3D Façade Element Extraction from Image-based Instance Segmentation and Scale-Invariant Object Contour Points

Florian Frank[1], Venus Shah[1], Simon Worbis[2], Ludwig Hoegner[2]

[1] IWT Friedrichshafen, Germany - (frank@iwt-bodensee.de, shah@iwt-bodensee.de)

[2] Hochschule München University of Applied Sciences, Department of Geoinformatics, Germany - (simon.worbis, ludwig.hoegner)@hm.edu

Keywords: Instance Segmentation, LoD3 Reconstruction, Pose Estimation, Invariant Feature Extraction, RGB Images

Abstract

Real-world 3D reconstructions of building façades in LoD3 and beyond are not yet widely available on the mass market due to financial and technological barriers, as well as the challenges of automated modeling. We propose a novel method for extracting 3D façade elements using image-based instance segmentation and scale-invariant object contour points (SIOCP). Our methodology uses RGB images, camera parameters, absolute 6DoF pose and position, as well as LoD2 building information. The images are processed using instance segmentation with YOLOv8 and SAM, complemented by classical and enhanced algorithms for line and edge detection. The SIOCP method refines object contour lines from instance segmentation by incorporating LoD2 building data and 6DoF information. Subsequently, the keypoints are estimated and the single-camera image 6DoF pose is reconstructed using a PnP solver. From these 6DoF poses a photogrammetrically point cloud is generated, and semantically- and instance-segmented with SuperCluster. The segmentation results are intended for future comparisons with other point clouds and LoD3 reconstructions. The presented approach is still under development, so the current results are limited. In summary, this paper introduces a key component of our vision for LoD3 reconstruction by using handheld devices.

1. Introduction

Future city models require LoD3 quality and can be considered as a cross-sectional technology for many domains. Examples include urban planning, rescue during fire, and shading analysis for photovoltaic systems. However, LoD3 models have not yet reached the mass market. Current real-world available LoD3 reconstructions are mainly based on MLS and semi-automatic data processing (Wysocki O. et al., 2022). The high financial and technological barriers limit the widespread creation of LoD3 models. As a result, in the SAVE (Schwab B. and Wysocki O., 2021) research project in Europe, only one real-world street reconstruction was carried out. It becomes clear that the data processing aspect of the project was the most time-consuming.

To address this challenge, we propose a novel method for extracting 3D façade elements using image-based instance segmentation and scale-invariant object contour points. We introduce a new data processing pipeline for RGB image processing. The aim of this paper is to facilitate LoD3 reconstruction for the mass market by lowering technical and financial entry barriers. The ideal scenario would enable the reconstruction of LoD3 models using widely available consumer devices, such as smartphones.

2. Related Work

Everything began with the idea of object-based outdoor localization by using a monocamera system. For this purpose, a handcart was built with a 720° gimbal for environmental perception with a monocamera system. Additionally, the handcart uses an absolute positioning system consisting of GNSS RTK and IMU (Frank et al., 2024b). The handcart was used to digitalize the Campus Fallenbrunnen in Friedrichshafen, Germany and TUM experimental farm Roggenstein. During digitalization, the handcart placed in different positions and at each position 200 to 360 12MP images were taken. This high resolution of the images allows visually to detect façade details, like door handle bars. Further, this high level of detail led to the creation of a 39-class object catalog with instructions for instance segmentation labeling of façade elements (Frank et al., 2024c).

This is where out paper contributes to LoD3 reconstruction. LoD3 reconstruction methods can be distinguished into image-based, point cloud-based, and hybrid approaches. Each reconstruction approach relies on a dataset, which is processed using statistical or AI methods to create LoD3 models. Point cloud-based approaches are primarily based on mobile laser scans (MLS) and may involve color-enriched point clouds. For example, Scan2LoD3 (Wysocki et al., 2023) proposes a method based on Bayesian networks, while A. Yarroudh et al. utilize a Grounding Dino AI approach. The camera-based approaches by B.G. Pantoja-Rosero et al. (Pantoja-Rosero et al., 2022) and H. Huang et al. (Huang et al., 2020) use structure-from-motion (SfM) point clouds for outer shell detection and segment openable objects using deep learning methods. Both approaches generate outer shells at LoD2 level using statistical methods for point cloud processing (Nan and Wonka, 2017), while windows and doors are instance-segmented in images via deep learning models such as DenseNet56 and TernausNet. The semi-automated approach to LoD3 creation (Harshit et al., 2024) co-registers Apple LiDAR and UAV-derived photogrammetric point clouds. Its LoD3 reconstruction pipeline consists of Autodesk Revit and FME Workbench for processing.

Finally, our approach differs from others because we do not create a high-density point cloud to derive the building's outer shell. Instead, we first extract the objects, refine the object contours, and reconstruct the camera poses and façade elements using scale-invariant object contour points (SIOCP).

To validate our SIOCP results, a high-density point cloud is generated and related to following works: 3D deep learning experienced a breakthrough in segmenting 3D data with (Qi et al., 2017a), which introduced *PointNet*, a network architecture for segmentation that manages fundamental properties of point clouds, such as unordered and permutation-invariant points. However, *PointNet* fails to leverage the spatial neighborhood of points for feature extraction, which is crucial for identifying local patterns. Subsequent works addressed this limitation with architectures such as *PointNet++* (Qi et al., 2017b) and *DGCNN* (Wang et al., 2019). These methods primarily focus on accuracy rather than the efficient processing of large-scale data, a critical factor when creating large-scale LoD3 models, as individual building façades can consist of millions of points. Processing such large datasets can push modern hardware to its limits, making data aggregation essential. However, downsampling point clouds into dense voxel grids contradicts the inherently sparse structure of point clouds (Graham et al., 2018). This approach results in a large proportion of grid cells being unoccupied, significantly increasing computational costs and memory consumption. Equally important, coordinate-based voxelization ignores object semantics, leading to inaccuracies (Landrieu and Simonovsky, 2018). Conversely, (Landrieu and Simonovsky, 2018) propose an approach that explicitly utilizes semantic information to partition the point cloud into so-called *superpoints*, which are then embedded as nodes in a graph neural network for learning. Similar to how *PointNet++* improved on *PointNet*, the *Superpoint Transformer (SPT)* (Robert et al., 2023) builds on the superpoint graph method. SPT employs a multi-scale processing scheme to exploit rich neighborhood information and introduces an attention mechanism (Vaswani et al., 2017), thereby achieving high performance.

3. Methodology

The methodology section describes a workflow for automatically creating LoD3 building models from street-level image data. An overview of the approach is presented in Section **System Architecture**. Section **Data Annotation and Preparation** explains the process of data annotation and preparation. Sections **Object Extraction** and **Scale-Invariant Object Contour Points (SIOCP)** outline our approach for identifying object contours in images and recovering precise camera poses by matching those contours. Based on these poses, a dense point cloud can be reconstructed from the images using semi-global matching. Finally, section **Object Segmentation with SuperCluster** describes the object segmentation of the generated point cloud, allowing us to evaluate the previous object extraction results.

3.1 System Architecture

The system architecture is based on the concept of a monocamera object localization approach. This idea was further developed into a suitable object catalog (Frank et al., 2024a), which includes buildings and vegetation for localization. Additionally, a handcart tool for data acquisition (Frank et al., 2024b) and a data annotation strategy for building façade elements were developed (Frank et al., 2024c). The starting point of the proposed system architecture is the acquired data types, which consist of 12MP RGB images, absolute 6DoF pose and position data derived from GNSS RTK, IMU, and gimbal pose, as well as date and time stamps. Figure 1 provides an overview of the system architecture and introduces the proposed methodology. The sequence of steps in the presented approach is indicated by numbers highlighted in yellow. The first step was data annotation

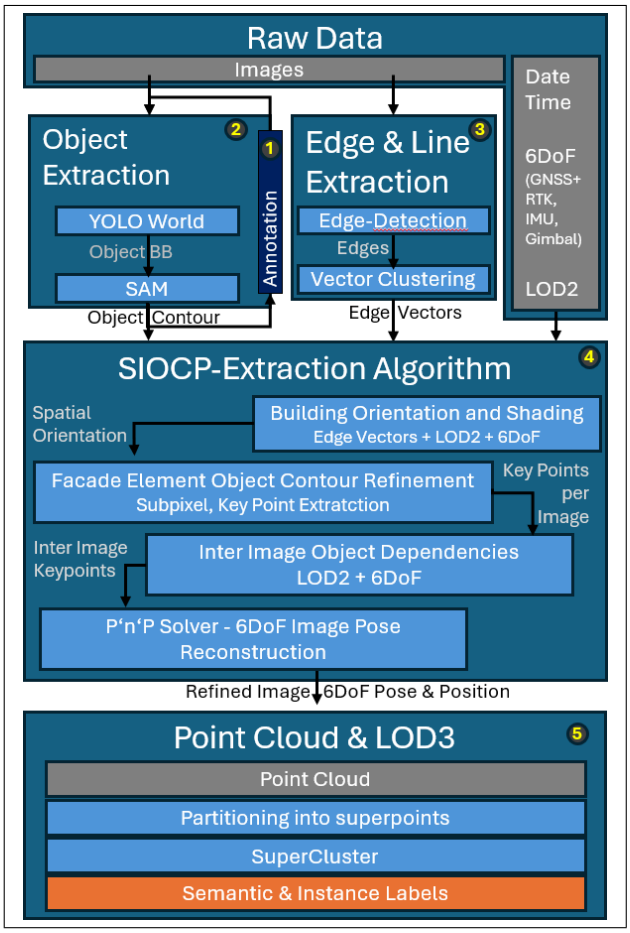


Figure 1. System architecture overview to introduce in the methodology.

and preparation for the approach. This was necessary due to certain technological barriers, which are discussed in subsection 3.2. The object extraction (see subsection 3.3) is used dual to accelerate the annotation process and segmentation automation. In the third step, described in subsection 3.4, raw images were processed into edge vectors. The fourth step combines object contours, edge vectors, and raw input data to generate Scale-Invariant Object Contour Points (SIOCP). These SIOCPs enable precise reconstruction of the camera’s 6-DoF pose and position. To compare the results we photogrammetrically reconstructed the point cloud and compard data with other point clouds.

3.2 Data Annotation and Preparation

For this approach, 115 positions with 240 to 370 RGB images each were captured at the Campus Fallenbrunnen and the experimental farm Roggenstein, resulting in a total of 20,365 12MP images to process. Applying the façade data annotation strategy (Frank et al., 2024c) to these images presents a significant technical challenge. Comparable datasets and approaches do not consider such a high level of detail. The labeling process began with the manual annotation of images using CVAT (Sekachev et al., 2020). Manually labeling a single image with polygonal instance segmentation for 39 façade element classes takes, on average, 3 to 4 hours. In the worst case, annotating 500 images would take around 2,000 hours—equivalent to a full-time

job for an entire year. To ensure that a sufficient number of images could be annotated, we explored AI-assisted labeling tools. Several market-available tools, such as Roboflow Annotate and Label Studio, were tested. However, due to the complexity of the annotation task, these tools proved inefficient for three main reasons: 1. The AI tools did not offer a viable self-learning mechanism. 2. Classification often still had to be performed manually. 3. The costs were prohibitively high and unaffordable for a research institute. During the same period, GroundingDINO (Liu et al., 2024) and the Segment Anything Model (SAM) (Kirillov et al., 2023) were released. Consequently, the decision was made to develop our own AI-based labeling tool by integrating CVAT, SAM, and YOLOv8-world (Cheng et al., 2024) (see Subsection 3.3). In several iterations, the images were labeled while simultaneously training the object extraction toolchain. On average, AI-assisted labeling of a single image took at least 45 minutes. Additionally, recorded IMU, gimbal, and GNSS+RTK data were processed (Frank et al., 2024b). This provided a preliminary absolute 6DoF position and pose of the camera, which was further refined using SIOCP see section 3.5.

3.3 Object Extraction in Images based on Yolo and Segment-Anything Model

The object extraction as façade instance segmentation is one of the core components of this process. The circumstances can be described as gaps between our 39 façade element classes, related datasets, labeling time, training time, and the AI precision required to achieve a sufficient level of automation. Based on these circumstances, the following requirements were derived: To meet the challenge of detailed object extraction under the given conditions, pre-trained AI models must be taken into account. Regarding object classification and labeling in CVAT, an open-vocabulary detection model should be used as an interactor. To assist the manual annotation, we began with segmentation models. In a second step, the segmentation was enhanced with an object detection model to improve automation. For automation, the output of the object detection model serves as input for the segmentation model.

We started with the state-of-the-art segmentation AI tools released at the end of 2023 and in 2024. The most recent one was Meta’s Segment Anything Model (SAM) (Kirillov et al., 2023). SAM is a pre-trained, closed black-box model with data interfaces that can be run offline, but it is not trainable. Meta offers a free demo application with a ’segment everything’ function available at https://segment-anything.com/demo. Its out of the box ’segment everything’ function on our dataset and some inner-city pictures did not match the required segmentation classes. However, the manual segmentation using mouse-click annotations was significantly accelerated by this model.

Ultralytics, in cooperation with Meta, released a trainable cut-down version of SAM. The model was fine-tuned with 50 and 100 annotated images, but it produced worse results than Meta’s original SAM. Thus, we decided to proceed with the original Meta SAM for segmentation.

For object detection, an open-vocabulary object detection model was necessary due to its text interface, which allows for extracting only specific classes. GroundingDINO (Liu et al., 2024) and YOLOv8-world (Cheng et al., 2024) identified as suitable candidates. However, running HuggingFace’s GroundingDINO in combination with SAM resulted in mismatched CUDA versions. Thus, running both models in parallel without code changes or a recompiled version was not an out-of-the-box solution. YOLOv8-world became the preferred alternative, as it worked out of the box and had better community support. We started with the largest HuggingFace YOLOv8x-worldv2 model, which has a mean average precision (mAP) of 47.1. For training YOLOv8, we split the 12MP (3040x4032) images into 1024 patches (5x4) with an overlap of 218 pixels in the x-direction and 264 pixels in the y-direction. Additionally, we enriched the model with our object classes and trained it. The training and test data were split in an 80/20 ratio, as many unknown façade element classes could not be inherited or related to other model classes. The model was trained for 500 epochs, and the results are shown in the results section. The YOLOv8 results were promising, so we integrated YOLOv8 as the detector in CVAT.

The YOLO output was then passed as input to the SAM model for automated instance segmentation. Several options are available for this integration: passing the bounding boxes (BB), tilted bounding boxes, or points for SAM instance segmentation. Results are shown in the results section. Finally, the relative pose of the recognized object is critical for successful instance segmentation. Objects parallel to the camera image plane are easier to segment than tilted ones.

3.4 Edge and Line Extraction

The edge and line extraction is used to obtain pixel-accurate line vectors from the images. Standard OpenCV (Bradski, 2000) methods, such as Canny edge detection, are applied to extract edges from 12MP RGB images. First, the image is blurred using a small 3x3 kernel to preserve important details. Afterward, Canny edge detection is performed without thresholds. The resulting black-and-white image of extracted lines is processed using the probabilistic Hough line transform (HoughLinesP). The parameters are set with a minimum line length of 10 pixels, a maximum gap of 3 pixels, and a threshold of 100, which delivered workable results for most images. HoughLinesP creates several small line vectors along a single interrupted diagonal pixel line. These interrupted lines are then padded using a nearest-vector clustering algorithm. The algorithm merges vectors that are aligned within a distance of 2 pixels and have an angular difference of less than 0.5°. Finally, the lines are filtered based on their length and connected vectors.

3.5 Scale-Invariant Object Contour Points (SIOCP)

The SIOCP algorithm combines edges, instance-segmented objects, LoD2 building models, and 6DoF camera pose and position to create scale-invariant object contour points (SIOCP).

In the first step, building orientation and shading are estimated. Germany has digitalized all buildings in LoD2, which allows us to determine the expected range of vertical building lines in the images. This is done by transforming the LoD2 building models into a bird’s-eye view and ray tracing their visibility from the GNSS-RTK position and the offset vector to the camera in a 360° view. The range of view is limited by the angular resolution of the 16 mm lens and the Sony IMX477 image sensor, and to maintain an error of less than 3 mm, the detection distance is set to 50 m. The estimated edge lines are sorted into vertical and horizontal lines. The strategy is to use vertical lines first to compensate for compass drift and adjust the horizontal points by aligning the image plane with a parallel ground offset between the images. Horizontal building edges, seen from the grounded camera’s point of view, create vanishing points. These vanishing points are used to estimate and check the plausibility of horizontal façade element parts by comparing vector

orientations. The spatial orientation of each building in relation to the camera's 6DoF pose can be estimated in this step. Additionally, the sun vector in 3D space is calculated, which can later be used for exact object extraction.

The following parts of SIOCP are theoretical strategies and still under development. Thus, there are no results yet, but they will be briefly discussed later.

In the next step, the contours of instance-segmented façade elements are refined using extracted lines. The approach considers the hierarchical relationships between façade elements, starting from outer building walls and moving to detailed elements such as window frames, casements, glass panes, and handles. These dependencies reflect how most elements are integrated into the façade and provide additional information about the probable shape and edge locations. The 3D orientation and sun vector are used to calculate shading, which helps estimate color gradients. Practical shading caused by elements such as balconies or adjacent walls is also considered. For example, rain gutters and pipes are typically round-shaped, so their color gradient depends on the camera's 6DoF pose and the sun vector. All this information is combined to match the lines with the object contours. Sharp known edges of each element are refined at the sub-pixel level to improve their accuracy. The corrected contour lines are then used to extract scale-invariant object contour keypoints. These keypoints are located at orthogonal irregularities along the contour, such as sharp edges or the midpoints of circular shapes. Each keypoint is described by its vector, the 6DoF camera pose and position, and its object class.

Keypoint tracking across images taken from different 6DoF positions is performed by matching their rays and relative positions on the embedded façade surface. A combination of brute-force and closest-point matching is used. Finally, the best keypoints from two images are selected for 6DoF camera pose reconstruction. Keypoints covering the largest area in both images are used to reconstruct the 6DoF camera pose and position via a perspective-n-point solver.

3.6 Object Segmentation with SuperCluster

We used SuperCluster (Robert et al., 2024), a panoptic segmentation architecture (Kirillov et al., 2019) based on SPT, which efficiently processes large point clouds. The method derives both, semantic and instance labels, with the latter being particularly relevant for us. However, the fact that both contribute to a common loss function leads to a harmonious understanding of the scene. Initially, the method computes hand-crafted features per point to build base-level superpoints. Since these represent the highest possible resolution of the point cloud, it is crucial to find an initial partition that accurately reflects object boundaries. To achieve this, a search technique must be applied to find a parametrization consistent with the featured data and scene. Using parameters based on the DALES dataset, and without adjusting them, we explored the model on data from an agricultural building obtained via mobile laser scanning. The dataset contains nine façade sections, each approximately 20 meters wide, manually labeled into semantic classes. Not all classes are represented in each individual section. We trained SuperCluster for 2000 epochs without pre-trained weights due to incompatible number of classes, using a train/test split of 80/20 %.

4. Results

In this section, we present a subset of the achieved methodology and its functional components. The functional components include Data Annotation, Object Extraction, Line Vector Extraction, and Object Segmentation using SuperCluster.

4.1 Data Annotation and Preparation

Annotation was and remains an ongoing task for this methodology. At this point, 200 images have been annotated at the pixel level to generate ground truth data (Frank et al., 2024c) from the Campus Friedrichshafen, Germany. Table 1 shows the occurrences of façade elements in 200 instance-segmented and annotated images. Glass elements are very common because of the modern architecture of one of the campus buildings. The other campus buildings are 1930s barracks that were renovated in 2010. A notable is the high proportion of glass elements compared to walls, with a 1:3 ratio in the façades. The results of the automated annotation are presented in the next subsection.

Class	Amount
Communication Technology	11
Bell	7
Mailbox	12
Handle	44
Support	93
Stairs	12
Door	47
Living Beings	18
Railing	167
Sky	49
Darkening	178
Window sill	297
Glass	1455
Scaffold	34
Obstacle	274
Wall	592
Roof	96
Frame	520
Lighting & Lamps	82
Ventilation/Heating/Climate	65
Text & Graphics	154
Traffic Sign	104
Pipe	181
Casement	466
Window	327
Road	110
Walkable	156
Vegetation	186
Vehicle	92
Ground	149
Fence, Wall	41

Table 1. Façade elements and their occurrences in 200 images

4.2 Object Extraction in Images based on Yolo and Segment-Anything Model

The object extraction subsection is divided into SAM training, YOLO training, and automatic annotation combining YOLO and SAM. Training and validation are separated by an 80:20 ratio. This ratio was selected due to the limited quantity of images.

The HuggingFace SAM model was fine-tuned using 100 images over 50 epochs. Figure 2 shows that the model begins to overfit after 5 epochs of training. The pre-trained and fine-tuned HuggingFace and Meta SAM models were compared in CVAT during manual labeling. The fine-tuned HuggingFace model produced weak results due to overfitting and masking overly large areas. Compared to the Meta model, the HuggingFace model also had issues with shading and edge detection when

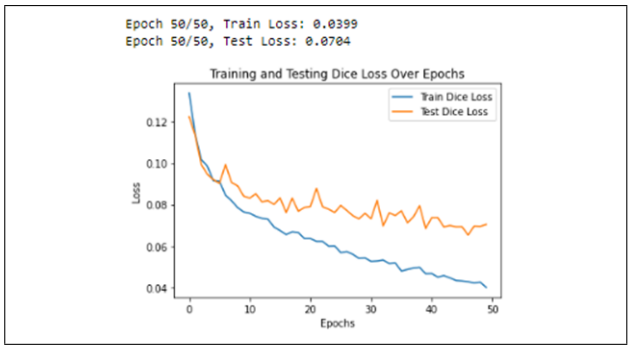


Figure 2. Overfiting Huggingface SAM model

clicking around the image. Due to the significant gap in performance between the two models, no further scientific metrics or comparisons were conducted. Finally, the decision was made to use the Meta model.

The pre-trained YOLOv8-world weights were enhanced with façade element classes and trained for 500 epochs. Figure 3 visualizes the training results over the epochs. The downward

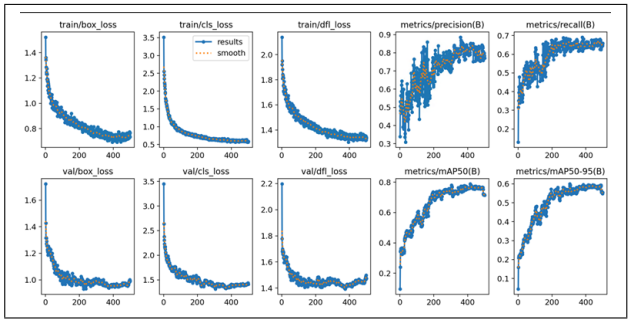


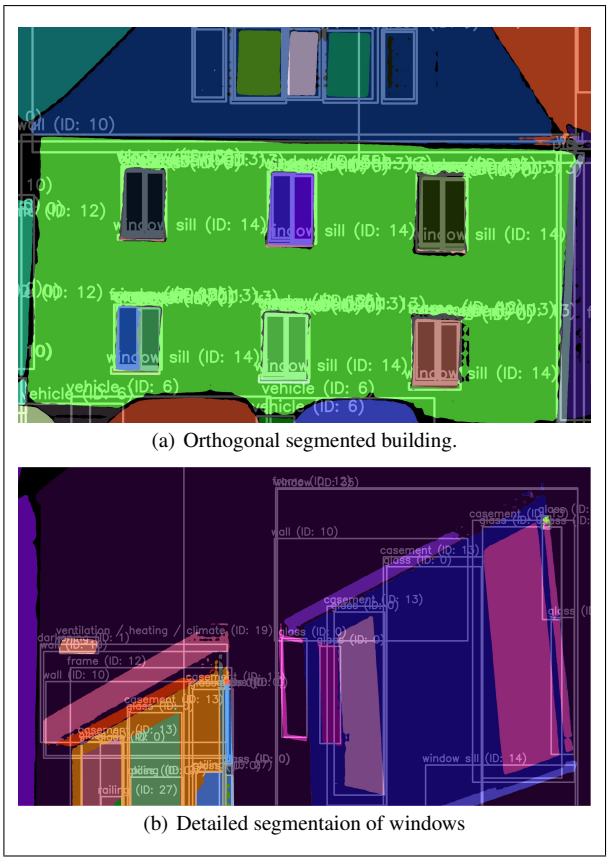
Figure 3. Resluts of YOLOv8 during training after 500 epochs.

trend of box and class loss in both training and validation indicates that the model is learning. However, towards the end, the model shows slight overfitting, as seen in the focal loss. The mAP50-95 value is within the range achieved by the model on the COCO dataset overall.

The results of the automatic instance segmentation pipeline, combining YOLOv8 and SAM, are shown in Figure 4. The visualization displays the overlay of YOLO bounding boxes and SAM's segmentation results. The stacked bounding boxes around the segmented building windows may appear confusing, but they result from the detailed segmentation of openable parts such as frames, casements, and glass. In the upper subfigure (a), an orthogonal and closed building façade is segmented, where the pipeline performs appropriately. Subfigure (b) shows an image with an askew camera view of windows. Here, YOLO misclassifies some shading elements as walls because the shading looks like a balcony. Metrics compared to the ground truth have not yet been calculated, because the pipeline requires further adaptation and refinement. More details can be found in the discussion section.

4.3 Edge and Line Extraction

The results of the edge and line extraction methodology are shown in Figure 5. The image is converted to greyscale for better line visualization. The original RGB image presents a challenge for the algorithm due to alternating light and shadow



(a) Orthogonal segmented building.

(b) Detailed segmentaion of windows

Figure 4. Instance segmentation results from an automated pipeline of YOLOv8 and SAM.

conditions. The extracted line vectors accurately overlay the edges. However, some lines are shorter than the actual edge lengths. This is caused by the Canny edge detection and the padding algorithm. Canny edge detection creates interruptions in the edges, which are subsequently padded. The figure shows twice padded line vectors, where the second padding filter criteria is more accurately.

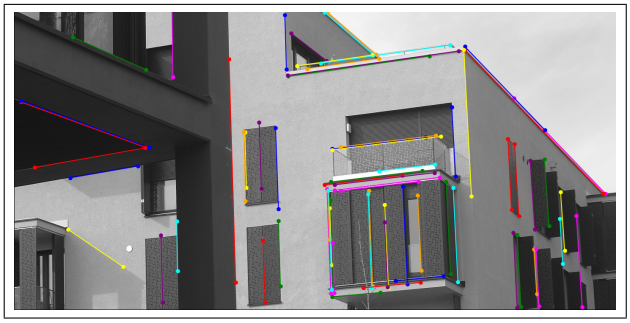
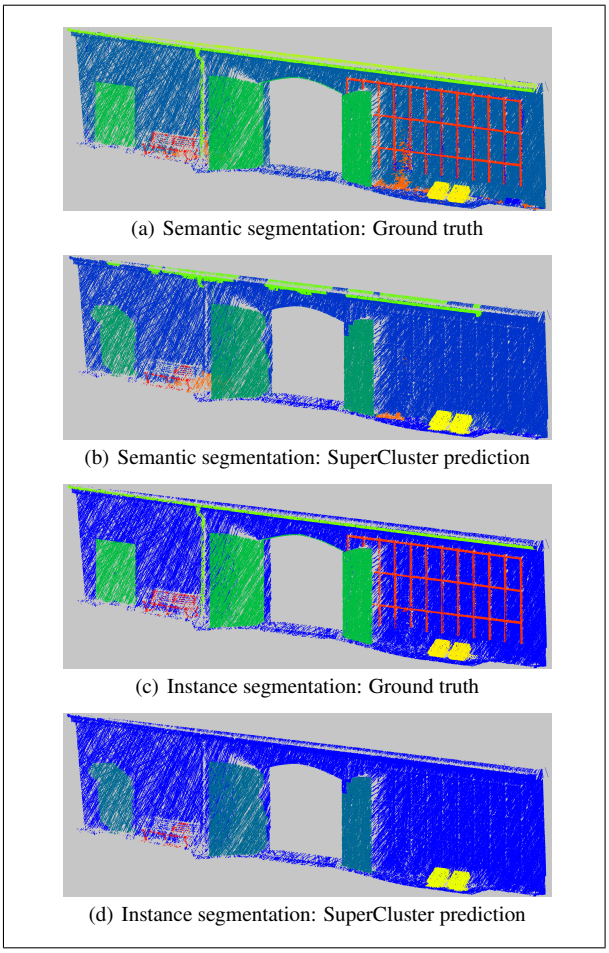


Figure 5. Greyscale image of a building with challenging sunlight and shadows, overlaid with extracted lines.

4.4 Object Segmentation with SuperCluster

Figure 6 shows the semantic and instance segmentation results for a sample from the test set. As the training only took place on very little data, the results are mediocre. For example, Figure (d) shows that the rain gutter is not recognized in the instance segmentation. We expect improved accuracy with more available training data and additional effort in parameter optimization. Additionally, since the utilized point clouds are dense, we

assume that SuperCluster will perform equally on photogrammetric data.



(a) Semantic segmentation: Ground truth

(b) Semantic segmentation: SuperCluster prediction

(c) Instance segmentation: Ground truth

(d) Instance segmentation: SuperCluster prediction

Figure 6. Results of semi automated labeling with SuperCluster.

5. Discussion

The presented methodology for LoD3 reconstruction is still under development. Nevertheless, we present a methodology as a proof of concept. The **dataset** used covers only a small portion of building and architectural types for reconstruction, resulting in a lack of data variation for training and testing. This indicates that the dataset must be expanded to include a wider range of building types. Furthermore, the data was recorded during the summer on sunny days because the recording handcart is not rainproof. This results in a best-case scenario for the instance segmentation approach. The dataset should also be expanded to include different weather conditions and recording times for greater robustness. Additionally, the limited number of 200 annotated images is not an ideal starting point for improving object detection and segmentation. While there is no fixed ”magic number” for AI training, various forums suggest using at least 1,000 images from different perspectives, object scales, and times of day to ensure robustness.

The **object extraction** is a core component of the reconstruction pipeline. From a research perspective, the investigation into the overfitting of the Ultralytics **SAM** did not make sense. To achieve successful fine-tuning, a significant amount of data would be required, which would exceed our time constraints. Furthermore, the comparison between Meta SAM and Ultralytics SAM was based on subjective human perception during annotation in SAM and output comparison. In our opinion, the pre-trained Meta model performed much better, which is why we did not conduct a statistical evaluation. **YOLOv8** world was trained on 160 annotated images and showed slight overfitting towards the end of the training process. Considering the limited number of images, the performance on 40 test images was comparable to the COCO dataset level. However, its performance in shaded regions and areas with low color transitions was weaker. Additionally, YOLOv8 exhibited similarity issues with shaded coverings, misclassifying them as balconies or multiple walls due to their appearance. Further, it lacks logical information, such as recognizing that a balcony is larger and typically has a door on its upper side. A potential solution could be to enhance the model through feature fusion, incorporating information from the sun vector, ego camera vector, façade orientation vectors, and illumination data. The pipeline passing bounding boxes from **YOLOv8 to SAM** delivers acceptable results for façade elements positioned in the orthogonal image plane. However, when applied to tilted or non-orthogonal façade elements, the results deteriorate. The main issue is the mismatch between distorted façade elements in the image space and the rectangular bounding boxes. A possible solution could be to determine the bounding box and extract line vectors (see section 3.4), then pass the inner closest points to SAM. This highlights the potential for optimization by integrating additional features and external information.

The extraction of **lines and edges** produces good vectors in a respectable number of images. However, line vanishing effects occur due to blurring with a 3x3 kernel or color gradients, resulting in interrupted lines. A potential solution could be to process the unblurred image using Canny edge detection with smaller bandwidth criteria. Another issue arises during vectorization with OpenCV’s ‘HoughLinesP‘, which generates many small line vectors that need to be padded afterward. Due to the padding process, some vectors are lost, especially at the beginning and end of an edge (see Figure 5). A possible improvement would be to use an alternative vectorization method, such as a snake algorithm, which performs vectorization and padding in a single step. Due to the limited number of annotated images in the dataset, the extraction algorithm has not been thoroughly tested. One option is to test the method using the Barcelona Images for Perceptual Edge Detection (BIPED) dataset (Soria et al., 2023).

The **SIOCP** integration is partially implemented but not yet fully operational. As a result, there are currently no results to present.

6. Conclusion and Outlook

The paper presents a new methodological approach for monocamera 6DoF position and pose reconstruction in RGB images, based on SIOCPs with building façades. Additionally, the extracted data can be used for photogrammetric LoD3 reconstruction. Our achievements include:

- An instance-segmented façade dataset with 39 element classes and 200 images was presented, and its annotation remains an ongoing task.
- The methodology introduces an approach for scale-invariant object contour points (SIOCP) of façade elements for 6DoF

mono-camera pose and position reconstruction, as well as photogrammetric point cloud-based segmentation.

- The YOLOv8-World and SAM pipeline assists the complex instance annotation process, reducing manual effort by more than 75%.
- Image preprocessing components, including YOLO, SAM, and edge extraction, are functioning appropriately and are ready for integration into the SIOCP framework.
- Matched SIOCP keypoints or photogrammetrically generated point clouds can be used for LoD3 building façade reconstruction.

Nevertheless, the implementation and cross-referencing of the entire methodology is still an ongoing task. Several necessary improvements, listed in the results section, must be made to ensure the complete pipeline works reliably in an automated way. The presented approach is a step toward simplifying LoD3 building reconstruction. Our vision is to generate LoD3 buildings using a handheld device, such as a smartphone.

References

Bradski, G., 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Cheng, T., Song, L., Ge, Y., Liu, W., Wang, X., Shan, Y., 2024. YOLO-World: Real-Time Open-Vocabulary Object Detection. *arXiv preprint arXiv:2401.17270*.

Frank, F., Buckel, P., Hoegner, L., Hofstedt, P., 2024a. A Landmark Selection Method for Object-Based Visual Outdoor Localization Approaches of Automated Ground Vehicles. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, X-4/W5-2024, 163–169. https://isprs-annals.copernicus.org/articles/X-4-W5-2024/163/2024/.

Frank, F., Hoegner, L., Buckel, P., Dalm, K., 2024b. An alternative raw data acquisition approach for reconstruction of lod3 models. T. H. Kolbe, A. Donaubauer, C. Beil (eds), *Recent Advances in 3D Geoinformation Science*, Springer Nature Switzerland, Cham, 459–477.

Frank, F., Richter, R., Hoegner, L., 2024c. A concept for methodical classification, assignment of attributes, and labeling of facade elements in camera images for LOD3 reconstruction of buildings. *DGPF Jahrestagung 2024, Band 32*, Deutsche Gesellschaft für Photogrammetrie, Fernerkundung und Geoinformation (DGPF), Remagen, Germany, 181–190.

Graham, B., Engelcke, M., Van Der Maaten, L., 2018. 3d semantic segmentation with submanifold sparse convolutional networks. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 9224–9232.

Harshit, Chaurasia, P., Zlatanova, S., Jain, K., 2024. Low-Cost Data, High-Quality Models: A Semi-Automated Approach to LOD3 Creation. *ISPRS International Journal of Geo-Information*, 13(4), 119. https://www.mdpi.com/2220-9964/13/4/119.

Huang, H., Michelini, M., Schmitz, M., Roth, L., Mayer, H., 2020. LOD3 BUILDING RECONSTRUCTION FROM MULTI-SOURCE IMAGES. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLIII-B2-2020, 427–434. https://isprs-archives.copernicus.org/articles/XLIII-B2-2020/427/2020/.

Kirillov, A., He, K., Girshick, R., Rother, C., Dollár, P., 2019. Panoptic segmentation. *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9404–9413.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., Girshick, R., 2023. Segment anything. *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, 3992–4003.

Landrieu, L., Simonovsky, M., 2018. Large-scale point cloud semantic segmentation with superpoint graphs. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4558–4567.

Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Jiang, Q., Li, C., Yang, J., Su, H., Zhu, J., Zhang, L., 2024. Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection. https://arxiv.org/abs/2303.05499.

Nan, L., Wonka, P., 2017. Polyfit: Polygonal surface reconstruction from point clouds. *Proceedings of the IEEE International Conference on Computer Vision*, 2353–2361.

Pantoja-Rosero, B., Achanta, R., Kozinski, M., Fua, P., Perez-Cruz, F., Beyer, K., 2022. Generating LOD3 building models from structure-from-motion and semantic segmentation. *Automation in Construction*, 141, 104430. https://linkinghub.elsevier.com/retrieve/pii/S092658052200303X.

Qi, C. R., Su, H., Mo, K., Guibas, L. J., 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*, 652–660.

Qi, C. R., Yi, L., Su, H., Guibas, L. J., 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in neural information processing systems*, 30.

Robert, D., Raguet, H., Landrieu, L., 2023. Efficient 3d semantic segmentation with superpoint transformer. *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 17195–17204.

Robert, D., Raguet, H., Landrieu, L., 2024. Scalable 3d panoptic segmentation as superpoint graph clustering. *2024 International Conference on 3D Vision (3DV)*, IEEE, 179–189.

Schwab B., Wysocki O., 2021. LoD3 Road Space Models. https://github.com/savenow/lod3-road-space-models.

Sekachev, B., Manovich, N., Zhiltsov, M., Zhavoronkov, A., Kalinin, D., Hoff, B., TOsmanov, Kruchinin, D., Zankevich, A., DmitriySidnev, Markelov, M., Johannes222, Chenuet, M., a andre, telenachos, Melnikov, A., Kim, J., Ilouz, L., Glazov, N., Priya4607, Tehrani, R., Jeong, S., Skubriev, V., Yonekura, S., vugia truong, zliang7, lizhming, Truong, T., 2020. opencv/cvat: v1.1.0. https://doi.org/10.5281/zenodo.4009388.

Soria, X., Sappa, A., Humanante, P., Akbarinia, A., 2023. Dense extreme inception network for edge detection. *Pattern Recognition*, 139, 109461. https://www.sciencedirect.com/science/article/pii/S0031320323001619.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need. *Advances in Neural Information Processing Systems*.

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., Solomon, J. M., 2019. Dynamic graph cnn for learning on point clouds. *ACM Transactions on Graphics (tog)*, 38(5), 1–12.

Wysocki O., Schwab B., Willenborg B., 2022. Awesome CityGML. https://github.com/OloOcki/awesome-citygml.

Wysocki, O., Xia, Y., Wysocki, M., Grilli, E., Hoegner, L., Cremers, D., Stilla, U., 2023. Scan2LoD3: Reconstructing semantic 3D building models at LoD3 using ray casting and Bayesian networks. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, IEEE, Vancouver, BC, Canada, 6548–6558.