–2) (2015) 1–148, http://dx.doi.org/10.1561/0600000052.

[26] M.S. Hamid, N.A. Manap, R.A. Hamzah, A.F. Kadmin, Stereo matching algorithm based on deep learning: A survey, J. King Saud Univ. - Comput. Inf. Sci. 34 (5) (2022) 1663–1673, http://dx.doi.org/10.1016/j.jksuci.2020.08.011.

[27] J. Villanueva, A. Blanco, Optimization of ground control point (GCP) configuration for unmanned aerial vehicle (UAV) survey using structure from motion (SFM), Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci. 42 (2019) 167–174, http://dx.doi.org/10.5194/isprs-archives-XLII-4-W12-167-2019.

[28] J. Parente, E. Rodrigues, B. Rangel, J. Poças Martins, Integration of convolutional and adversarial networks into building design: A review, J. Build. Eng. 76 (2023) 107155, http://dx.doi.org/10.1016/j.jobe.2023.107155.

[29] P. Musialski, P. Wonka, D.G. Aliaga, M. Wimmer, L. Van Gool, W. Purgathofer, A survey of urban reconstruction, Comput. Graph. Forum 32 (6) (2013) 146–177, http://dx.doi.org/10.1111/cgf.12077.

[30] A. Klimkowska, S. Cavazzi, R. Leach, S. Grebby, Detailed three-dimensional building façade reconstruction: A review on applications, data and technologies, Remote. Sens. 14 (11) (2022) 2579, http://dx.doi.org/10.3390/rs14112579.

[31] Y. Li, L. Peng, C. Wu, J. Zhang, Street view imagery (SVI) in the built environment: A theoretical and systematic review, Buildings 12 (8) (2022) 1167, http://dx.doi.org/10.3390/buildings12081167.

[32] M. Schmitz, H. Mayer, A convolutional network for semantic facade segmentation and interpretation, Int. Arch. Photogramm. Remote. Sens. Spat. Inf. Sci. XLI-B3 (2016) 709–715, http://dx.doi.org/10.5194/isprs-archives-XLI-B3-709-2016.

[33] H. Liu, Y. Xu, J. Zhang, J. Zhu, Y. Li, S.C.H. Hoi, DeepFacade: A deep learning approach to facade parsing with symmetric loss, IEEE Trans. Multimed. 22 (12) (2020) 3153–3165, http://dx.doi.org/10.1109/TMM.2020.2971431.

[34] G. Zhang, Y. Pan, L. Zhang, Deep learning for detecting building façade elements from images considering prior knowledge, Autom. Constr. 133 (2022) 104016, http://dx.doi.org/10.1016/j.autcon.2021.104016.

[35] H. Zhao, J. Shi, X. Qi, X. Wang, J. Jia, Pyramid scene parsing network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Honolulu, HI, USA, 2017, pp. 2881–2890, http://dx.doi.org/10.1109/CVPR.2017.660.

[36] J. Fu, J. Liu, H. Tian, Y. Li, Y. Bao, Z. Fang, H. Lu, Dual attention network for scene segmentation, in: 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2019, pp. 3141–3149, http://dx.doi.org/10.1109/CVPR.2019.00326.

[37] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, S. Zagoruyko, End-to-end object detection with transformers, in: European Conference on Computer Vision, Springer, 2020, pp. 213–229, http://dx.doi.org/10.48550/arXiv.2005.12872.

[38] Y. Lu, W. Wei, P. Li, T. Zhong, Y. Nong, X. Shi, A deep learning method for building façade parsing utilizing improved SOLOv2 instance segmentation, Energy Build. 295 (2023) 113275, http://dx.doi.org/10.1016/j.enbuild.2023.113275, URL https://www.sciencedirect.com/science/article/pii/S0378778823005054.

[39] B. Wang, J. Zhang, R. Zhang, Y. Li, L. Li, Y. Nakashima, Improving facade parsing with vision transformers and line integration, Adv. Eng. Inform. 60 (2024) 102463, http://dx.doi.org/10.1016/j.aei.2024.102463.

[40] O. Teboul, I. Kokkinos, L. Simon, P. Koutsourakis, N. Paragios, Parsing facades with shape grammars and reinforcement learning, IEEE Trans. Pattern Anal. Mach. Intell. 35 (7) (2013) 1744–1756, http://dx.doi.org/10.1109/TPAMI.2012.252.

[41] M. Mathias, A.e. Martinović, L. Van Gool, ATLAS: A three-layered approach to facade parsing, Int. J. Comput. Vis. 118 (1) (2016) 22–48, http://dx.doi.org/10.1007/s11263-015-0868-z.

[42] M. Neuhausen, M. König, Automatic window detection in facade images, Autom. Constr. 96 (2018) 527–539, http://dx.doi.org/10.1016/j.autcon.2018.10.007.

[43] G. Kong, H. Fan, Enhanced facade parsing for street-level images using convolutional neural networks, IEEE Trans. Geosci. Remote Sens. 59 (12) (2021) 10519–10531, http://dx.doi.org/10.1109/TGRS.2020.3035878.

[44] K. Rahmani, H. Mayer, High quality facade segmentation based on structured random forest, region proposal network and rectangular fitting, ISPRS Ann. Photogramm. Remote. Sens. Spat. Inf. Sci. IV-2 (2018) 223–230, http://dx.doi.org/10.5194/isprs-annals-IV-2-223-2018.

[45] J.T. Szcześniak, Y.Q. Ang, S. Letellier-Duchesne, C.F. Reinhart, A method for using street view imagery to auto-extract window-to-wall ratios and its relevance for urban-level daylighting and energy simulations, Build. Environ. 207 (2022) 108108, http://dx.doi.org/10.1016/j.buildenv.2021.108108.

[46] C.-K. Li, H.-X. Zhang, J.-X. Liu, Y.-Q. Zhang, S.-C. Zou, Y.-T. Fang, Window detection in facades using heatmap fusion, J. Comput. Sci. Tech. 35 (4) (2020) 900–912, http://dx.doi.org/10.1007/s11390-020-0253-4.

[47] J. Cao, H. Metzmacher, J. O'Donnell, J. Frisch, V. Bazjanac, L. Kobbelt, C. van Treeck, Facade geometry generation from low-resolution aerial photographs for building energy modeling, Build. Environ. 123 (2017) 601–624, http://dx.doi.org/10.1016/j.buildenv.2017.07.018.

[48] B. Wang, M. Li, Z. Peng, W. Lu, Hierarchical attributed graph-based generative façade parsing for high-rise residential buildings, Autom. Constr. 164 (2024) 105471, http://dx.doi.org/10.1016/j.autcon.2024.105471.

[49] C. Ayala, R. Sesma, C. Aranda, M. Galar, A deep learning approach to an enhanced building footprint and road detection in high-resolution satellite imagery, Remote. Sens. 13 (16) (2021) 3135, http://dx.doi.org/10.3390/rs13163135.

[50] R. Goel, L.M.T. Garcia, A. Goodman, R. Johnson, R. Aldred, M. Murugesan, S. Brage, K. Bhalla, J. Woodcock, Estimating city-level travel patterns using street imagery: A case study of using google street view in Britain, PLoS One 13 (5) (2018) 1–22, http://dx.doi.org/10.1371/journal.pone.0196521.

[51] L. Long, How green are the streets? An analysis for central areas of Chinese cities using tencent street view, PLoS One 12 (2) (2017) 1–18, http://dx.doi.org/10.1371/journal.pone.0171110.

[52] M. Cavallo, 3D City Reconstruction from Google Street View Imagery, Università degli Studi di Verona, Verona, Italy, 2015, URL https://www.evl.uic.edu/documents/3drecomstrictionmcavallo.pdf, (Accessed 31 July 2025).

[53] B. Mildenhall, P.P. Srinivasan, M. Tancik, J.T. Barron, R. Ramamoorthi, R. Ng, Nerf: Representing scenes as neural radiance fields for view synthesis, Commun. ACM 65 (1) (2021) 99–106, http://dx.doi.org/10.48550/arXiv.2003.08934.

[54] T. Müller, A. Evans, C. Schied, A. Keller, Instant neural graphics primitives with a multiresolution hash encoding, ACM Trans. Graph. 41 (4) (2022) 1–15, http://dx.doi.org/10.1145/3528223.3530127.

[55] M. Tancik, V. Casser, X. Yan, S. Pradhan, B.P. Mildenhall, P. Srinivasan, J.T. Barron, H. Kretzschmar, Block-nerf: Scalable large scene neural view synthesis, in: 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, 2022, pp. 8238–8248, http://dx.doi.org/10.1109/CVPR52688.2022.00807.

[56] H. Riemenschneider, A. Bódis-Szomorú, J. Weissenberg, L. Van Gool, Learning where to classify in multi-view semantic segmentation, in: D. Fleet, T. Pajdla, B. Schiele, T. Tuytelaars (Eds.), Computer Vision – ECCV 2014, in: Lecture Notes in Computer Science, vol. 8693, Springer International Publishing, Cham, 2014, pp. 516–532, http://dx.doi.org/10.1007/978-3-319-10602-1_34.

[57] R.G. Lotte, N. Haala, M. Karpina, L.E.O.e.C.d. Aragão, Y.E. Shimabukuro, 3D façade labeling over complex scenarios: A case study using convolutional neural network and structure-from-motion, Remote. Sens. 10 (9) (2018) http://dx.doi.org/10.3390/rs10091435.

[58] S. Malihi, M.J. Valadan Zoej, M. Hahn, M. Mokhtarzade, Window detection from UAS-derived photogrammetric point cloud employing density-based filtering and perceptual organization, Remote. Sens. 10 (8) (2018) 1320, http://dx.doi.org/10.3390/rs10081320.

[59] K. Bacharidis, F. Sarri, L. Ragia, 3D building façade reconstruction using deep learning, ISPRS Int. J. Geo- Inf. 9 (5) (2020) 322, http://dx.doi.org/10.3390/ijgi9050322.

[60] K. Bacharidis, F. Sarri, V. Paravolidakis, L. Ragia, M. Zervakis, Fusing georeferenced and stereoscopic image data for 3D building facade reconstruction, ISPRS Int. J. Geo- Inf. 7 (4) (2018) 151, http://dx.doi.org/10.3390/ijgi7040151.

[61] A. Pal, J.J. Lin, S.-H. Hsieh, M. Golparvar-Fard, Activity-level construction progress monitoring through semantic segmentation of 3D-informed orthographic images, Autom. Constr. 157 (2024) 105157, http://dx.doi.org/10.1016/j.autcon.2023.105157, URL https://www.sciencedirect.com/science/article/pii/S092658052300417X.

[62] G. Nishida, A. Bousseau, D.G. Aliaga, Procedural modeling of a building from a single image, Comput. Graph. Forum 37 (2) (2018) 415–429, http://dx.doi.org/10.1111/cgf.13372.

[63] B. Pantoja-Rosero, R. Achanta, M. Kozinski, P. Fua, F. Perez-Cruz, K. Beyer, Generating LOD3 building models from structure-from-motion and semantic segmentation, Autom. Constr. 141 (2022) 104430, http://dx.doi.org/10.1016/j.autcon.2022.104430.

[64] W. Ward, X. Li, Y. Sun, M. Dai, H. Arbabi, D.D. Tingley, M. Mayfield, Estimating energy consumption of residential buildings at scale with drive-by image capture, Build. Environ. 234 (2023) 110188, http://dx.doi.org/10.1016/j.buildenv.2023.110188.

[65] A. Salehitangrizi, S. Jabari, M. Sheng, Y. Zhang, 3D modeling of façade elements using multi-view images from mobile scanning systems, Can. J. Remote Sens. 50 (1) (2024) 2309895, http://dx.doi.org/10.1080/07038992.2024.2309895.

[66] X. Zhang, K. Chen, H. Johan, M. Erdt, SLOD2+WIN: semantics-aware addition and LoD of 3D window details for LoD2 CityGML models with textures, Vis. Comput. (2024) http://dx.doi.org/10.1007/s00371-024-03304-7.

[67] N. Tarkhan, J.T. Szcześniak, C. Reinhart, Façade feature extraction for urban performance assessments: Evaluating algorithm applicability across diverse building morphologies, Sustain. Cities Soc. 105 (2024) 105280, http://dx.doi.org/10.1016/j.scs.2024.105280.

[68] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A.C. Berg, W.-Y. Lo, et al., Segment anything, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2023, pp. 4015–4026, http://dx.doi.org/10.48550/arXiv.2304.02643.

[69] M.O. Wong, H. Ying, M. Yin, X. Yi, L. Xiao, W. Duan, C. He, L. Tang, Semantic 3D reconstruction-oriented image dataset for building component segmentation, Autom. Constr. 165 (2024) 105558, http://dx.doi.org/10.1016/j.autcon.2024.105558.

[70] Y. Dehbi, F. Hadiji, G. Gröger, K. Kersting, L. Plümer, Statistical relational learning of grammar rules for 3D building reconstruction, Trans. GIS 21 (1) (2017) 134–150, http://dx.doi.org/10.1111/tgis.12200.

[71] Z. Li, L. Zhang, P.T. Mathiopoulos, F. Liu, L. Zhang, S. Li, H. Liu, A hierarchical methodology for urban facade parsing from TLS point clouds, ISPRS J. Photogramm. Remote Sens. 123 (2017) 75–93, http://dx.doi.org/10.1016/j.isprsjprs.2016.11.008.

[72] A. Fryskowska, J. Stachelek, A no-reference method of geometric content quality analysis of 3D models generated from laser scanning point clouds for hBIM, J. Cult. Herit. 34 (2018) 95–108, http://dx.doi.org/10.1016/j.culher.2018.04.003.

[73] R. Gadde, V. Jampani, R. Marlet, P.V. Gehler, Efficient 2D and 3D facade segmentation using auto-context, IEEE Trans. Pattern Anal. Mach. Intell. 40 (5) (2018) 1273–1280, http://dx.doi.org/10.1109/TPAMI.2017.2696526.

[74] X. Wen, H. Xie, H. Liu, L. Yan, Accurate reconstruction of the LoD3 building model by integrating multi-source point clouds and oblique remote sensing imagery, ISPRS Int. J. Geo- Inf. 8 (3) (2019) 135, http://dx.doi.org/10.3390/ijgi8030135.

[75] H. Fan, Y. Wang, J. Gong, Layout graph model for semantic façade reconstruction using laser point clouds, Geo- Spat. Inf. Sci. 24 (2021) 1–19, http://dx.doi.org/10.1080/10095020.2021.1922316.

[76] S. Hachisuka, A. Tono, M. Fisher, Harbingers of NeRF-to-BIM: a case study of semantic segmentation on building structure with neural radiance fields, in: EC3 Conference 2023, European Council on Computing in Construction, Heraklion, Crete, Greece, 2023, pp. 576–583, http://dx.doi.org/10.35490/EC3.2023.284.

[77] J.L. Schönberger, E. Zheng, J.-M. Frahm, M. Pollefeys, Pixelwise view selection for unstructured multi-view stereo, in: B. Leibe, J. Matas, N. Sebe, M. Welling (Eds.), Computer Vision – ECCV 2016, Springer International Publishing, Cham, 2016, pp. 501–518, http://dx.doi.org/10.1007/978-3-319-46487-9_31.

[78] D.G. Lowe, Distinctive image features from scale-invariant keypoints, Int. J. Comput. Vis. 60 (2) (2004) 91–110, http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94.

[79] M.A. Fischler, R.C. Bolles, Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography, Commun. ACM 24 (6) (1981) 381–395, http://dx.doi.org/10.1145/358669.358692.

[80] G. Kong, H. Fan, Enhanced facade parsing for street-level images using convolutional neural networks, IEEE Trans. Geosci. Remote Sens. 59 (12) (2021) 10519–10531, http://dx.doi.org/10.1109/TGRS.2020.3045604.

[81] C.-K. Liang, L.-W. Chang, H.H. Chen, Analysis and compensation of rolling shutter effect, IEEE Trans. Image Process. 17 (8) (2008) 1323–1330, http://dx.doi.org/10.1109/TIP.2008.925384.

[82] Z. Ma, R. Liao, X. Tao, L. Xu, J. Jia, E. Wu, Handling motion blur in multi-frame super-resolution, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, Boston, MA, USA, 2015, pp. 5224–5232, http://dx.doi.org/10.1109/CVPR.2015.7299159.

[83] J.L. Schönberger, J.-M. Frahm, Structure-from-motion revisited, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, Las Vegas, NV, USA, 2016, pp. 4104–4113, http://dx.doi.org/10.1109/CVPR.2016.445.

[84] P. Zhu, W.R. Para, A. Frühstück, J. Femiani, P. Wonka, Large-scale architectural asset extraction from panoramic imagery, IEEE Trans. Vis. Comput. Graphics 28 (2) (2022) 1301–1316, http://dx.doi.org/10.1109/TVCG.2020.3010694.

[85] M. Mangold, M. Österbring, H. Wallbaum, L. Thuvander, P. Femenias, Socio-economic impact of renovation and energy retrofitting of the gothenburg building stock, Energy Build. 123 (2016) 41–49, http://dx.doi.org/10.1016/j.enbuild.2016.04.033, URL https://www.sciencedirect.com/science/article/pii/S0378778816302894.

[86] T. Hall, S. Vidén, The million homes programme: a review of the great Swedish planning project, Plan. Perspect. 20 (3) (2005) 301–328, http://dx.doi.org/10.1080/02665430500130233, arXiv:https://doi.org/10.1080/02665430500130233.

[87] G. Savvidou, B. Nykvist, Heat demand in the Swedish residential building stock - pathways on demand reduction potential based on socio-technical analysis, Energy Policy 144 (2020) 111679, http://dx.doi.org/10.1016/j.enpol.2020.111679, URL https://www.sciencedirect.com/science/article/pii/S0301421520304080.

[88] M. Everingham, L. Van Gool, C.K.I. Williams, J. Winn, A. Zisserman, The pascal visual object classes (VOC) challenge, Int. J. Comput. Vis. 88 (2) (2010) 303–338, http://dx.doi.org/10.1007/s11263-009-0275-4.

[89] H. Bulow-Hube, The effect of glazing type and size on annual heating and cooling demand for Swedish offices, 1998, URL https://www.osti.gov/etdeweb/biblio/635147, (Accessed 31 July 2025).

[90] L. Pan, D. Baráth, M. Pollefeys, J.L. Schönberger, Global structure-from-motion revisited, in: A.s. Leonardis, E. Ricci, S. Roth, O. Russakovsky, T. Sattler, G. Varol (Eds.), Computer Vision – ECCV 2024, Springer Nature Switzerland, Cham, 2025, pp. 58–77, http://dx.doi.org/10.1007/978-3-031-73661-2_4.