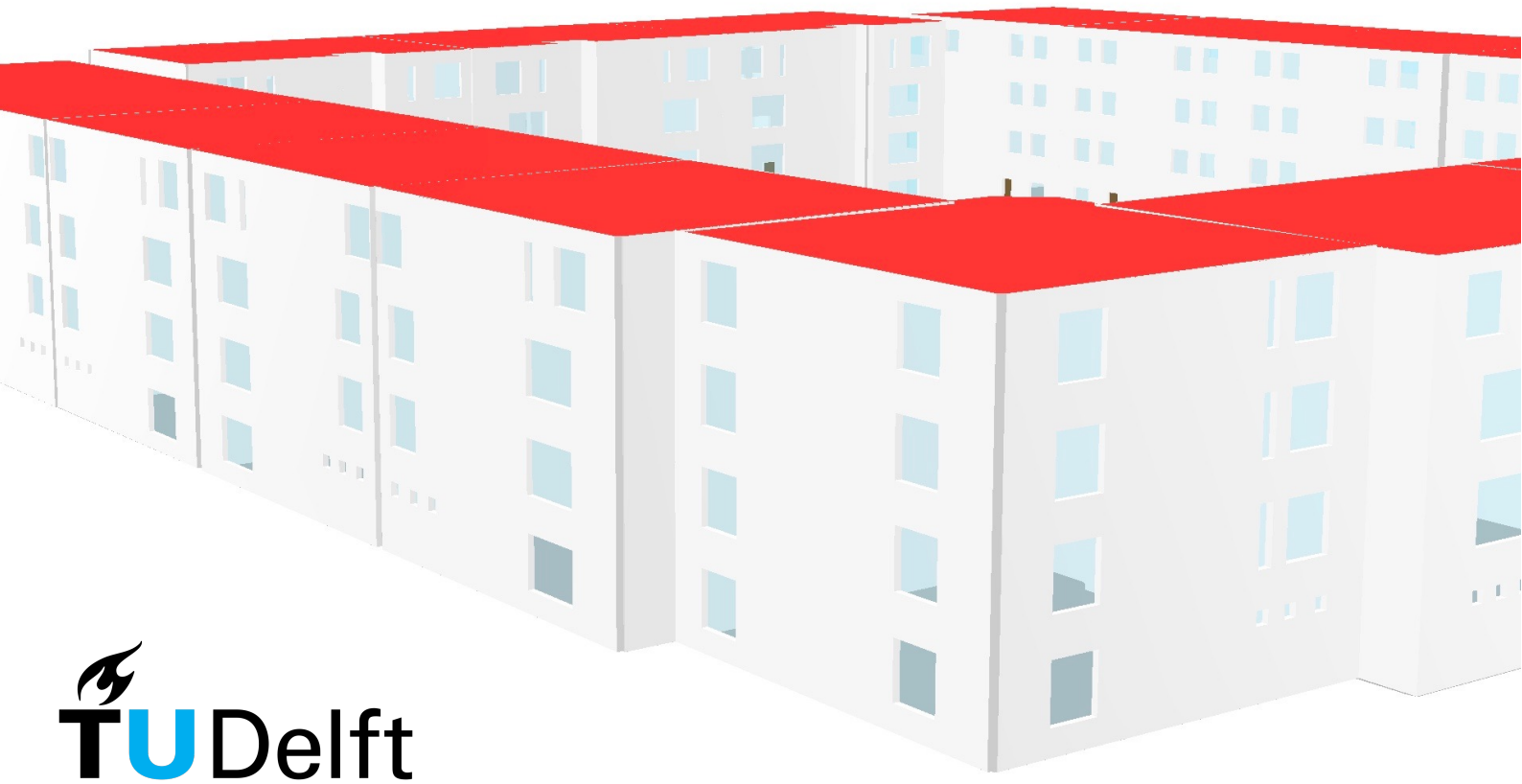
MSc thesis in Geomatics

A data driven approach to add openings to 3D BAG building models

Yitong Xia

2023



TU Delft

MSc thesis in Geomatics

A data driven approach to add openings to 3D BAG building model

Yitong Xia

June 2023

A thesis submitted to the Delft University of Technology in partial fulfillment of the requirements for the degree of Master of Science in Geomatics

The work in this thesis was carried out in the:

3D geoinformation group
Delft University of Technology

Supervisors:	Prof.dr. Jantien Stoter
	Ir. Weixiao Gao
Co-reader:	Dr. Ken Arroyo Ohori

Abstract

The reconstruction of 3D city models has garnered significant interest in recent years. However, the majority of existing reconstruction methods primarily focus on LOD2 models, while LOD3 model reconstruction often relies on manual labor, and the primary data sources are street view images. This research aims to advance this field by reconstructing LOD3 models through the addition of windows and doors to existing LOD2 models, thereby maximizing the utility of available 3D building models, as well as the accurate addition of windows and doors. This research innovatively utilizes aerial oblique images as the data source for extracting building openings and employs 3D BAG LOD2.2 models as the basic 3D building structures. The 3D facades are projected onto the 2D aerial image space using perspective projection and registration is employed on the projection facade and oblique aerial images. Subsequently, Mask R-CNN is employed to detect and extract the building openings from these projections. Following the extraction, the layout of the openings within the same facade is optimized in terms of both size and position. Lastly, the relative positions of the openings on the facade images are combined with the 3D coordinates of the corresponding facade to calculate the positions of the openings in 3D space. This information is then integrated into the LOD3 model, resulting in a more detailed and accurate representation of the buildings.

This approach successfully reconstructs the final LOD3 model in CityJSON format, which passes the val3dity validation. By effectively utilizing existing 3D building models, this approach conserves a considerable amount of computational resources required for reconstruction. The simplicity and high level of automation of this approach make it a promising solution for reconstructing large-scale LOD3 buildings, leading to more accurate and detailed large 3D urban models.

Acknowledgements

I wish to express my deepest gratitude to my supervisors, Dr. Jantien Stoter and Weixiao Gao, also my co-reader Ken Arroyo Ohori and the delegate Liangliang Nan. The environment they created, which was characterized by consistent positivity, encouragement, and guidance, was instrumental in the completion of my project. The journey was fraught with challenges, yet their unwavering support continually instilled the confidence in me to persevere.

My heartfelt appreciation also extends to my classmate, Fengyan Zhang, whose assistance and valuable insights significantly enriched my first year of group work. His exceptional partnership has left an indelible mark on my academic journey.

These two years have unfolded like an unforgettable journey, with my parents and my grandparents standing as stalwart supporters. Their enlightened perspective and forward-thinking approach created the opportunity for me to study abroad and encounter diverse life experiences. Their unfailing comfort in my times of fatigue has been my source of strength. It fills me with great pride to be their child, and words fall short to convey my deep-seated gratitude. Even though my grandfather has become a star in the galaxy, I firmly believe that his pride in my accomplishments outshines all others.

I also wish to extend my sincere thanks to my supporters, Guigui Zhang, Mama Tong, Huaban Zhang, and Guoguo Deng. Their relentless encouragement was a beacon of hope that propelled me to see this journey to the end. I wish them joy and happiness as they navigate their paths across the world, no need to be the most useful person, but be the happiest.

My affectionate thanks go to my kittens, Mimi, Bubbletea, Biabia, and Pudding that I lost, who have been my oasis of calm in periods of stress. Despite our linguistic differences, they have provided immense comfort.

Lastly, I am profoundly grateful for Yunhao's companionship during my time in Europe. Your stable emotional presence and unwavering support during my Master's and PhD applications served as the pillars upon which I learned to navigate the pressures of academic life. I look forward to our continued mutual support as we journey through life together.

Contents

List of Figures

List of Tables

Acronyms

1 Introduction

1.1 Background and motivation

The rapid expansion of urbanization is one of the results of global industrialization, and the global trend of urbanization has become obvious as more and more people are living in urban areas. To deal with the complex situation, it has become crucial to improve the living environment in densely populated urban areas, by planning for more sustainable urban development. The concept of Smart Cities is an important approach to improving the quality of life of the growing urban population, which is defined as using Information and communication technologies (ICT) to make a city (administration, education, transportation, etc.) more intelligent and efficient [Su et al. [2011]]. Smart cities help public authorities to know what is happening, where, and when.

An important component of Smart Cities is the integration of data from a variety of sources (e.g. remote sensing images, point cloud, and digital imagery) through spatial analysis and visualization in order to obtain sustainable and city-specific development solutions. Thus, there is an increasing need for 3D city models accurately, comprehensively, and appropriately represent buildings. A 3D city model is a digital representation and simulation of the urban environment using three-dimensional geometry [Batty et al. [2001]; Singh et al. [2013]; Peters et al. [2022a]], that includes buildings, rivers, vegetation, bridges, etc. It provides a detailed and accurate representation of the urban environment, which can be used to support urban planning and decision-making activities.

The utilization of 3D city models is supported for a variety of reasons. First, compared to traditional 2D data, 3D city models can not only replace the 2D data in the majority of Geographical Information Science (GIS) use cases but can also imitate realistic surroundings more precisely than 2D data, increasing the reliability of the findings and their interpretation [Biljecki et al. [2015]]. 3D city models offer a consistently accurate and photorealistic virtual representation of real-world scenes in urban areas [Willenborg et al. [2018]]. There are currently 3D city models for several countries and cities throughout the world, for instance, 3D BAG covering the entire Netherlands [Peters et al. [2022b]], digital twin and 3D models in Helsinki, Finland [Leberl et al. [2010]], 3D Berlin, 3D Munich in Germany, etc. 3D BAG is an open dataset containing 3D building models of the entire Netherlands, and covering various Level of Detail (LOD), including LOD0, LOD1.2, LOD1.3, and LOD2.2. The aim of the 3D BAG is to create precise and up-to-date 3D building models for urban applications under the open license.

As a major component of the city, the simulation and visualization of buildings are of great interest. The OD3 building models, owing to their detailed representation of architectural elements, significantly enhance the accuracy and realism of urban visualizations, simulations, and various types of urban analysis. The generation technology for 3D building models has been developed to a relatively advanced level, especially LOD1 and LOD2 models. Even so, the details of the reconstructed 3D building models are not perfect because it is challenging to obtain key building elements, like the doors and windows on the façade. The LOD3 model is more challenging to construct than the LOD2 building model since it has more complex building elements. The current LOD3 models can be generated with the following methods: by dense LiDAR point clouds [Akmalia et al. [2014]; Leberl et al. [2010]], by imagery [Pantoja-Rosero et al. [2022]; Huang et al. [2020]], and by Building Information Model (BIM) models [Geiger et al. [2015]], and finally by combining multiple sources of data to extend the LOD2 model into LOD3 [Zhang et al. [2019]; Gruen et al. [2019]].

In addition to LiDAR point clouds, the imagery also contains a rich set of building-related information (including windows and doors). With the advancement of sensors and the platforms on which they are mounted, there are several airborne, terrestrial, and mobile image datasets available, such as Street View Images (SVI) and airborne oblique images, from which it is possible to extract LOD3 model components like windows and doors. As a result, some researchers have explored using rich building information found in different imagery to construct LOD3 models, and this approach works well. The current widely used method is the photogrammetry-based method, which generates dense point clouds using Structure from Motion (SfM) and Multi-View Stereo (MVS) and reconstructs 3D models. But there are also limitations to the existing LOD3 building model reconstruction techniques, such as the need for manual labor throughout the process [Zhang et al. [2019]; AlHalawani et al. [2013]; Nan et al. [2010]], and the reconstruction is only effective for buildings with regular openings distribution [AlHalawani et al. [2013]], while the reconstruction method is not applicable when the distribution of the openings is irregular.

The current LOD2 building model is not always used to its full potential in the majority of the existing LOD3 reconstruction techniques, and LOD2 is not always extended to LOD3. In this research, we consider how can we use the data we already have (3D BAG LOD2.2 building models and oblique aerial imagery) to generate the LOD3 building model dataset. This research proposes a pipeline to apply the photogrammetry-based technique to extract 2D façade texture images and use the deep learning model to detect openings from the façade texture images. Then we optimize the layout of the façade openings using a regularization algorithm in 2D space, transform the 2D openings into 3D ones, and integrate them into the existing 3D BAG LOD2.2 building model and form a recessed window.

1.2 Research objective identification

1.2.1 Research objectives

Considering the current status of the 3D City model development and its limitations, the main question of this research is:

How to upgrade the 3D BAG LOD2.2 building model to LOD3 by extracting openings information from oblique aerial images?

In this research, we introduce a novel pipeline designed to generate LOD3 building models by leveraging the power of photogrammetry and deep learning techniques. Specifically, the proposed approach employs photogrammetry-based techniques and least square regression to extract façade texture images from oblique aerial images, and Mask R-CNN to detect and extract openings from the texture images. These extracted openings are then seamlessly integrated into the 3D BAG LOD2 building model, ultimately resulting in the desired LOD3 model. This streamlined process allows users to obtain LOD3 building models directly, without the need for manual intervention or manipulation, while maintaining the simplicity and accessibility of input data. This innovative methodology not only improves the efficiency of generating LOD3 building models but also ensures a higher degree of accuracy in the final output.

The research can be subdivided into the following sub-questions:

- How to identify the individual façade texture image of each 3D façade from oblique aerial images, and maximize the number of extractable façades?
- How to address the systematic errors between 3D BAG and oblique aerial images using data registration?
- How can openings be detected and extracted from façade texture images?
- How to optimally integrate extracted 2D openings with 3D building models?

1.2.2 Innovations

This research introduces several innovative approaches in comparison to existing research. Firstly, it utilizes oblique aerial images as the primary data source for extracting building openings, which is a departure from the majority of studies that typically rely on SVI. Secondly, this research effectively overcame systematic errors between the LOD2 building models and oblique aerial images, by using least square regression to calculate the linear offset between them, to achieve the best integration. Lastly, unlike other studies that directly compute the 3D coordinates of openings using photogrammetry-based techniques, this paper presents a unique approach that involves projecting 2D façade information back into 3D space. These advancements extend the available data sources for the LOD3 model generation process, making LOD3 model generation faster and easier, while most of the

current studies focus on the LOD3 model reconstruction of a single building, this method can achieve a larger scale LOD3 model reconstruction.

1.3 Thesis structure

This thesis consists of the following chapters:

- Chapter 2 presents the theoretical background of this research;
- Chapter 3 describes the related works of this research, two types of different data sources with multiple methods are introduced. Deep learning techniques in façade parsing and façade element layout optimization algorithms are also presented in this chapter;
- Chapter 4 introduces the whole proposed pipeline for generating LOD3 building models. Three stages including façade image extraction, openings detection and layout optimization, and final integration of openings and 3D building models are detailed illustrated. Finally, the metrics that measure the accuracy of the method are presented.
- Chapter 5 shows the implementation details of the pipeline, including the utilized libraries and software, and the parameter tuning involved in the pipeline, including the region growing algorithm, Mask R-CNN, and DBSCAN algorithm.
- Chapter 6 detailed presents the results and analysis for each step of the pipeline, and the universal applicability of the pipeline is demonstrated by running it on a new group of buildings.
- Chapter 7 gives a summary and review of the whole research, and gives answers to the research objectives and research sub-questions posed in Chapter 1. The contributions and limitations of this project and future works are also discussed.

1.4 Uniqueness of the research

The existing literature encompasses a range of methods on LOD3 model generation, 3D city model enhancement, and façade parsing [Dobson [2023], Apra [2022], Wang [2022]]. The uniqueness of this research is underscored by the following unique aspects:

- **Data Source Differentiation**: Unlike the data sources SVI adopted in previous studies, this research utilizes oblique aerial imagery as the primary source for image data, longer acquisition distances and lower resolutions make it more difficult to use.
- **Diverse source of 3D City Models**: The building model and texture images in the referenced research in Wang's research [Wang [2022]] stem from the same data source SVI. In contrast, this research employs distinct data sources for the 3D building model and texture images. Our approach obviates the need for regenerating 3D building

models, facilitating a faster and more applicable solution for introducing openings in a larger range of buildings.

- **Alternative 2D to 3D Conversion method**: Wang employs a back-projection technique to convert 2D openings into 3D ones [Wang [2022]] . This research instead leverages the principles of similar triangles for such calculations.
- **Different focus of building element**: Irène also focuses on 3D city model enhancement, and prioritizes the extraction of dormers, windows, and chimneys from orthophotos [Apra [2022]], while this research is geared towards the extraction of façades and openings.

Each of these elements defines the unique contribution of our research in the context of existing literature on LOD3 model generation and 3D city model enhancement.

2 Theoretical Background

This chapter gives the theoretical background of the research.

2.1 3D City Model

A 3D city model is a digital representation and simulation of the urban environment using three-dimensional geometry [Batty et al. [2001]; Peters et al. [2022a]; Singh et al. [2013]; Biljecki et al. [2015]] that includes buildings, rivers, vegetation, bridges, etc. They are constructed at multiple levels of detail to provide concepts of various resolutions and at different levels of abstraction Biljecki et al. [2015].

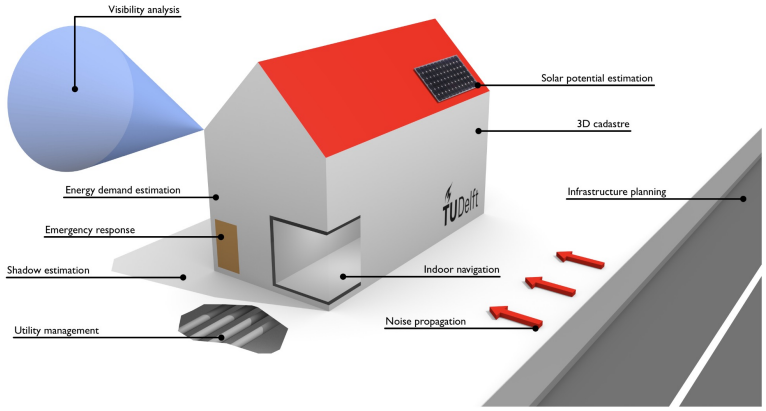


Figure 2.1: 3D city models (Figure from [Biljecki et al. [2015]])

The current 3D city models can be generated from multiple techniques and data sources, including the photogrammetry-based method [Singh et al. [2013], Bullinger et al. [2021]] and laser scanning-based method [Pu et al. [2006]], extrusion from 2D building footprints [Ledoux and Meijers [2011]], and architectural models, etc. Photogrammetry-based methods and laser scanning-based methods are by far the most mature techniques and they are both widely used. The advantages of the above techniques are that the reconstruction data are easy to obtain, and the reconstruction is highly automated and requires minimal human involvement. It can also realize large-scale automated reconstruction and obtain the 3D information of the city economically and efficiently [Pang and Biljecki [2022]].

The current researcher has defined 12 categories of 3D city model use cases: emergency services, urban planning, telecommunications, architecture, facilities and utility management, marketing and economic development, property analysis, tourism and entertainment, e-commerce, environment, education and learning, city portals [Batty et al. [2001]]. 3D city models can be classified into non-visualization use cases and visualization-based cases as well, and the current non-visualization use cases suggest that the role of 3D models has gone beyond visualization to more development and utilization areas [Biljecki et al. [2015]], now their analytical capabilities are becoming much more crucial.

2.2 Level of Detail

One of the important characteristics of the 3D city model is the LOD. LOD is a measure of how accurately a 3D city model has been created and how closely it adheres to the relevant subset of reality [Biljecki [2017]]. It is mostly used to characterize the geometric detail of a model, primarily of buildings, in the 3D GIS domain [Biljecki et al. [2016]]. The LOD between geographic data might vary depending on the nature of data, spatial scale, acquisition procedure, and other aspects [Biljecki et al. [2016]]. The detailed definition of LOD is shown in figure 2.2 and Table 2.1. It is worth noting that LOD3 is a lot more detailed model than LOD2 in this comparison. For many applications, the benefits presented in LOD3 are quite helpful. The creation of a LOD3 3D city model is always a worthwhile subject to address because LOD3 can play a greater role in many spatial analyses, including illumination analysis and heat-loss estimation.

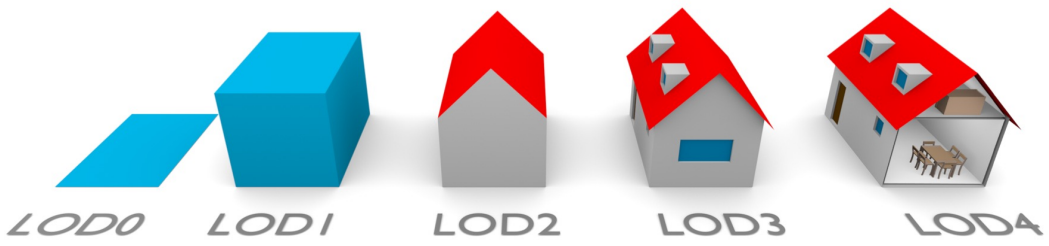


Figure 2.2: LOD specification (Figure from [Biljecki et al. [2016]])

2.3 Oblique aerial imagery

The oblique aerial images are taken at an angle by digital oblique cameras mounted on drones or aircraft. The digital cameras can be classified based on their configurations, and the current state-of-art system is the five-view camera system, which takes five images for

Table 2.1: LOD definition developed by OGC

LOD	Definition
LOD0	Buildings are represented by footprint or roof edge polygons.
LOD1	Buildings are represented as blocks model, comprising prismatic buildings with flat roof structures.
LOD2	Buildings have differentiated roof structures and thematically differentiated boundary surfaces.
LOD3	Buildings have detailed wall and roof structures potentially including doors and windows.
LOD4	LOD4 buildings complete LOD3 buildings by adding interior structures.

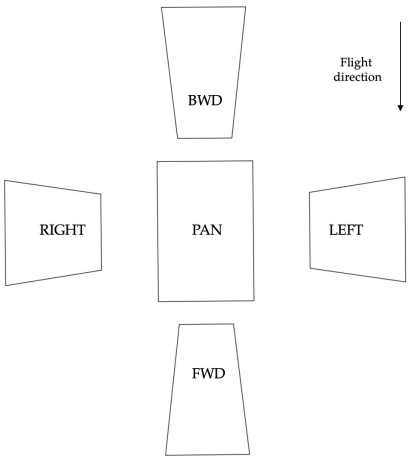


Figure 2.3: Five view camera system

every acquisition position, including the forward side, backside, left side, and right side of the flight direction. Onboard sensors autonomously measure camera extrinsic parameters during the imaging process, which facilitates the 3D reconstruction process.

The application of oblique imagery is diverse and expanding in civil and mapping fields [Remondino and Gerke [2015]]. The oblique imagery provides a new data source for 3D city models. Since there is a large overlapping area between oblique imagery and high resolution, it is often used for pixel-based multi-stereo image matching to reconstruct dense 3D point clouds. Automatic interpretation of these dense point clouds enables the generation of 3D city models, typically for LOD2 [Haala et al. [2015a]] (Figure 2.4). It is also used to improve the quality of the 3D city models by extracting façade textures from the [Yang et al. [2015a]]. There are also several typical application scenarios, such as urban change detection and post-damage management [Kakooei and Baleghi [2023]], cadastral management

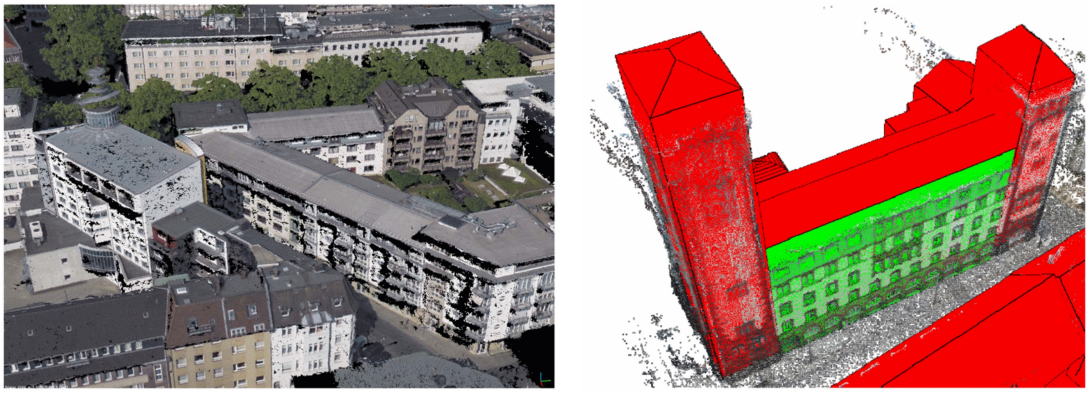


Figure 2.4: Dense matching point cloud (left) and 3D city model (right) (Figure from [Yang et al. [2015a]])

[Habbecke and Kobbelt [2012]], etc.

Utilizing digital angle cameras has a number of benefits. Oblique imagery offers the opportunity to observe ground features from multiple views, which more accurately depicts the features' actual conditions than orthophotos do. Oblique imagery can also show give a better, detailed interpretation of the buildings [Yang et al. [2015a]].

The optical axis must be deviating from vertical while taking the oblique aerial images [Haala et al. [2015b]]. As a result, oblique aerial imagery is generally suitable for 3D data collection at such structures since building façades are easily visible in these images.

3 Related work

The current classification of 3D building model reconstruction can be summarized in [Oniga et al. [2022]]:

- Level of automation: fully-automatic, semi-automatic, automatic with cadastral information;
- 3D reconstruction approaches: model-driven (bottom-up) approaches, data-driven (top-down) approaches;
- Data source: LiDAR data (Terrestrial Laser Scanning (TLS) and Airborne Laser Scanning (ALS)), imagery data (Unmanned Aircraft Systems (UAS) images, SVI images, satellite images), topographic data.

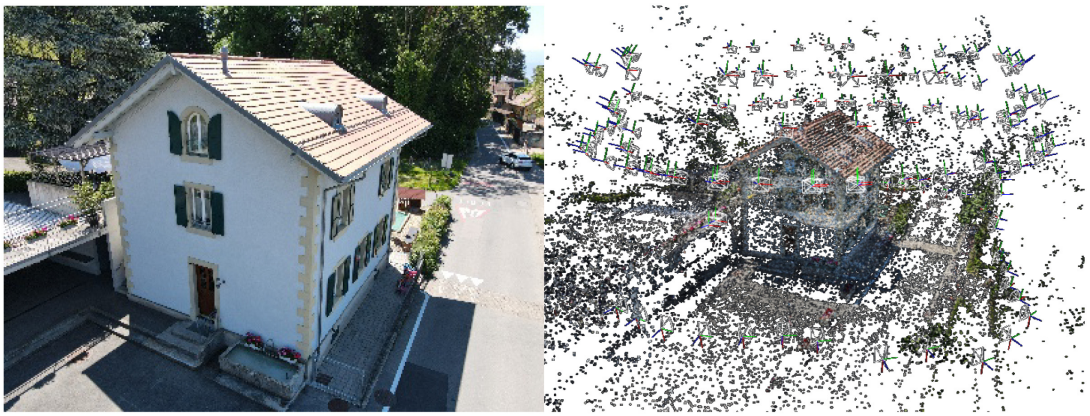
The bottom-up approaches involve reconstructing 3D building models by extracting and assembling fundamental geometric primitives, such as point clouds, 3D segments, and 3D planes, without incorporating any prior knowledge about the specific characteristics or features of the buildings being reconstructed. This technique relies solely on the data-driven identification and organization of primitive components to generate the final 3D model, which can lead to challenges in accurately representing complex architectural structures. On the other hand, top-down approaches utilize prior knowledge about buildings to guide the reconstruction process. This methodology reconstructs building models by selecting and adapting the most suitable candidate from a pre-defined library of known models, based on the degree of consistency with the input data. As a result, top-down approaches typically demonstrate greater robustness in generating accurate 3D building models compared to bottom-up approaches. However, this technique is inherently limited by the comprehensiveness and diversity of the prior model library, which may not encompass all possible building types or architectural variations, potentially restricting the applicability and versatility of the top-down approach in certain scenarios.

In this chapter, we will focus on presenting various methodologies for LOD2 and LOD3 building model reconstruction. The data sources of the reconstruction will be provided first. We will explore both bottom-up and top-down approaches, illustrating their applications and effectiveness across diverse data of scenarios. By examining these strategies in detail, this chapter aims to provide a thorough understanding of reconstructing 3D building models from various data inputs.

3.1 Data sources

Reconstructing a building model from raw data is a complex process. Generally, an OGC-standard LOD2 or LOD3 model cannot be reconstructed directly and usually requires several steps to generate. A semantic LOD2 or LOD3 model is basically generated from surface mesh, and the data sources of 3D surface mesh reconstruction include LiDAR point cloud data, imagery data, and topographical data.

For the imagery data, it is first necessary to generate a dense point cloud of the building object by a set of images of the same object taken from different angles by photogrammetry-based methods, such as SfM and MVS (Figure 3.2). The currently utilized image sources include satellite photogrammetry [He et al. [2023]; Li et al. [2021]; Bullinger et al. [2021]], SVI [Pang and Biljecki [2022],] panorama photogrammetry [Torii et al. [2009], Micusik and Kosecka [2009]], oblique aerial images [Wu et al. [2018], Haala et al. [2015a]] and UAS images [Oniga et al. [2022]], etc.



(a) (b)

Figure 3.1: Point cloud generated from imagery (Figure from He et al. [2023])

Three-Dimension point cloud data have been a crucial data source for 3D city model reconstruction [Wang et al. [2019]]. Laser scanning-based methods mainly utilize LiDAR point clouds for 3D reconstruction. In some studies, other data sources are also used to assist. Currently, ALS [Overby et al. [2004], Chen et al. [2018]], TLS [Pu et al. [2006]; Arayici [2007]], and Mobile Laser Scanning (MLS) [Li et al. [2016]] are commonly used as the data source of 3D city model reconstruction.

The point clouds, including LiDAR point clouds and photogrammetry-based point clouds, are utilized to apply the surface reconstruction algorithms to create a surface mesh. There are multiple existing surface reconstruction algorithms, including Delaunay triangulation, and Poisson surface reconstruction. The currently advanced and lightweight software Poly-

fit is specifically designed for polygonal surface reconstruction [Nan and Wonka [2017a]]. As soon as the surface mesh is generated, a LOD2 or LOD3 building model can be created based on the mesh with various techniques.

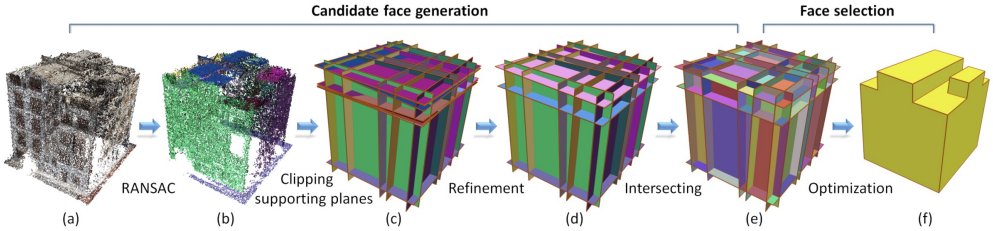


Figure 3.2: Surface model generation using Polyfit (Figure from Nan and Wonka [2017a])

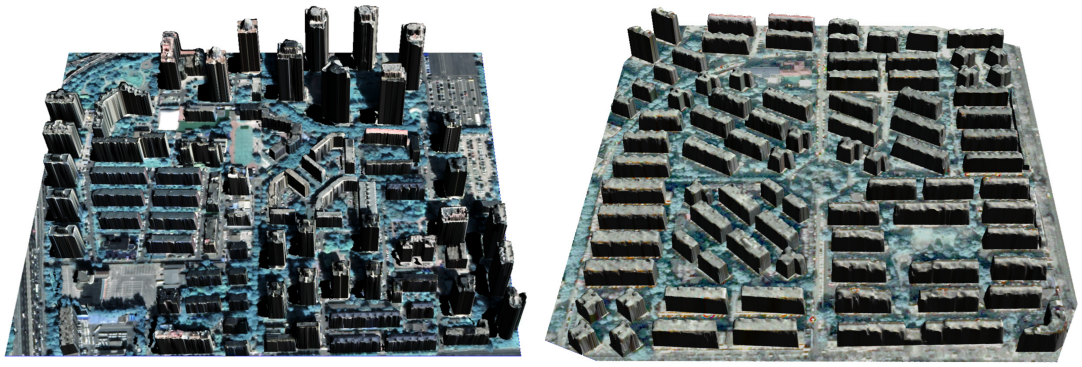
3.2 LOD2 building model reconstruction

As an important intermediate product of the LOD3 model generation process, in this section, the LOD2 model generation process will be emphasized.

For satellite images, He et al. suggested a high-precision 3D building model reconstruction method that utilized the registration between Digital Surface Model (DSM) from satellite stereos and Open Street Map (OSM) [He et al. [2023]] (Figure 3.3). This technique obtained building footprint from OSM data and height from DSM which was generated from satellite stereo images, then registered two of them using a feature-based approach and applied 3D reconstruction based on them. The proposed method has the benefit of being able to achieve high-precision reconstruction with easily accessible open data. The limitation of this method is that only one building can be generated at a time, which makes it difficult for large-scale 3D city reconstruction.

Bullinger et al. presented a novel method for 3D surface reconstruction utilizing multi-date satellite imagery [Bullinger et al. [2021]]. It leveraged temporal information from multiple satellite images to apply SfM to generate accurate and detailed 3D models. The suggested method enhanced the accuracy and effectiveness of the reconstructed 3D surfaces and reduced artifacts compared to existing methods. The disadvantages of this method are that the pre-processing process like atmospheric correction and co-registration might affect the final reconstruction result, and the dense matching process possesses a high computational cost, especially for larger data sets, making it unsuitable for the reconstruction of large 3D city models.

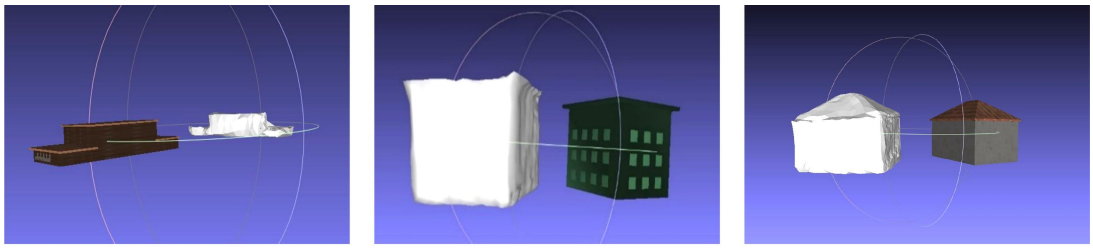
Based on street view imagery, Biljecki et al. presented a deep learning-based method for 3D building reconstruction from single SVI [Pang and Biljecki [2022]]. They implemented single-view reconstruction, two-view reconstruction, and mesh refinement to reconstruct 3D



(a) (b)

Figure 3.3: The 3D building models of Beijing and Shanghai (Figure from [He et al. [2023]])

building models, and Chamfer Distance (CD) was employed to assess the result. The results indicated that the single-view reconstruction was the least accurate in terms of geometric reconstruction, with large errors in reconstructed volume and surface area, while results of two-view reconstruction have better CD and better structure prediction (Fig 3.4).



(a) (b) (c)

Figure 3.4: 3D city model result of two-view reconstruction (Figure from [Pang and Biljecki [2022]])

This method can be effective even in cases with occlusions and different lighting conditions, and the required input data is so simple that only one image is enough. Rich information on façade elements can also be obtained with this method. However, the accuracy of the 3D reconstruction result may be limited by the quality and resolution of the input SVI, and it does not perform well in the reconstruction of complex and irregular buildings. There are also existing techniques for the accuracy assessment. Bruno and Roncella compared the 3D city models generated from Google street view images and LiDAR and aerial images to assess the accuracy [Bruno and Roncella [2019]]. The outcomes demonstrated that the SVI-generated 3D model approach is superior from a cost and large-scale model generation perspective. However, the quality of the reconstructed 3D model is influenced by the quantity and quality of SVI, and the accuracy is lower than that of LiDAR-based and aerial image-based 3D models, especially on complex building reconstruction.

Torii et al. presented a pipeline for the generation of 3D city models using Google Street View panorama images with SfM [Torii et al. [2009]]. This method combined computer vision techniques with geometric modeling and optimization to reconstruct 3D building façades and 3D urban environments. The resulting 3D models demonstrate the potential of using readily available street view images for efficient and large-scale urban modeling, offering a cost-effective alternative to traditional methods that rely on expensive and time-consuming data acquisition.

Since oblique aerial imagery is taken with the optical axis deviating from the vertical and is captured with an angle (e.g., 30 degrees, 45 degrees), the building façades are well visible in the images, thus, they are nice data sources for the reconstruction of detailed 3D city models. Wu et al. presented a method to create 3D urban models by combining oblique aerial and terrestrial images [Wu et al. [2018]]. Based on the complementary characteristics of the two images, image registration was performed and a dense point cloud was generated to create the final 3D urban model. Compared to using each type of imagery separately, combining oblique images and terrestrial images can obtain more accurate and detailed 3D urban models and handle the occluded areas which are in urban scenes more effectively.

Oniga et al. introduced a semi-automatic approach for 3D urban model generation on the basis of low-cost oblique UAS images [Oniga et al. [2022]]. The highlight of this pipeline is the method of 3D model reconstruction, by creating the planar faces by region-growing algorithm and the piece-wise intersection of planar faces.

To summarize, photogrammetry-based pipelines are mostly used to generate dense point clouds by aligning different types of images and converting them into 3D city models by different methods. The data are usually the single type of imagery or imagery and supporting data, or a combination of multiple types of imagery.

For ALS point clouds, Chen et.al proposed multi-Source recTification of gEometric Primitives (mSTEP) [Chen et al. [2018]] to generate LOD2 building models, by generating the LOD1 model first using building footprint, base level, top level, and optimizing and refining the details of the rooftops using ALS and the architectural knowledge database, to improve the accuracy and realism of the building models. The benefits of this method are that it is fully automated and can handle complex buildings and high-density urban areas. However, the limitations are the strong reliance on the architectural knowledge database and therefore has limited applicability, as well as the level of details of the roof depending on the quality of the ALS. Overby et al. presented an automatic 3D building model reconstruction technique using ALS point cloud and cadastral data [Overby et al. [2004]]. Hough transform was applied to detect roof planes from the point cloud, and combined footprint data to generate 3D building models. This pipeline is robust as Hough transform is effective to detect roofs in various quality of data efficiency, it is also suitable for large-scale 3D urban model reconstruction.

Pirotti et al. introduced a deep learning-based method to detect roofs and façades from ALS point clouds [PIROTTI et al. [2019]]. The ALS point cloud was firstly filtered and organized

into a voxel-based representation, then utilized a 3D Convolutional Neural Network (CNN) to classify each voxel as roof, façade, or background, and generated the resulting building models. The advantage of this method is that the voxel-based representation could make it easy to process large ALS datasets, which is efficient for large-scale 3D urban reconstruction. However, the accuracy of the reconstruction results is low, and the CNN model still needs to be optimized by tuning the hyperparameters of the hidden layer.

The MLS system combines a 3D laser scanner, Global Navigation Satellite System (GNSS), Inertial Measurement Unit (IMU), and cameras [Wang et al. [2019]]. The MLS point clouds are characterized by high density, high accuracy, and flexible data acquisition. MLS point cloud is frequently utilized in urban areas for a variety of purposes, including urban transportation facility and building modeling, autonomous driving, etc. Since it can collect full details about objects, MLS has become a suitable data source for higher LOD building modeling. Li et al. proposed a pipeline to extract and simplify building façade pieces utilizing morphological filtering with MLS point clouds [Li et al. [2016]]. The point cloud projection algorithm was presented to convert the point cloud to a raster image and obtained façade elements in the image space, then applied the inverse transformation to transform façade features from 2D to 3D space.

3.3 LOD3 model reconstruction

Since the details of LOD3 buildings are greatly enhanced in details compared with the lower LOD buildings models(see Table 2.1), the reconstruction techniques are more complex and usually require the integration of multiple sources of data to extract sufficient structural information, such as oblique images, terrestrial LiDAR point clouds, and airborne LiDAR point clouds. The reconstruction techniques are more advanced as well, including interactive manual editing, fitting and integrating building geometric primitives and 3D templates[Huang et al. [2020], Nan et al. [2015]], and extracting detailed façade information from images [Pantoja-Rosero et al. [2022], Zhang et al. [2019]]. In most approaches, the LOD2 model is generated first and then the LOD3 model is generated based on the LOD2 model [Pantoja-Rosero et al. [2022], Huang et al. [2020], Zhang et al. [2019]].

Pantoja-Rosero et al. offered a two-stage method combining SfM and semantic segmentation [Pantoja-Rosero et al. [2022]]: first, generate the LOD2 model by SfM, and then add openings information to the LOD2 model by semantic segmentation. Polyfit [Nan and Wonka [2017b]] was utilized to generate LOD2 building models, and TernausNet was trained to semantically detect opening corners, openings, and façades on 2D images. In order to convert the detected 2D opening to 3D, Scale-invariant Feature Transform (SIFT) was used to detect the same key information from two different images and then triangulate the points to the 3D ones. This method performed 3D reconstruction using SfM as well as multiple uses of deep learning (in opening segmentation and detection and key points filtering after SIFT), which is more demanding on computational resources.

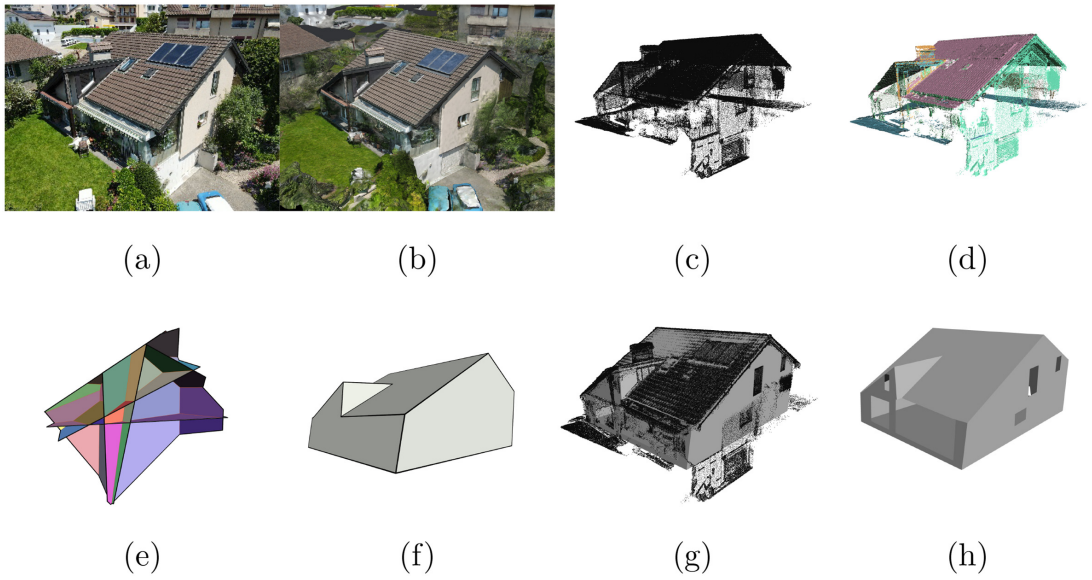


Figure 3.5: LOD3 model generation combining SfM and semantic segmentation (Figure from [Pantoja-Rosero et al. [2022]])

Huang et al. presented a parametrized top-down pipeline, to reconstruct the LOD3 building model (Figure 3.6) using a shell model [Huang et al. [2020]]. The building components can be obtained from different data sources, for instance, roof structures are extracted from aerial images, and façade elements are extracted from terrestrial images. After obtaining their optimal fitting primitives from the predefined primitive library, they are semantically and geometrically integrated into a shell model, which is a generative statistical model for LOD3 building reconstruction. It is a hybrid of Constructive Solid Geometry (CSG) and Boundary Representation (BRep), thus the building components can be integrated into a model with CSG operations. FC-DenseNet56 was used to detect openings in rectified façade texture images. The detection results were projected back into the 3D space to be integrated into the original building model. The combination of aerial imagery and terrestrial imagery can capture detailed information about the building more comprehensively, resulting in a higher level of detail in the model of roofs and windows, and the results meet the requirements of the LOD3 model. The weakness of this research is that building components detection relies heavily on the predefined primitive library, and thus it is difficult to accommodate diverse building structures in large 3D city model reconstructions.

Nan et al. proposed a method to assemble 3D templates to generate LOD3 building models on coarse-textured building models [Nan et al. [2015]] (seeing Figure 3.7). A coarse model was first generated using SfM and MVS and optimized by image rectification, then the detailed structures of the building were reconstructed by finding and assembling the most appropriate 3D templates for each textured façade. This framework successfully generated more detailed building models than existing techniques, the template assembly algorithm is able to effectively illustrate the façade structures. However, the framework is more suitable

Figure 3.6: LOD3 model reconstruction (Figure from [Huang et al. [2020]])

for buildings with similar detailed structures, with buildings that the repetition does not exist, the template should be identified by relying on manual labor.

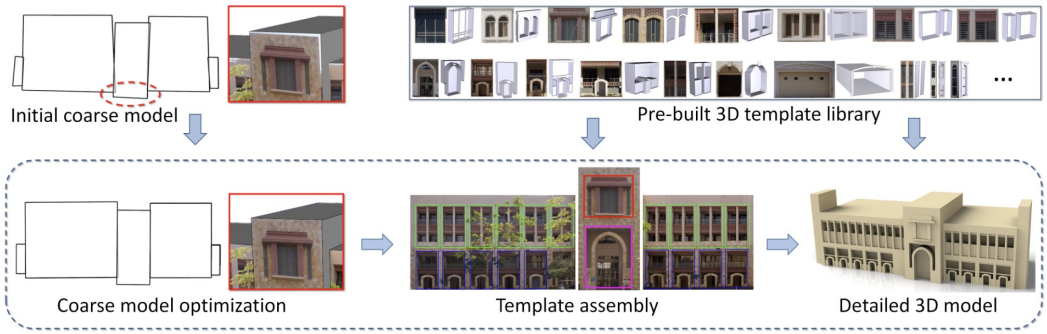


Figure 3.7: 3D template assembly framework (Figure from [Nan et al. [2015]])

Wen et al. introduced a method for LOD3 building model reconstruction using oblique images and multi-source point clouds [Wen et al. [2019]], while multi-source point clouds were utilized to guarantee surface accuracy and oblique images were used to obtain good edge performance. The building roofs were extracted from the Unmanned Aerial Vehicle (UAV) LiDAR point cloud by normal vector clustering and performed as the constraints to extract the corresponding integrated UAV and terrestrial point clouds, and building plane primitives were extracted using the Random Sample Consensus algorithm (RANSAC). Feature lines were extracted from oblique images using Edge Drawing Line technique (EDLine), which was employed to enhance the reconstruction precision of the edges of the building. For the topological errors that were difficult to fix automatically were fixed by interactive manual editing. In this study, the outline constraints generated by the roof point cloud are similar to the role of the footprint data in other studies. The reconstruction accuracy of this multi-source data method was significantly improved compared to the reconstruction results from a single data source, which benefited from making full use of the feature lines extracted from imagery and point clouds for the reconstruction of the basic building frames. However, this work is not fully automated and requires high-accuracy registration of multiple data sources.

Zhang et al. proposed a pipeline to enhance the LOD2 CityGML model by extracting open-

ings from high-resolution façade images to obtain the LOD3 model [Zhang et al. [2019]]. For the existing LOD2 model, corresponding corrected façade images were obtained, and 2D façade elements were detected and extracted by Mask R-CNN, then integrated into the LOD2 model. The façade information was enhanced in the resulting LOD3 models and the visual quality was improved, while the original building structure was preserved as well. Moreover, this method does not involve a large amount of computation, so it can be extended to large urban models.

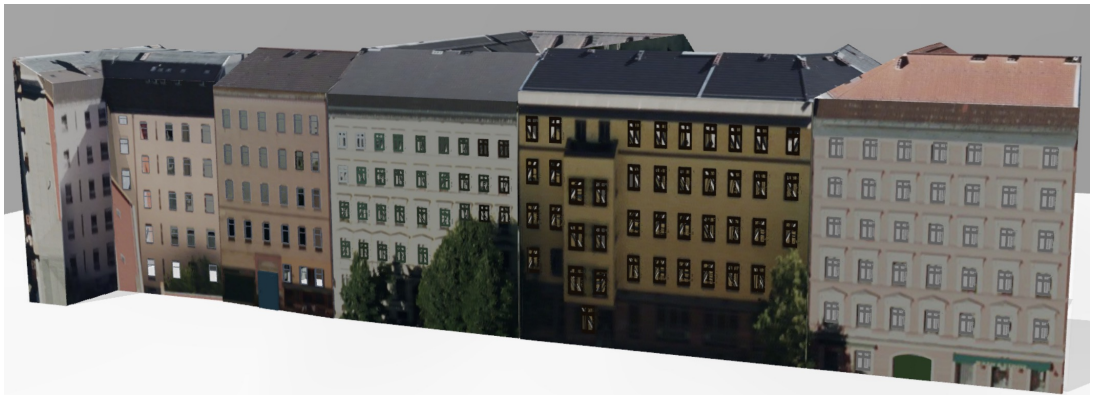


Figure 3.8: Enhanced LOD3 CityGML models (Figure from [Zhang et al. [2019]])

3.4 Façade element detection and addition

To achieve LOD3 3D building models with enhanced detail, as required by LOD3 standards, it's common to incorporate comprehensive semantic information, such as the inclusion of doors and windows. Consequently, the goal of façade element detection is to identify these openings from façade texture images. This detection task employs a variety of techniques, ranging from pixel-based approaches [Yang et al. [2015b], Liu et al. [2020a]], to those that rely on advanced deep learning methodologies [Liu et al. [2020b], Hensel et al. [2019], Redmon and Farhadi [2018]].

Textures and façade elements can be added through fully automatic, semi-automatic, or manual methods. Fully automatic approaches typically offer a faster processing time, as they reduce the need for human intervention, and are easily scalable for larger datasets. However, these methods may not consistently deliver accurate outcomes, particularly when dealing with intricate or irregular building models and façades. Semi-automatic techniques strike a balance between efficiency and quality, enabling users to contribute their input to the process for enhanced results. This makes them more adept at handling complex and irregular building façades. However, their processing speed is comparatively slower than fully automatic methods, rendering them less suitable for managing extensive urban model

datasets.

Liu et al. proposed a method combining deep CNN and a symmetric regularization term to derive a new loss function for training for end-to-end training [Liu et al. [2020a]], where prior knowledge is involved (the assumption that most windows and doors have a highly symmetric rectangle shape). The symmetric regularization and prior knowledge help improve the performance of predicting the location and shape of windows, and the results are more accurate and visually pleasing.

(a) image (b) ground truth (c) proposed method (d) image (e) ground truth (f) proposed method

Figure 3.9: Windows detection results using DeepFacade(Figure from [Liu et al. [2020a]])

A bottom-up multi-level features extraction approach for building façade recognition using oblique aerial images is presented by Yang et al.[Yang et al. [2015b]]. This approach employed edge detection, region growing, and Hough transforms to identify building façades. To segment façade components, the method utilized morphological operations, connected component labeling, and a modified K-means clustering algorithm. Finally, machine learning algorithms were applied to classify the extracted features. This approach achieves accurate building façade recognition and segmentation in oblique aerial images (Figure 3.10 (a)), but it is only applicable to the case of regular distribution of elements and will fail in the case of the incomplete façades and irregular shape of façades (Figure 3.10 (b) and (c)).

3.4.1 Neural Network for openings detection

Object detection and segmentation have consistently been subjects of considerable interest and research within the field of computer vision [Liu et al. [2020b]]. These tasks have seen significant advancements because of the rapid growth of deep learning techniques, which enable the automatic extraction of features from large-scale data and their subsequent application to detection and segmentation problems. As a consequence, deep learning has become a dominant tool in the development and implementation of modern object detection and segmentation models, greatly broadening the scope of computer vision research and its practical applications.

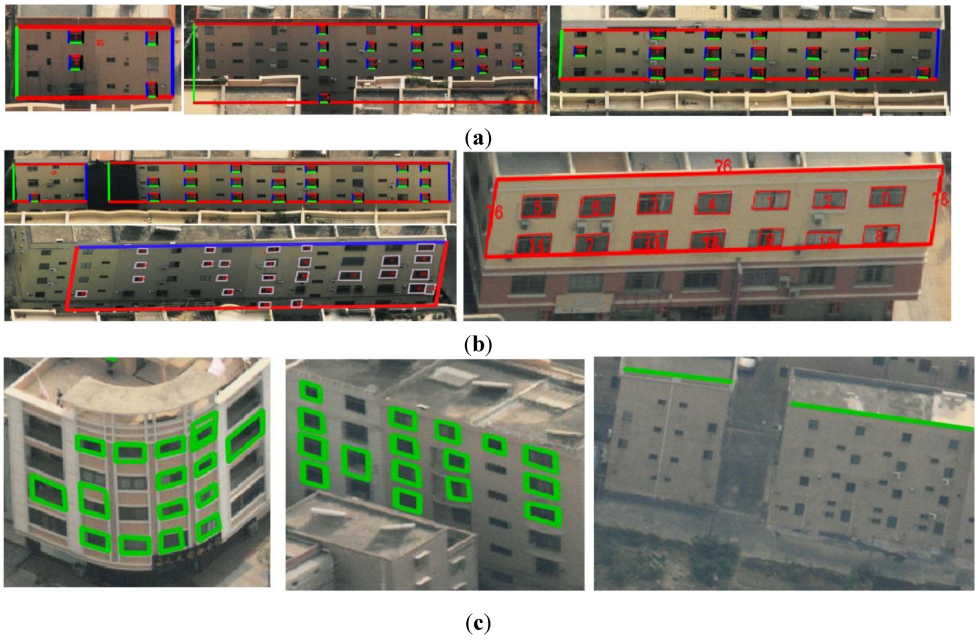


Figure 3.10: Three types of extracted façade elements results (Figure from [Yang et al. [2015b]])

3.4.2 YOLOv3

Hu et al. used YOLOv3 to detect the openings in their study. YOLOv3 is a real-time object detection model designed to process images in a single forward pass through the network [Redmon and Farhadi [2018]]. Unlike Faster R-CNN and Mask R-CNN, YOLOv3 processes images in a single pass, leveraging a unified architecture, the Darknet-53 backbone, multi-scale predictions, anchor boxes, a tailored loss function, and improved bounding box regression and class prediction techniques. The network separates the image into multiple regions, predicts bounding boxes and probabilities for each region, and then weights the bounding boxes according to the probabilities. Darknet-53 serves as the backbone for YOLOv3 and feature pyramid network (FPN) predicts the bounding box. Darknet-53 is a convolutional neural network that is 53 layers deep, which is designed for fast computation while maintaining high accuracy. It is tested that Darknet-53 is 1.5 times faster than ResNet-101. FPN can predict bounding boxes at three scales, allowing the model to detect objects with varying sizes more effectively and accurately. YOLOv3 employs predefined anchor boxes to enhance the accuracy of object location and size predictions by calculating offsets and scales for these boxes. For bounding box regression, it uses logistic regression, which results in improved performance in predicting bounding box coordinates. Furthermore, YOLOv3 utilizes a multi-label classification approach with logistic activation functions for class prediction, enhancing the ability for detecting objects that belong to multiple classes.

3.4.3 Faster R-CNN

Hensel et al. used the deep neuron network Faster R-CNN to detect the windows and doors from façade images[Hensel et al. [2019]], as Faster R-CNN can implement object detection and segmentation efficiently and accurately [Ren et al. [2015]]. Faster R-CNN is an object instance detection and segmentation deep neuron network that is developed based on Fast Region-based Convolutional Network method (Fast R-CNN). In Fast R-CNN, region proposals are generated from the selective search method, while in Faster R-CNN Region Proposal Network (RPN) is introduced to generate region proposals from feature maps, which enables the end-to-end trainable object detection models that can implement object detection in near real-time. Furthermore, the detection network shares the convolutional layers with RPN, which greatly improves the efficiency and reduces the computation for object detection. The basic structure of Faster R-CNN is shown in Figure 3.11. In the first stage, a set of convolutional layers and pooling layers are employed to capture image information and generate a feature map. The feature map is fed into the RPN then, and generates region proposals, which represent the likelihood scores of containing an object. Region of Interest (RoI) pooling resizes each region proposal to a fixed size, which facilitates subsequent use of the Fully Connected (FC) layers. The fixed-size region proposals are then processed by a series of FC layers to perform two tasks: classify the objects and assign them to specific classes, and refine the bounding box to improve the localization of the detected objects.

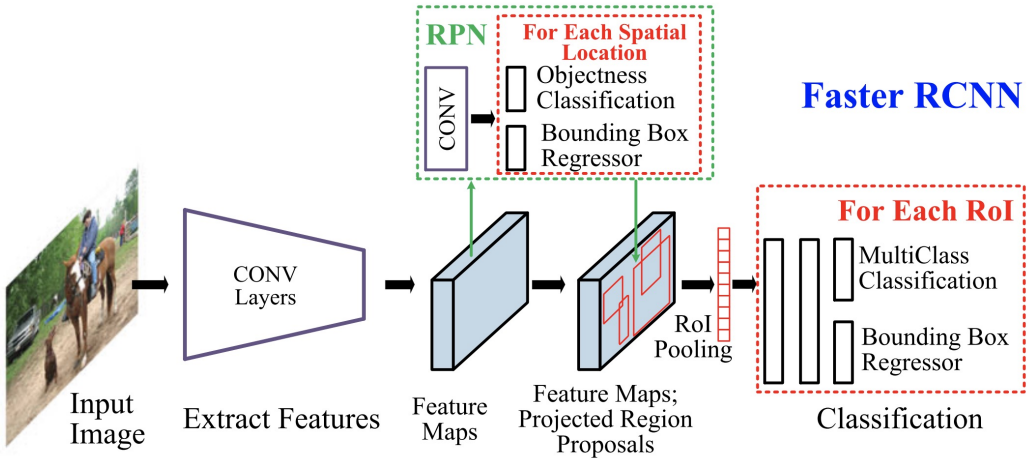


Figure 3.11: General structure of Faster R-CNN (Figure from [Liu et al. [2020b]])

3.4.4 Mask R-CNN

Zhang et al. [Zhang et al. [2019]] and Liu et al. [Liu et al. [2020a]] chose Mask R-CNN to detect windows and doors [Zhang et al. [2019]]. Mask R-CNN is a flexible and general

framework for object instance segmentation, which is built upon Faster R-CNN [He et al. [2017]]. The workflow is basically similar to Faster R-CNN, but there are two innovations. The first innovation of Mask R-CNN is a parallel branch that is added to the model to predict binary masks, consisting of convolutional layers and a sigmoid activation function, generating pixel-level masks for each object. Faster R-CNN produces bounding box regression. The second is RoI pooling is replaced by RoIAlign. Since Faster R-CNN is not pixel-to-pixel alignment between network inputs and outputs, RoI will bring the misalignment problem when integerization and resizing to region proposals to fixed-size, and the quantization-free layer called RoIAlign in Mask R-CNN could fix the errors. For each region proposal, RoIAlign generates a sampling grid with a fixed size in the bounding box, and uses bilinear interpolation to calculate the precise interpolated values of the sampling points using the distance-based weighted average with four nearest feature maps. The last step of RoIAlign is performing max or average pooling to generate the resulting pooled output. In this way, exact spatial locations are preserved and pixel-based masks are generated.

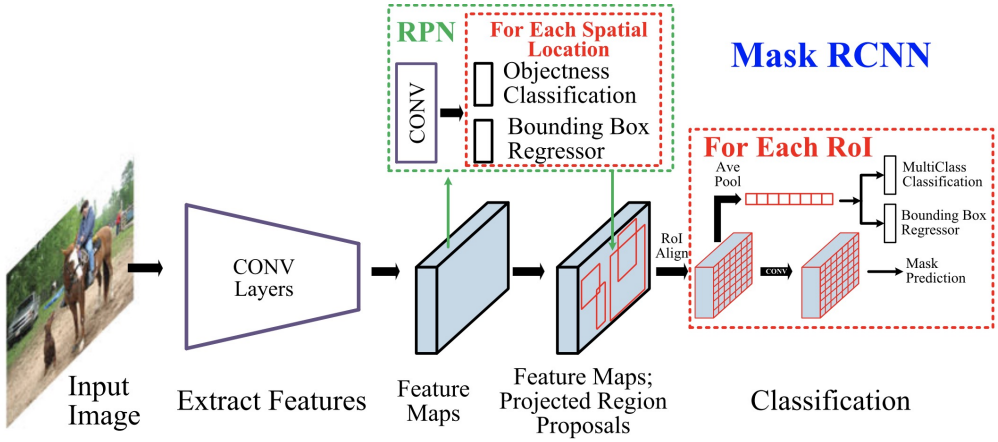


Figure 3.12: General structure of Mask R-CNN (Figure from [Liu et al. [2020b]])

3.4.5 Evaluation of deep learning model

Evaluating the accuracy of deep learning models is essential for several reasons. First, assessing the performance and confidence level of a single model in a specific task provides information that can lead to further improvements. For instance, model optimization can be achieved by adjusting critical hyperparameters, including learning rate and batch size. Moreover, the confidence level in the model predictions is particularly important in deep learning-based applications, where the reliability of outcomes is crucial. Accuracy assessment offers insights into the degree of confidence in the model performance. Evaluating accuracy helps determine the model's generalization capabilities when dealing with new and unfamiliar datasets, a critical attribute for deep learning models. As a quantitative

measure of model performance, accuracy evaluation not only allows for gauging the effectiveness of a single model but also enables a comparison between different models on the same dataset. This comparison can inform the selection of the most suitable model for a given research context, ensuring optimal outcomes in the target application.

Some general model evaluation parameters are commonly used in many scenarios:

- Intersection over Union (IoU): it is a parameter utilized in object detection and image segmentation tasks to evaluate the similarity between the predicted mask and the ground truth mask. It is calculated as the ratio between the area of intersection and the area of the union of the two mask:

$$\text{IoU} = \frac{(mask_{predicted} \cap mask_{groundtruth})}{(mask_{predicted} \cup mask_{groundtruth})} \tag{3.1}$$

  IoU values range from 0 to 1, where a value of 0 indicates no overlap and a value of 1 indicates a perfect match between the predicted and ground truth mask. In practice, a higher IoU score signifies the better performance of the model.

- True Positive (TP) and False Positive (FP): TP and FP compare the prediction against the ground truth, which helps in understanding the performance of the model. TP means the model correctly classifies a positive object as positive, and FP means the model mistakenly classifies a negative object as positive. However, TP and FP are more used to evaluate binary classification tasks.

- Precision: it measures how well the model correctly classifies positive instances out of all instances it classifies as positive, thus is also called positive predictive value. TP and FP are utilized to calculate it:

$$Precision = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.2}$$

  The higher precision indicates that the model has a lower false positive rate and better performance the model has.

- Recall: it is used to evaluate the performance of classification results, by calculating the proportion of actual positive instances that are correctly identified by the model. It is calculated by TP and False Negative (FN), where FN are the positive objects that are mistakenly classified as negative.

$$Recall = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{3.3}$$

  A higher recall value indicates that the model performs more effectively at identifying positive instances and minimizing FN.

- F1-score: it is the harmonic mean of recall and precision. It symmetrically represents both precision and recall in one metric [Buitinck et al. [2013]].

$$F1\_score = 2 \times \frac{precision \times recall}{precision + recall} \tag{3.4}$$

The F1-score ranges from 0 to 1. A higher F1-score represents a better balance between precision and recall, meaning the model is performing well in terms of both FP and FN.

Using recall only to evaluate the model can not provide a complete picture of the performance of the model, since FP are not considered. Precision and F1-score together can provide a better evaluation of the performance.

The extraction results are illustrated in Figure 6.5 and 6.6, In general, window extractions demonstrate a high degree of accuracy, with confidence levels exceeding 90%. In contrast, door detection presents a more significant challenge, often resulting in lower confidence levels and instances where doors are mistakenly detected as windows. Consequently, during the detection process, confidence thresholds for windows and doors are adjusted to 95% and 70%, respectively, in order to obtain more accurate door results. When the recognition results for windows and doors overlap, doors are given higher priority. Although the detection of balconies is not the primary focus of this study, it is worth noting that balconies are occasionally misclassified as windows due to similarities in appearance and structure. During the extraction process, the use of oblique aerial images was found to have certain limitations. As these images are captured from an overhead perspective, balconies protruding from the building façade can obstruct the view of windows and doors situated below. This obstruction can have a considerable impact on the extraction results, potentially reducing the accuracy and completeness of the identified features.

3.5 Façade layout regularization

The architectural design often takes regularity into aesthetic considerations, which frequently results in the windows of buildings being in parallel alignment. This has been taken into account in this research for the aesthetics of the LOD3 model and therefore introduced layout optimization in the pipeline. Layout optimization refers to the optimization and regularization of the layout of façade elements, generally through two main aspects of optimization: size and location. Presently, various approaches exist for façade layout regularization, encompassing interactive refactoring, manual optimization, and more automated techniques, which include deep learning-based methods as well as regularization strategies employing Binary Integer Programming (BIP) and mixed integer linear programming (MILP) [Hensel et al. [2019], Hu et al. [2020], Liu et al. [2020a]]. In the subsequent self-study, we will provide an in-depth examination of each of these approaches.

Hensel et al. proposed a pipeline to generate detailed façades and LOD3 models combining deep learning and MILP [Hensel et al. [2019]]. This pipeline aligned the openings by MILP. MILP may easily add new constraints without changing detection results, since the MILP optimization algorithm decides about coordinate changes in one step, unlike the other two-step optimization.

Hu et al. presented a fast and regularized pipeline for reconstructing and regularizing

façade primitives from SVI using BIP [Hu et al. [2020]]. In this pipeline, YOLOv3 was employed to detect façade primitives. In BIP optimization, the size of the bounding boxes for regularized arrangements is automatically clustered while taking geometric fitness, regularity, and extra constraints into account. Qualitative evaluations, quantitative evaluations (the number of used model spaces), and runtime comparisons were applied to evaluate the result. It can be seen from Figure 3.13 that the semantic objects are more consistently and neatly organized, and after regularization, the number of the chosen model space is greatly decreased. The runtime is also much faster than the previous MILP approach.

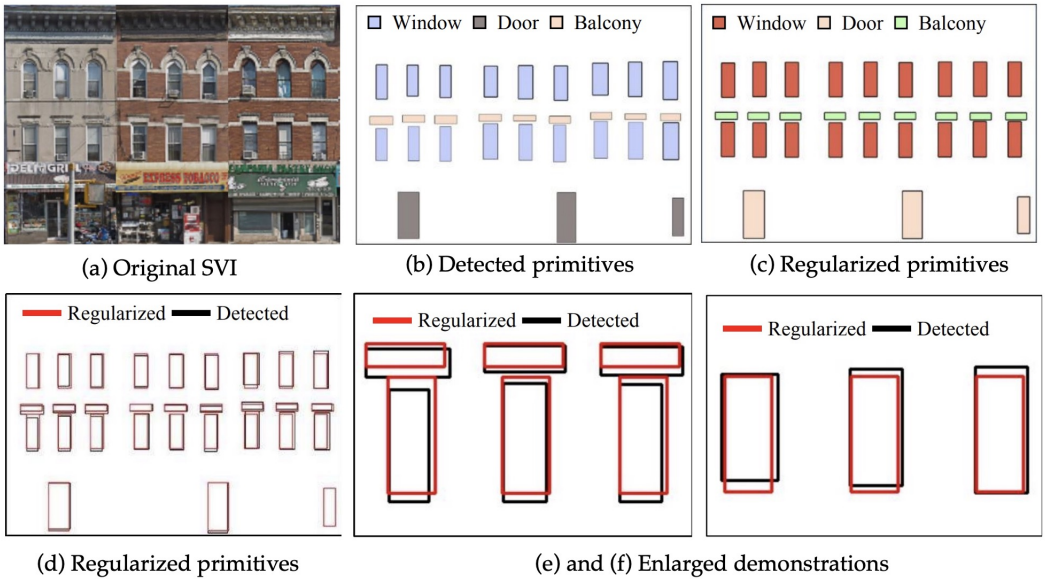


Figure 3.13: Comparison of ground truth image and regularized results (Figure from [Hu et al. [2020]])

Based on the assumption that most openings have a symmetric rectangle shape, Liu et al. presented a symmetric loss function used in deep neural network FCN-8s [Liu et al. [2020a]], and penalize any non-symmetric detection results, where the rectangle loss and the detector loss are the two components of the loss function. Since the symmetric assumption was introduced into the deep neuron network for regularization, prior knowledge about structures of façade elements can well improve the accuracy of the shape and location of the detected openings.

The method of layout regularization is also proposed in the LOD3 model reconstruction method of Pantoja-Rosero et al [Pantoja-Rosero et al. [2022]]. In terms of position adjustment, RANSAC with linear regression was utilized to obtain multiple vertical and horizontal lines in the local system, then aligned edges and centroid of the openings with the regressed lines. In terms of size adjustment, adjusted edges of the openings with similar areas to achieve the same area. Such regularization would make openings results neater, but it is

difficult to fit on irregular façades.

Jiang et al. proposed a constraint detection algorithm to implement the layout regularization of the façade elements [Jiang et al. [2016]]. Based on the three proposed constraints: alignment constraints, same-size constraints, and same-spacing constraints, Based on the proposed three constraints, find out the most optimal constraint formula, generate a candidate group, and apply it to regularize the layout using the energy function. find the matching constraint formula and apply it to regularize the layout. The limitation of the method is that semantic information is not taken into account and another is that the performances on complex-shaped building façades are not yet clear.

4 Methodology

4.1 Overview

This chapter presents an overview of the proposed pipeline, encompassing all stages from data pre-processing to the integration of the final results into the LOD2.2 building models. The flowchart is shown in Figure 4.1.

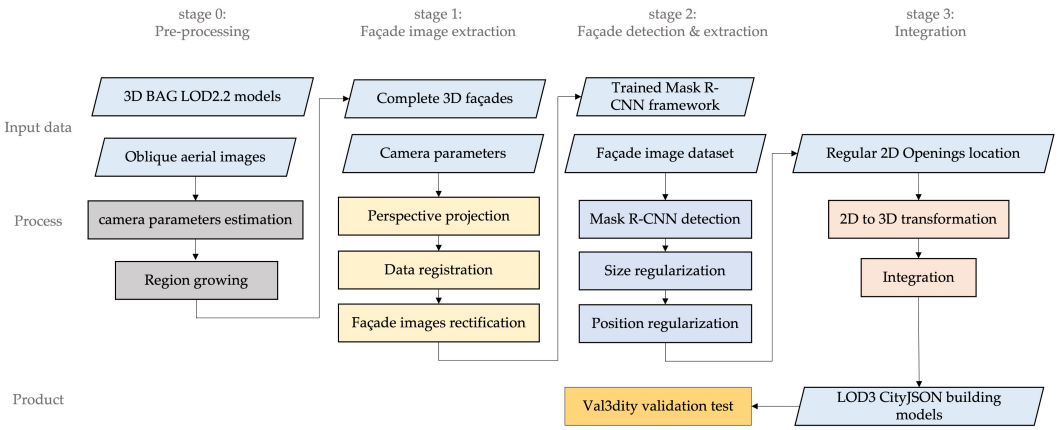


Figure 4.1: Pipeline workflow

Initially, we present the data pre-processing stage in Chapter 4.2, where we highlight the benefits of utilizing OBJ data and elaborate on the co-planar surface merging technique. We also describe the method for obtaining camera parameters from oblique aerial images, which aids in estimating the approximate coverage of each image and determining the complete containment of buildings within the images.

Subsequently, the first stage of the pipeline involves projecting the corner points of 3D façades onto the 2D image space using perspective projection in Chapter 4.3. The resulting rectangles, formed by the projected 2D points, serve as constraints for extracting façade images. We employ least squares regression (LSR) to adjust projection offsets and perform rectification to obtain optimal façade images.

In the second stage, the Mask R-CNN framework is utilized for the detection and segmentation of openings in the obtained façade images, which is presented in Chapter 4.4. We employ an existing dataset of façade images, along with a smaller subset generated in the

first stage, for model training and validation. The trained model is then applied to all façade images to identify the pixel locations and sizes of each opening. We assess the accuracy of the Mask R-CNN result using Intersection over Union (IoU) and overall accuracy metrics. In order to obtain a better layout, we normalize the detected openings in terms of size and position, which is explained in Chapter 4.5.

In the final stage, the 2D to 3D conversion and final integration are stated in Chapter **??** and Chapter 4.7, the 2D coordinates are converted into 3D coordinates (via scaling) and incorporated into the original LOD2.2 building models. This pipeline demonstrates a novel approach to extracting 2D façade images utilizing current 3D building models, detecting openings, and augmenting 3D building models with detailed information on openings.

4.2 Data pre-processing

4.2.1 Camera parameters adjustment

The acquisition of camera parameters is carried out using Pix4D software. Pix4D was selected for this research due to its superior performance in camera calibration, even without Ground Control Points (GCP) in the test area. By importing the oblique aerial images from four orientations of the experimental area, along with the corresponding camera coordinates for each image, the coordinate system is set to RD28992. Camera calibration of Pix4D is divided into four steps:

- **Feature extraction**: keypoints are automatically detected in each image;
- **Image matching**: find matches for each keypoints across all the images based on the similarity;
- **Camera parameters estimation**: use matches and the principle of epipolar geometry to estimate the initial intrinsic and extrinsic camera parameters;
- **Bundle adjustment**: it is achieved by minimizing the re-projection error between the observed points in the image and the expected location based on estimated camera parameters and 3D points. In this step, camera parameters are simultaneously adjusted and improved.
- **Georeferencing**: GCPs are utilized to ensure the reconstruction accuracy if available.

However, it's important to acknowledge that the procurement of GCP is typically not incorporated during oblique aerial image acquisition. Consequently, oblique aerial image datasets are often devoid of GCPs. This omission could compromise the accuracy of the camera parameters estimation, potentially instigating disparities between the estimated outcomes and the verifiable ground truth results, as illuminated in the subsequent stage.

4.2.2 Region growing method to fix co-planar surfaces

As illustrated in Figure 4.2, the co-planar surfaces of type WallSurface in the 3D BAG are somewhat split, both in .json and .obj formats, due to each vertex in the roof partition that is on the wall being extruded to an edge in the process of generating the wall by extrusion, multiple vertices cause the splitting of the wall. These irregular shapes are not conducive to the manipulation of 3D façades, nor are they suitable for extracting 2D façade images via 3D façade projection in this research. Retaining the original triangle surface or split surface would result in automatically extracted 2D images containing numerous incomplete windows. Consequently, it is essential to apply shape detection to address this issue.

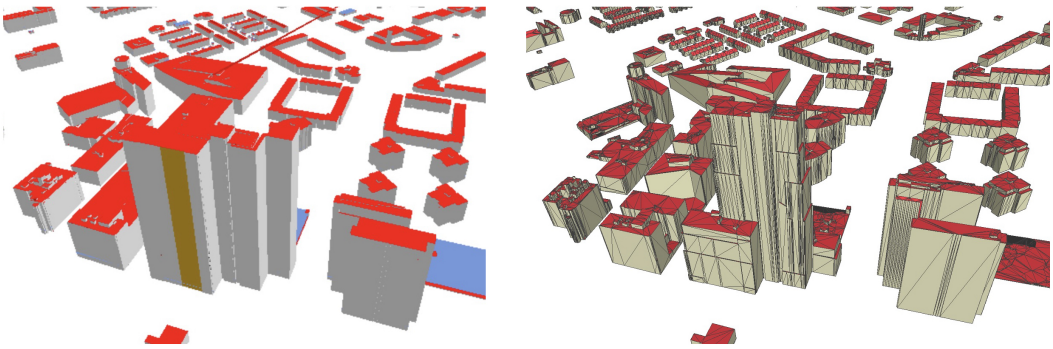
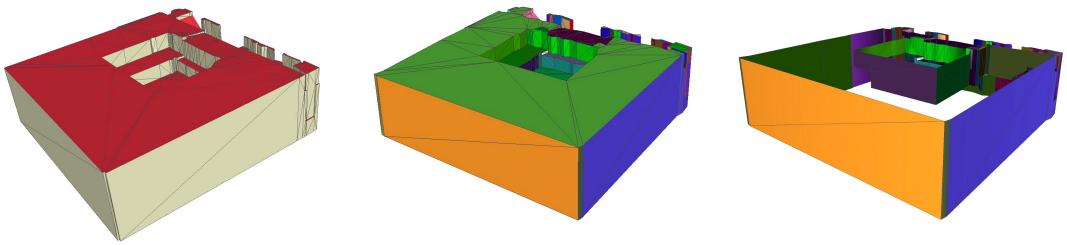


Figure 4.2: Co-planar surfaces are split in .json (left) and .obj (right) format

In this research, the co-planar surfaces problem is resolved using the region-growing algorithm. An empty mesh surface set is first initialized to represent the segmented region. Seed faces are then randomly selected from the mesh and added to the surface set. For each neighboring seed surface, the similarity measure between the neighbor and the seed is calculated. The similarity here is generally defined by a set of criteria that determines whether adjacent faces should be included in the same region. Once the similarity measure doesn't reach the threshold, add the neighbor to the segmented region set. Recursively repeat the measurements for the newly added face, until there are no more faces left to measure.

Following the shape detection process, the segmented meshes are color-coded based on the results, and faces determined to be in the same plane are assigned the same color (Figure 4.3 (a) and (b)). In comparison to the original meshes, the structure of the region-growing meshes is much more distinct, with the majority of the façades exhibiting a relatively complete shape. This substantially aids in extracting 2D façade images and significantly enhances the completeness of the extracted results. Furthermore, it preserves more accurate information regarding the location and the number of windows.

In this research, the primary focus is on the extraction of building façades, which necessitates the filtering and removal of numerous surfaces, such as roofs and footprints, that are not relevant to the task at hand. The wall surfaces in the 3D BAG dataset are generated through



(a) Initial building mesh (b) Mesh after merging of co-planar faces (b) Merged wall surfaces only

Figure 4.3: Three stages of 3D BAG building models

extrusion [Peters et al. [2022a]], which implies that these surfaces are perpendicular to the ground in theory. To achieve this objective, the normal vectors of the surfaces are calculated, and only the surfaces with horizontal normal vectors are retained for subsequent façade extraction steps. This approach ensures that irrelevant surfaces are effectively excluded from the analysis, thereby streamlining and fastening the extraction process (Fig 4.3 (c)).

4.3 Façade Extraction

The process of detecting and extracting façade images from 2D oblique images and subsequently associating them with their respective 3D façades is a complex task. This complexity arises primarily due to the lack of location information within oblique images, which makes it challenging to establish correspondence between the 2D and 3D space. As a result, the façade extraction methods presented in the related works are not applicable to this specific research. To address this issue, we proposed a novel approach to connect 3D and 2D space by perspective projection, using the camera parameters of oblique images. Perspective projection aims to create a 2D presentation of a 3D space, and the concept of it is to project 3D points onto a 2D image plane (equation 4.1). By leveraging these parameters, 3D façades can be projected into 2D space, using the resulting projected rectangles as constraints for the façade extraction process. This approach allows for more accurate and efficient extraction of façade images that correspond to their 3D façades. To ensure the accuracy and reliability of the extracted façade images, we also took into consideration potential systematic errors that may arise during the projection and extraction process. By identifying and mitigating these errors using the least square regression, we are able to obtain façade images in 2D space that closely correspond to their 3D façades, thereby significantly improving the overall effectiveness of the proposed approach.

The entire process can be summarized into three steps: first, the façades in 3D space are projected into 2D space using perspective projection; second, the projection results are used to perform least squares regression with some ground truth values that are manually created; the regression results are then employed to estimate and correct the offset for the rest of the projection results. And finally, the correction results are rectified to obtain the ideal

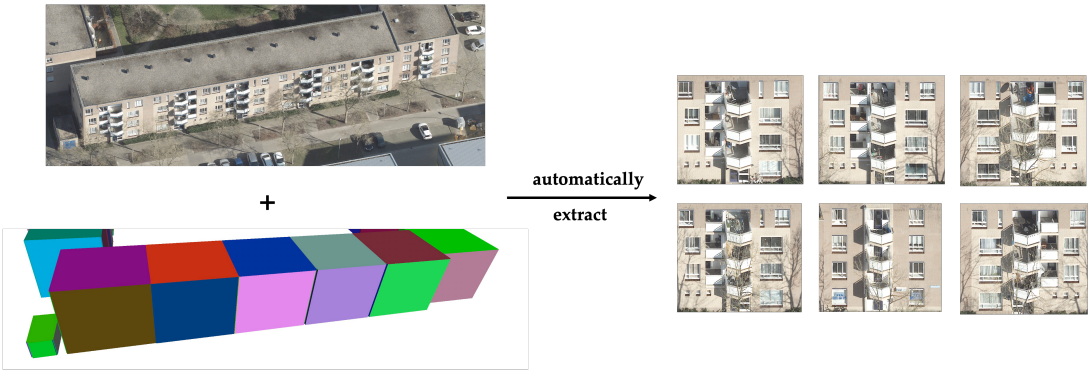


Figure 4.4: General workflow of the 1st stage

final façade images. The input data and the ideal result of this stage are illustrated in Figure 4.4

The perspective projection is performed on the corners of the façade. The projected 2D points can form a rectangular constraint. At the moment, we have a preliminary 3D-2D correspondence about the same façade. However, this correspondence is currently imprecise. This correspondence will be brought to a more precise level in subsequent phases.

4.3.1 3D Façade projection

The objective of 3D façade projection is to project all the 3D façades from the 3D BAG building model onto their corresponding imagery. The projection of a point from a 3D space to a 2D image space is implemented by combining the intrinsic and extrinsic parameters of the camera. The intrinsic parameters are focal length and principle point that characterize the internal properties of the camera. The extrinsic parameters involve a rotation matrix and translation matrix, representing the position and pose of the camera in 3D space while capturing images. The perspective projection is illustrated in equation 4.1:

$$\begin{pmatrix} x_u \\ x_u \end{pmatrix} = \begin{pmatrix} \frac{fX'}{Z'} \\ \frac{fY'}{Z'} \end{pmatrix} + \begin{pmatrix} c_x \\ c_y \end{pmatrix} \tag{4.1}$$

where $(x_u), y_u)$ is the pixel coordinate of the 3D point projection, (c_x, c_y) is the principle point and (X', Y', Z') is the 3D point in camera coordinate system.

In the situation of buildings fully captured by images, this research does not address the issue of determining whether individual façades are visible to the camera from a specific location. Instead, the research employs the projection of all of the 3D façades onto the 2D image plane. The reason behind this approach is that in the subsequent stage of window recognition using Mask R-CNN, the projection results for façades that are not visible to the camera will be filtered out as there are no openings that can be recognized. This method

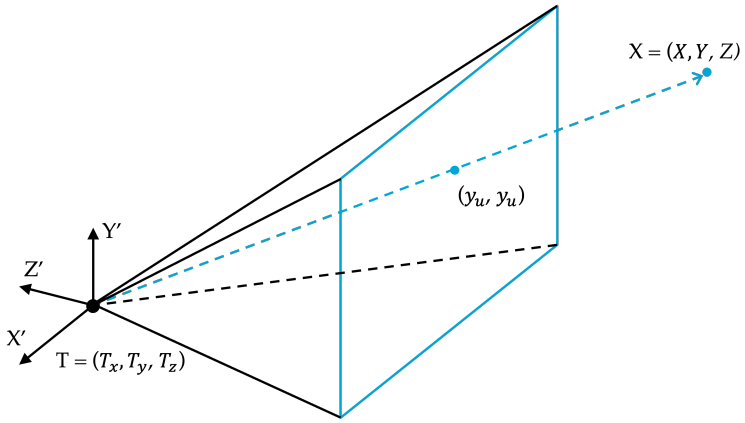


Figure 4.5: Perspective projection (from 3D to 2D)

effectively streamlines the process and focuses on the façades that are indeed visible and relevant for further analysis, eliminating the need to ascertain visibility at the outset.

The projection process is performed on the corner of each façade, transforming these 3D points into their respective 2D points that are illustrated by pixel coordinates in the image space. As a result, the projected 2D points establish a rectangular constraint within the 2D image space, as depicted in Figure 4.6 (a). This step yields a rudimentary correspondence between the 3D and 2D representations of the same façade. It is essential to note, however, that this correspondence is currently of limited precision. In the subsequent stages of the processing, refinements will be made to establish a more accurate and reliable correspondence between the 3D and 2D façade representations.

4.3.2 Projection result optimization by data registration

In Figure 4.6 (a) and (b), it can be observed that there are small offsets between the projection results and the ground truth results. This error is caused by the absence of GCP in the camera parameters estimation stage. It is crucial to optimize the result as this pipeline relies heavily on the completeness and correctness of the extracted façade images. Moreover, the offsets vary for different points at distinct locations, as well as the same point having different offsets in the X and Y directions in the image coordinate system. Owing to the challenges associated with the repeated utilization of deep learning techniques such as Mask R-CNN or Plane R-CNN [Liu et al. [2019]] for detecting corresponding façade images, and the concomitant increase in pipeline complexity, it is more advantageous and convenient to identify the characteristics intrinsic to offsets and apply data registration between projection results and images. This approach reduces the difficulties posed by employing deep learning techniques and streamlines the process. Image registration is the technique of spatially aligning two image datasets with each other, and ensuring the corresponding points in the images highly match. Upon examining the relationship between the projection

results and the ground truth points, it is discovered the corresponding lines are parallel and of equal length. There is only an offset in position, which exhibits a linear relationship for images captured in the same direction. Therefore, in selecting the registration model, there is no need to consider rotation or scaling, only translation is needed. This method streamlines the registration process and enhances the overall efficiency of the pipeline. Consequently, we opted to employ LSR to determine the best-fit lines for the X and Y offsets, respectively.

LSR is a statistical technique that minimizes the sum of squared residuals between the observed values (ground truth pixel coordinates) and the predicted values (estimated offsets) in this research, which can be described by the equation 4.2:

$$y = mx + c \tag{4.2}$$

where m and c can be calculated using the equation:

$$m = \frac{\sum(x_i - \bar{x}) \times (y_i - \bar{y})}{\sum(x_i - \bar{x})^2} \tag{4.3}$$

$$c = \bar{y} - m \times \bar{x} \tag{4.4}$$

This method enables us to predict the offset values of 2D points for which the ground truth values are unknown. We performed LSR on the projected results and ground truth results on X and Y separately for the ten façades and used the regression results to predict all the projected results. Upon acquiring a nice regression function, it can be applied to the other buildings contained in the images to enhance the accuracy of the projected result. This process involves using the derived regression model to predict outcomes for each projected result that we obtained from the perspective projection. The performance of the model on the other buildings will also be further assessed and validated to make sure the accuracy of the final result. As presented in Figure 4.7, the LSR model demonstrates an exceptionally strong fit, as the R-squared value of 0.999, indicates that approximately 99.9% of the variability in the dependent variable can be explained by the independent variable(s) in the model. This suggests that the model is highly effective at capturing the relationship between the calculated projected results and the ground truth results. It is important to emphasize that the regression function obtained after registration on one image can be applied to other images taken from the same direction.

Given this characteristic, it is not required to perform the registration process repetitively. Instead, the derived regression models are applied to all the oblique aerial images. This approach enables us to strike a balance between efficiency and quality during the extraction process of facade texture images. However, images taken from different perspectives necessitate distinct registration processes and the subsequent acquisition of corresponding regression functions for optimization.

(a) (b)

Figure 4.6: Comparison of initial projection result and optimization result

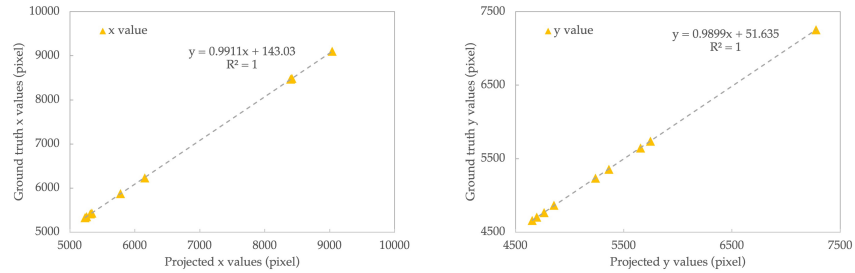
4.3.3 Image rectification

So far, the offsets between the projected values and ground truth values have been corrected, resulting in a relatively accurate 3D-2D correspondence for the same façade. The last stage of this process involves image extraction and rectification, as a façade texture image from the perspective of the front (orthophoto) is required, which helps to obtain the correct proportional relationship between the façade and openings. The optimized constraints (red rectangles in Figure 4.6) (b) are employed for extraction purposes. Rectification changes the perspective of the façade texture images, transforming the top-down perspective of the oblique aerial images themselves into a straightforward one, ensuring that the façade images accurately represent the actual building structure (Figure 4.8 (b)). The extracted images are then rectified using perspective transformation to match the size of the corresponding 3D façades. The corner coordinates of the target rectified image should be specified based on the real width and height of the corresponding 3D façade. Subsequently, the transformation matrix can be calculated using the original and target coordinate pairs to implement the perspective transformation.

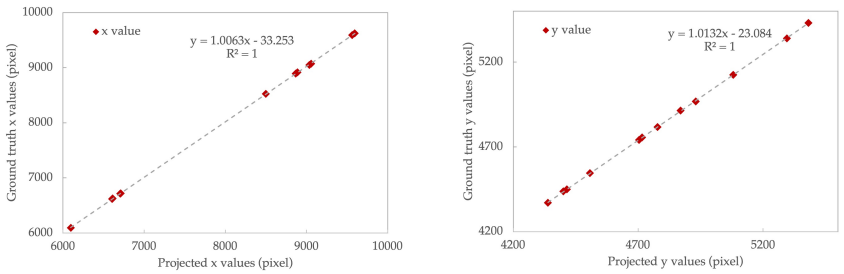
$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = H \times \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \tag{4.5}$$

$$(x', y') = (\frac{x}{w}, \frac{y}{w}) \tag{4.6}$$

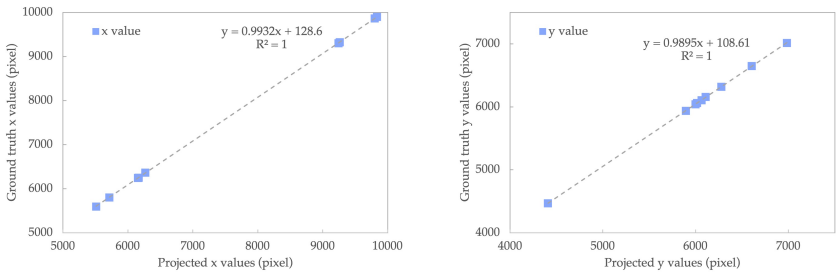
Where u, v are the coordinates of the point in the original image; x, y are the projected coordinates in the second image (not yet normalized); x', y' are the normalized coordinates in the target image, w is the normalized factor and H is the homography matrix, which contains all the information about the transformation (translation, rotation, scale, and perspective).



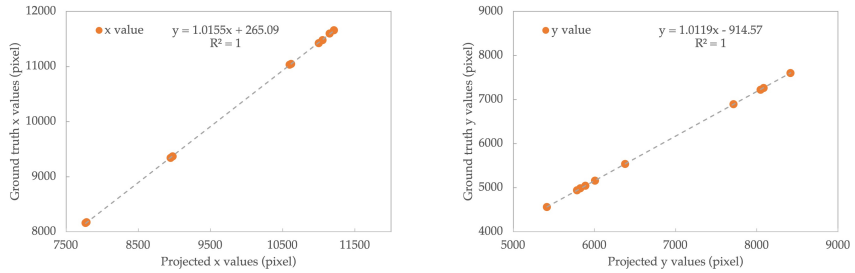
(a) Regression result of back-looking images



(b) Regression result of right-looking images



(c) Regression result of forward-looking images



(d) Regression result of left-looking images

Figure 4.7: Regression between projected and true values in four directions of view

(a) projection result (b) rectification result

Figure 4.8: Perspctive transformation

4.4 Openings detection & segmentation

In this research, the deep learning framework Mask R-CNN is chosen to perform openings detection and segmentation, and the principles and advantages of Mask R-CNN are described in 3.4.4. To enhance the performance of Mask R-CNN, we introduce ResNet-101 as the backbone architecture for feature extraction. ResNet-101 is a deep convolutional neural network architecture that belongs to Residual Network (ResNet) family [He et al. [2016]]. The innovation of ResNet is the utilization of residual connections, the architecture is constructed by stacking multiple residual building blocks. This technique helps alleviate the vanishing gradient problem and improves the overall optimization process during training, which enables the efficient training of much deeper neural networks without the degradation of performance typically encountered in deep architectures. ResNet-101 comprises 101 layers, consisting of convolutional layers, batch normalization layers, and ReLU activation layers. It has outstanding performance in image classification, object detection, and semantic segmentation tasks.

The façade training dataset is obtained from the City of Amsterdam [Amsterdam [2020]]. It is an open façade dataset containing over 900 annotated façade images in Amsterdam, and each image was manually annotated and labeled with three classes, windows, doors, and sky. The dataset is split into *train* and *val* folders with two corresponding JSON files in the MS COCO format. Among them, 820 images are used for training and 90 images are used for validation. In addition, in order to improve the performance of Mask R-CNN model, this research added 30 manually annotated openings images and merged them into a training dataset. After training the Mask R-CNN model, the detection and segmentation on a custom dataset can be performed on the extracted façade image dataset in this research.

4.5 Openings layout optimization

In practical architectural designing, most of the windows and doors are predominantly arranged in a regular pattern on the façade to enhance visual aesthetics, a factor we have taken into account in our research. However, detection outcomes may not always reflect this regularity. Therefore, there's a need for layout optimization to ensure the arrangement aligns with typical patterns. In this research, we designed an optimization process that aims to improve the overall appearance and structure of the façade elements, ensuring a more accurate and visually appealing representation. We addressed the irregular and aesthetically unappealing arrangement of façade elements obtained from the Mask R-CNN extraction process by optimizing the distribution of these elements in terms of both size and position.

The constraints applied to the optimization of openings in this research are established as follows:

- Openings with initially similar dimensions should be adjusted to have identical width and length, thereby maintaining a consistent size across comparable façade elements. This constraint ensures uniformity and conformity among similar openings within the façade.
- Openings that are originally positioned in a normal arrangement (i.e., parallel and vertical alignment) should be adjusted to align horizontally and vertically. This constraint promotes a more organized and coherent arrangement of openings, contributing to the overall visual appeal and structural accuracy of the façade.
- Openings should be designed to maintain a horizontal width and a vertical height, adhering to the conventional orientation of such elements in building façades. This constraint ensures that the façade elements adhere to standard architectural practices and principles, thereby enhancing the realism and accuracy of the façade representation.

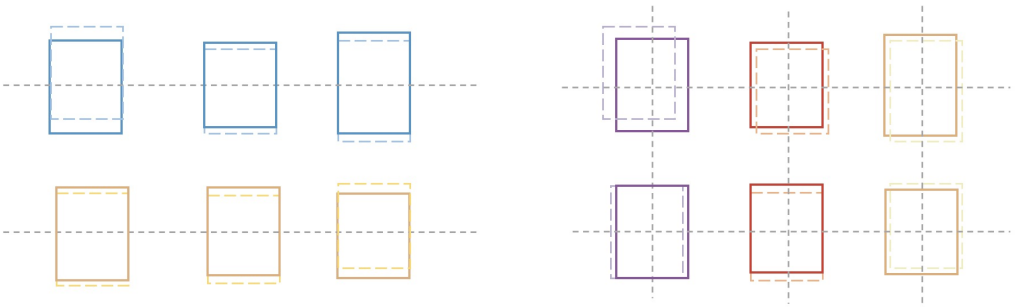
By applying these constraints during the optimization process, the resulting façade elements exhibit a more orderly, consistent, and aesthetically pleasing arrangement, ultimately improving the accuracy and visual appeal of the façade representation. The specific order of adjustment was to first adjust the position and determine the position of the centroid of each window, then adjust the size, while keeping the centroid unchanged.

4.5.1 Position regularization

The objective of position regularization is to fine-tune the position of the openings in a way that fulfills the second constraint as described in 4.5. To achieve this, we have opted to represent and adjust the position of each opening by utilizing its centroid (c_{ix}, c_{iy}). The horizontal and vertical positions are adjusted separately by the c_{ix} and c_{iy} coordinates.

To ensure alignment in the horizontal direction, we assume that openings in the same row exhibit nearly identical c_{iy} values. First, we sort the centroids in ascending order based on

their c_{iy}. Next, we determine whether the $(i+1)^{th}$ and i^{th} openings are relatively horizontal by comparing the difference between $c_{(i+1)y}$ and c_{iy} to a predefined threshold. If the difference exceeds the threshold, the windows are no longer considered part of the same row, and $c_{(i+1)y}$ is treated as the start of a new horizontal row. Subsequently, the openings with difference values within the threshold from the previous iteration are deemed to be part of the same row. We then calculate the average of the c_y values and replace the original c_{iy} values with this new average.



(a) Step 1: horizontal adjustment

(b) Step 2: vertical adjustment

Figure 4.9: The two-step operation to adjust the position: from the y and x directions

A similar approach is employed for vertical consistency, as openings in the same column exhibit nearly identical c_{ix}. Using a comparable method, we calculate the average of the c_x values for the same column and replace the original values. The threshold value is an arbitrarily determined value that accounts for the acceptable error in openings detection and segmentation. Figure 4.10 illustrates the two-step adjustment process, where the dotted line indicates the horizontal line formed by the centroids of openings within the same rows and columns. In this illustration, the dotted boxes represent the original positions of the openings, whereas the solid boxes indicate their positions after the current adjustment. Additionally, different colors signify distinct groups within the horizontal rows and vertical columns.

4.5.2 Size regularization

Upon completion of the position regularization, the centroid is maintained in a fixed position while the dimensions of the openings are modified by altering the values of the four opening corners. To achieve the first constraint, we employ the unsupervised Density-based spatial clustering of applications with noise (DBSCAN) method for classifying windows based on their length (w_{ix}) and width (w_{iy}) attributes [Ester et al. [1996]]. DBSCAN is particularly

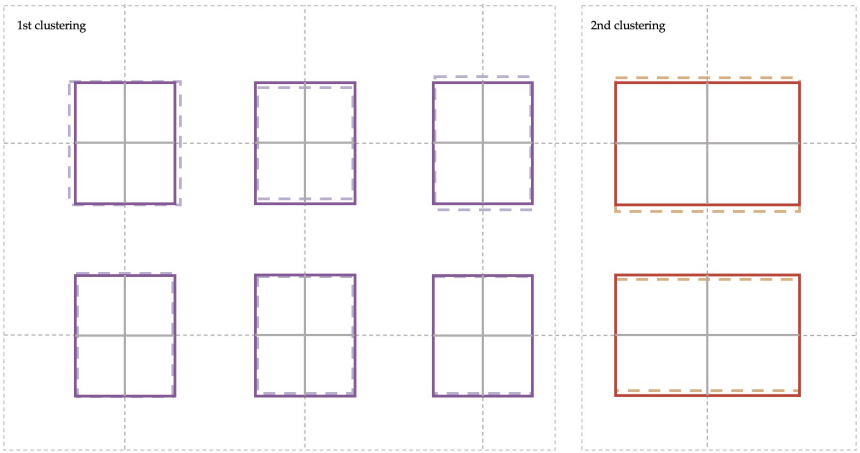


Figure 4.10: Size regularization

adept at grouping points in close proximity within a specified region according to a density criterion. The algorithm operates by determining if the density of points within a point's neighborhood surpasses a predefined threshold.

A key advantage of DBSCAN lies in its ability to efficiently cluster data points with arbitrary shapes, sizes, and densities [Ester et al. [1996]]. Furthermore, it does not necessitate the user to predetermine the number of clusters; instead, the algorithm autonomously identifies the optimal number based on the data's density. This characteristic is particularly well-suited for this research, given the uncertainty surrounding the groups of similar windows. The two main parameters required for DBSCAN are *eps* (epsilon), which defines the distance threshold for the neighborhood surrounding the data points, and *min_samples*, representing the minimum number of data points needed to form a dense region. In the context of this research, we set *eps* to 5 and *min_samples* to 1. The *min_samples* is set to 1 to accommodate cases in which only one opening exhibits a particular size.

By utilizing the DBSCAN clustering method, windows can be effectively classified based on their initial sizes. The second step is to calculate the average length and width for the openings of each class and use the average value to update the original value of each opening in this class. Since the centroid coordinates of each opening are currently fixed, the only way to modify the size of the openings is by adjusting the position of the four corners. Assume that the current centroid coordinate of the opening is (c_{ix}, c_{iy}), and the width and height of the cluster where the opening is classified are w and h respectively. The coordinates of the four corners of the opening are modified according to the upper left, lower left, lower right, and upper right are $(c_{ix}\text{-}\frac{w}{2}, c_{iy}\text{-}\frac{h}{2})$, $(c_{ix}\text{-}\frac{w}{2}, c_{iy}\text{+}\frac{h}{2})$, $(c_{ix}\text{+}\frac{h}{2}, c_{iy}\text{+}\frac{h}{2})$, and $(c_{ix}\text{+}\frac{h}{2}, c_{iy}\text{-}\frac{h}{2})$.

4.6 Conversion of 2D openings to 3D

In this research, it is assumed that the optimal matching rule for façades and openings is characterized by the following conditions: If it is desired to obtain a watertight 3D building model, a 3D opening surface is concaved inwards and lies parallel to the plane in which the façade is located, and the façade and openings are connected by four distinct connecting surfaces; if it is desired to present the openings in the form of holes, then the four corners of the openings are completely coplanar with the façade to form the holes.

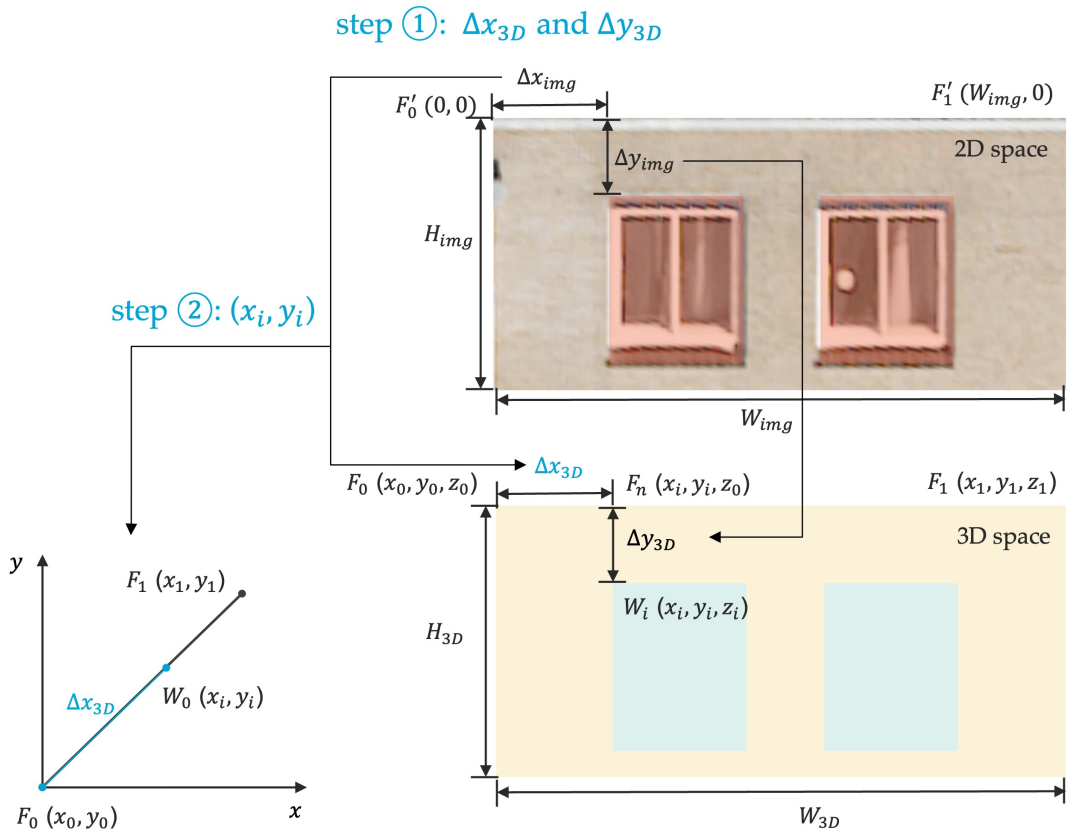


Figure 4.11: Conversion of 2D coordinates to 3D coordinates

To achieve the best possible alignment between the extracted openings and the 3D façade, we have developed a novel approach based on the similar triangle principle to transform 2D to 3D coordinates, which differs from traditional photogrammetry-based methods. The use of traditional photogrammetry-based back-projection methods could potentially reintroduce systematic errors, resulting in imprecise projection outcomes. The advantage of our approach is that it overcomes this issue, maintaining the correct relative positional relationship between the façade and the openings. This method involves combining the pixel coordinates of the opening corners with the 3D coordinates of corner points on the façade,

calculating the offset of each opening corner and image center (upper left corner), and separately computing the x, y, and z values of each opening corner by using the similarity and scaling relationship between the 2D façade and 3D façade. By employing this technique, we can effectively address the challenges associated with accurately mapping 2D opening coordinates onto their corresponding 3D positions within the façade. Because no matter which openings are expected to be presented, it can be guaranteed that the converted 3D openings and façades are coplanar, which contributes to the overall quality of the façade and opening extraction.

The details of this method are calculated as follows. Since we have rectified the façade images using the length and width of the corresponding façades in the first stage, the aspect ratio of the 2D image and the 3D façade is guaranteed to be the same. This is important because if the aspect ratios of the two do not match, the accuracy of the scaling result will be affected. In this case, the 4 corners of the 2D image correspond to the 4 corners of the 3D façade, in other words, the 4 corners of the 2D image are positioned in 3D space at the exact 3D coordinates of the corresponding corners of the 3D façade. In the example shown in Figure 4.11, there is a correspondence between F_0 and $F_0^{'}$, as well as between F_1 and $F_1^{'}$. The pixel distances of $F_{0'}$ and $F_1^{'}$ in the image represent the true distance of the 3D edge F_0 and F_1 in the 3D space. Similarly, Δx_{img} and Δy_{img} represent the 2D space offsets of the opening's corner, while Δx_{3D} and Δy_{3D} correspond to the 3D space offsets of the opening's corner relative to the upper-left origin. As a result, the Δx_{img} and Δx_{3D}, as well as Δy_{img} and Δy_{3D}, exhibit a proportional correspondence between 2D pixel distances the 3D spatial distances.

Based on the aforementioned correspondence, we first determine the z-value of W_0, which is z_i, by calculating the offset in 3D space Δy_{3D}. The actual lengths of H_{3d} and W_{3d} can be rapidly acquired from the 3D coordinates of the four known corners of the 3D façade. Using proportionality equation (4.7), we can obtain the length of Δy_{3D}, and subsequently determine the value of z_i according to equation (4.8).

$$\frac{\Delta y_{3D}}{\Delta y_{img}} = \frac{H_{3D}}{H_{img}} \qquad \Delta y_{3D} = \frac{H_{3D}}{H_{img}} \times \Delta y_{img} \tag{4.7}$$

$$z_i = z_0 - \Delta y_{3D} \tag{4.8}$$

The next step involves calculating the x_i and y_i values. To obtain these values, the length of Δx_{3D} needs to be determined first. Using a similar method for calculating Δy_{3D}, we can obtain the value of Δx_{3D} using (4.9), as illustrated in Part 2 of Figure 4.11. However, since the footprint of the façade is not parallel to any axis in the XOY plane, we need to project the calculation of x_i and y_i onto the XOY plane (as shown in Part 3 of Figure 4.11). Given that the 3D distance F_0F_1 and pixel distance of $F_0^{'}F_1^{'}$ are known and proportionally scaled, we can compute the x_i with equation (4.10) and y_i with equation (4.11) based on the corresponding scaling relationships.

$$\frac{\Delta x_{3D}}{\Delta x_{img}} = \frac{H_{3D}}{H_{img}} \qquad \Delta x_{3D} = \frac{H_{3D}}{H_{img}} \times \Delta x_{img} \tag{4.9}$$

$$\frac{x_i}{x_1 - x_0} = \frac{W_{3D}}{\Delta x_{3D}} \qquad x_i = \frac{W_{3D}}{\Delta x_{3D}} \times (x_1 - x_0) \tag{4.10}$$

$$\frac{y_i}{y_1 - y_0} = \frac{W_{3D}}{\Delta x_{3D}} \qquad y_i = \frac{W_{3D}}{\Delta x_{3D}} \times (y_1 - y_0) \tag{4.11}$$

Upon acquiring the 3D coordinates of the four corners of an opening, we proceed to re-evaluate the spatial relationship between the 3D façade and the 3D opening to ensure their co-planarity.

4.7 Integration of openings and 3D building model

To preserve the watertightness of the entire resulting 3D building model, this process opts to extrude the openings into the outer façade at a specified depth. For simplicity, a uniform depth is employed for the extrusion operation.
Upon obtaining the 3D openings based on the façade, we calculate a new opening plane situated along the inner side of the 3D façade at the specified depth. This new plane is parallel to both the original opening and the façade. Finally, we generate the connecting walls between the façade and the new opening by using the 3D coordinates of the original and extruded openings, maintaining a counterclockwise arrangement of vertices as illustrated in Figure 4.13.

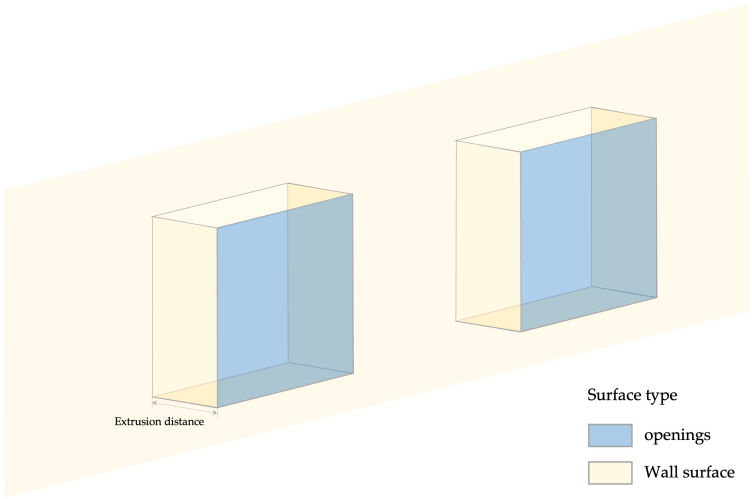


Figure 4.12: Resulting structure for façade and opening integration

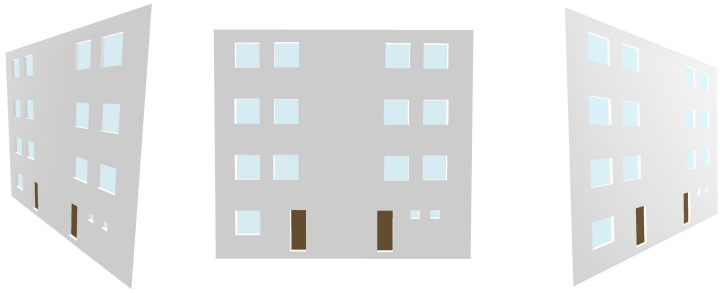


Figure 4.13: Example result in Azul

5 Implementation

This chapter provides a detailed explanation of the pipeline implementation. Section 5.1 introduces the experimental data, including the region under investigation, while Section 5.2 highlights the primary libraries and software utilized in this study. Section 5.3 delves into the parameters employed within the pipeline, along with a comparative analysis and the selection process for parameter tuning.

5.1 Datasets

This research focuses on the Almere region, specifically the northern part of Almere Centrum, a small community comprising eighteen buildings, as depicted in Figure 5.1.

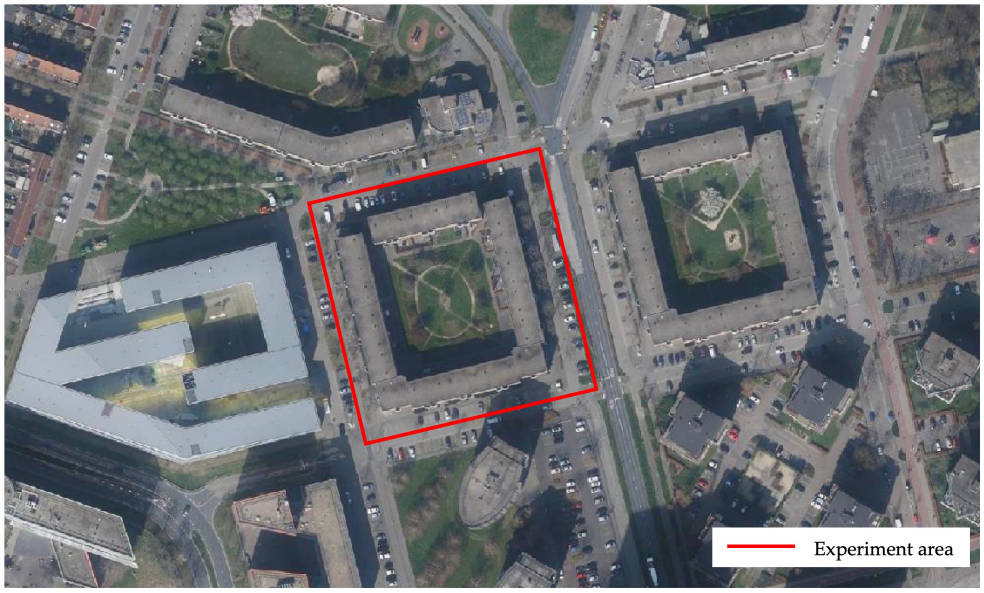


Figure 5.1: Experiment area (image source: PDOK Luchtfoto)

The study employs two sets of experimental data. The first dataset is 3D BAG LOD2.2 building models, which are open-source and can be directly acquired from the 3D BAG website. The tile covering the experimental area is identified as 3dbag_v210908_fd2cee53_4566. The second dataset is oblique aerial image data provided by the Gemeente Almere, captured

using CityMapper-2 airborne sensor systems. This dataset encompasses images taken from four different perspectives: forward-looking, back-looking, left-looking, and right-looking.

Considering the coverage of oblique aerial images in 3D space, a total of 15 images containing the experimental area were filtered and selected. These images were then imported into Pix4D for camera parameter estimation, which provided the camera intrinsic matrix, rotation matrix, and translation vector for each image. These parameters are crucial for the subsequent perspective projection process.

5.2 Libraries and software

The main libraries used in this thesis are listed as follows:

- Computational Geometry Algorithms Library (CGAL) [Oesau et al. [2023]]: the CGAL shape detection library is utilized to implement the Region growing algorithm in C++. It aims to solve the co-planar problem of 3D building models. It enables us to effectively identify co-planar surfaces and color them with the same colors.
- Detectron2 [Wu et al. [2019]]: Dectron2 is an open-source computer vision Python framework developed by Facebook AI Research, which is built based on PyTorch [Paszke et al. [2019]]. Detectron2 provides Mask R-CNN implementation and evaluation for this research.
- Open Source Computer Vision Library (OpenCV) [Bradski [2000]]: OpenCV is an open-source computer vision library providing functions and tools for image processing. The research employs the OpenCV-Python library to implement perspective projection, perspective transformation, and visualization in the second stage of the pipeline.
- Scikit-learn [Pedregosa et al. [2011]]: Scikit-learn is an open-source Python library providing supervised and unsupervised machine learning algorithms. In this study, we utilize the DBSCAN algorithm for façade layout optimization and the Least Squares Regression algorithm for projection offset optimization.

The software used in this research:

- COCO-annotator [Brooks [2019]]: COCO-Annotator is an open-source web-based image annotation tool, which supports multiple annotation types including bounding boxes, segmentation masks, etc. In this study, COCO-annotator is employed to create, label, and generate a Common Objects in Context (COCO) format dataset for the Mask R-CNN.
- Pix4D: Pix4D is a photogrammetry software that provides aerial triangulation services, point cloud generation, and 3D model generation functionalities. In this study, the aerial triangulation function is employed to obtain the camera intrinsic and extrinsic parameters.

- Azul [Arroyo Ohori [2020]]: Azul is an open-source 3D city model viewer. In this research, Azul is utilized to visualize the resulting LOD3 CityJSON model.
- Val3dity [Ledoux [2018]]: val3dity is an open-source tool to validate 3D primitives. Val3dity is employed to validate the resulting LOD3 CityJSON model in the last stage of the research.

5.3 Parameters tuning

5.3.1 Region growing parameters tuning

While applying the region growing algorithm, the *similarity* between surfaces is calculated based on three properties:

- the maximum distance from the furthest face vertex to a plane;
- the maximum accepted angle between the face normal and the normal of a plane;
- the minimum number of mesh faces that a region must have.

The objective of using the region-growing algorithm to address the co-planar issue is to merge wall surfaces that are originally in the same plane into a single face. Thus, the maximum accepted angle is a critical parameter, and this condition must be set stringently to obtain accurate results, ensuring that truly co-planar faces can be fused. The maximum distance is set to a larger value because it does not significantly impact the test results. However, if set too small, it may cause originally co-planar surfaces to be misjudged as non-coplanar surfaces. Since the façade is generally a rectangle, after being triangulated it is usually composed of more than or equal to 2 triangles, the minimum number is set to 2. Therefore, after several rounds of experimenting with different parameters, the optimal three values are set to 10, 10, and 2, respectively, to strike a balance between accuracy and performance.

5.3.2 Mask R-CNN hyperparameters tuning

The Mask R-CNN model was trained using the Amsterdam façade dataset on Google Colab. To optimize the performance of Mask R-CNN for object detection, hyperparameter tuning is essential. Key hyperparameters include:

- Learning rate: learning rate (LR) is a hyperparameter that controls how fast the model learns from the training data. A lower learning rate will lead to a slower training process but may lead to better performance.
- Batch size: batch size defines the number of training samples used in an iteration. A smaller batch size requires more iterations to converge, while a larger batch size can lead to more stable gradient updates and fewer iterations to converge.

- RoI Heads Batch Size Per Image: it controls the number of RoI sampled in one image during training for RoI heads.
- Iteration: an iteration time refers to a single update step in the training process. During each iteration, the model will process a batch of data, compute the gradients, and update the weights based on the gradients.

LR between 0.00025 to 0.001 is proved to be the most effective, thus it is set to 0.00025 in this research. The batch size is set to 2, and the Mask R-CNN model is firstly trained for a total of 10,000 iterations to observe the convergence. The accuracy and total loss during the training process are shown in Figure 5.3.

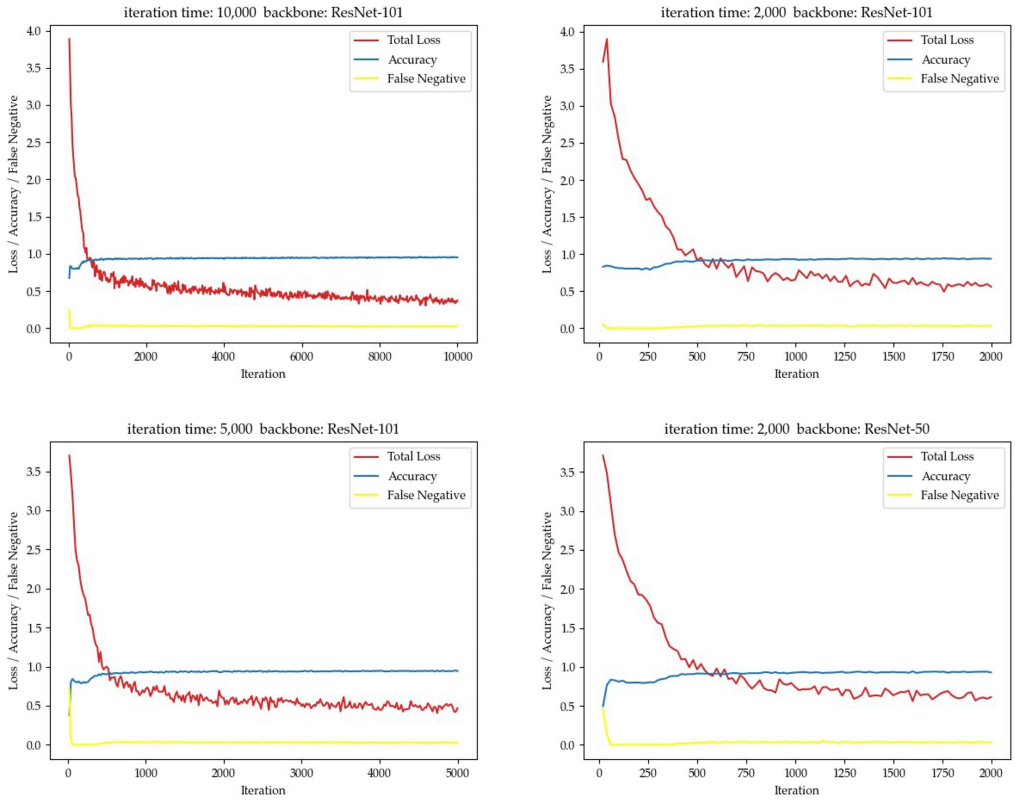


Figure 5.2: Mask R-CNN accuracy and loss with different iteration times and backbone

Overfitting occurs when the supervised machine learning model learns the training data exceptionally well but fails to fit and generalize effectively to unseen datasets [Ying [2019]]. To mitigate overfitting, solutions such as early stopping and L1 or L2 regularization are usually employed. Observations from Figure 5.3 reveal that the total loss plateaus after approximately 1,000 iterations, and the values of accuracy, loss, and FN tend to converge. To avoid overfitting, the iteration times were adjusted to 2000, while maintaining LR at 0.00025. A comparative analysis was also conducted to evaluate the performance of different back-

bones in the context of the Mask R-CNN model including ResNet-50 and ResNet-101. The distinction between the two is the depth of the networks: ResNet-101 has a deeper architecture with 101 layers, as opposed to ResNet-50's 50 layers. Deeper architectures can contribute to improved performance in more complex tasks. The results of the comparison were evaluated using AP. As illustrated in Table 5.1, the AP of ResNet-101 surpasses that of ResNet-50, although the difference is not substantial. It suggests that while the deeper architecture of ResNet-101 offers some performance benefits, the improvement is not highly pronounced in this research.

Table 5.1: Comparison of AP of ResNet-50 and ResNet-101 with 2,000 iterations(%)

Backbone	Type	AP	$AP_{windows}$	AP_{doors}
ResNet-50	AP_{segm}	72.441	56.290	**56.639**
ResNet-50	AP_{bbox}	71.472	64.177	55.382
ResNet-101	AP_{segm}	**73.201**	65.731	56.171
ResNet-101	AP_{bbox}	72.326	**65.737**	55.133

Table 5.2: Comparison of 2,000 iterations and 5,000 iterations with confidence threshold 0.8 (%)

Iteration	Type	AP	$AP_{windows}$	AP_{doors}
5000	AP_{segm}	75.492	67.928	61.708
5000	AP_{bbox}	72.793	65.146	57.990
2000	AP_{segm}	73.201	65.731	56.171
2000	AP_{bbox}	72.326	65.737	55.133

To demonstrate the reasonableness of the selection of Mask R-CNN in this study, the performance of Mask R-CNN and Faster R-CNN on the same dataset was tested, as shown in Table 5.3:

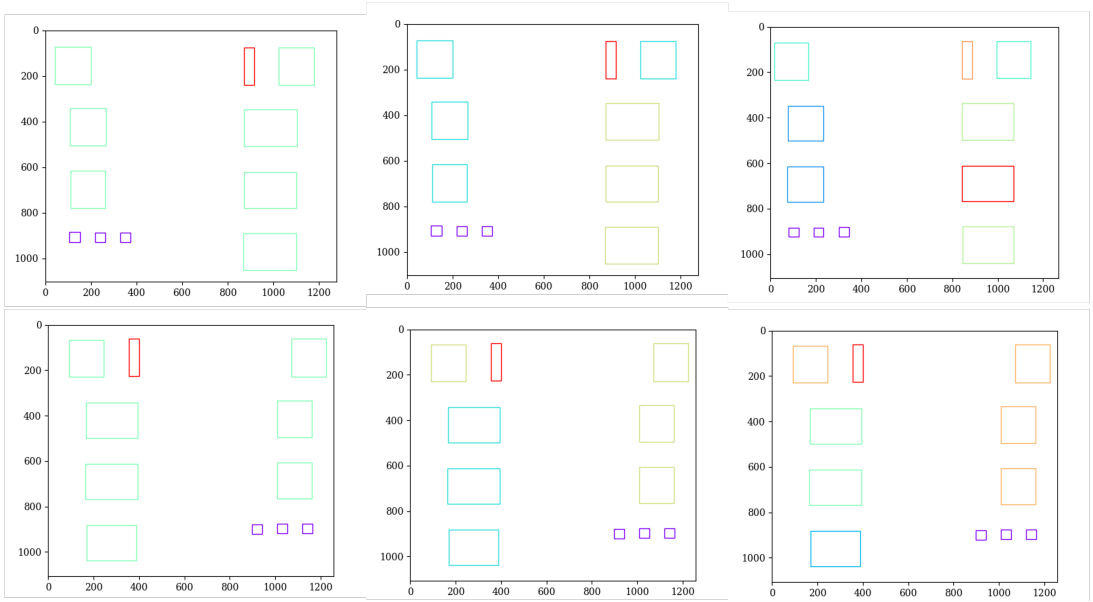
Table 5.3: Comparison of Mask R-CNN and Faster R-CNN using Amsterdam façade dataset (%)

Model	AP	$AP_{windows}$	AP_{doors}
Mask R-CNN	72.326	**67.928**	**61.708**
Faster R-CNN	**73.106**	65.569	57.671

In conclusion, the ultimate configuration of the Mask R-CNN model employed in this study utilizes the ResNet-101 backbone, a LR of 0.00025, and is trained for 5,000 iterations.

5.3.3 DBSCAN parameters tuning

Two key parameters are *eps* and *min_samples* in DBSCAN algorithm:



Eps = 70 *Eps* = 50 *Eps* = 10

Figure 5.3: Clustering of openings with different *eps*

- *eps*: it defines the maximum distance between two points to perform a cluster;
- *min_samples*: it determines the minimum number of the points required to form a cluster;

In this study, a series of experiments were conducted to iteratively determine the optimal *eps* value for the clustering algorithm. It was found that an *eps* value in the range of 10-50 yielded the best results. If the *eps* value was set too large, openings with significantly different sizes would be incorrectly clustered together. Conversely, if the *eps* value was set too small, openings of similar sizes would not be grouped together, resulting in an overly strict clustering criterion that fails to account for potential errors in the detection step. Ultimately, an *eps* value of 50 was selected as it provided a reasonable tolerance for errors while effectively separating openings of distinctly different sizes. It is reasonable to have a unique size of the opening on a façade, the *min_sample* parameter was set to 1.

6 Results and Evaluations

6.1 Result of each stage of the pipeline

6.1.1 Co-planar surfaces merging

Firstly, we addressed the co-planar issue in 3D BAG. After testing various parameter combinations (see detailed parameters in 5.3.1), the parameter set of 10-20-2 based on the experimental requirements is selected. Our goal was to merge adjacent surfaces that share the same plane as much as possible while also considering the results of multiple trials. By calculating the normal vector, we remove the roofs that do not participate in the pipeline and keep only the façade of the building.

Figure 6.1 illustrates the three stages of 3D BAG LOD2.2 building models: the initial raw data, the state after the co-planar merging process, and the state with the roof removed and retaining only the façades. The original dataset consists of a total of 1376 faces, which is reduced to 224 faces following the co-planar merging, and further decreases to 183 faces after removing the roof and ground surfaces. It is evident that the 3D building model surface reduction process effectively filters out unnecessary surfaces, leading to a significant decrease in the amount of time and computation resources required for processing, as the whole pipeline is carried out surface by surface. Consequently, this results in an enhanced efficiency of the entire pipeline, enabling a more streamlined and optimized execution of the subsequent steps in the workflow.

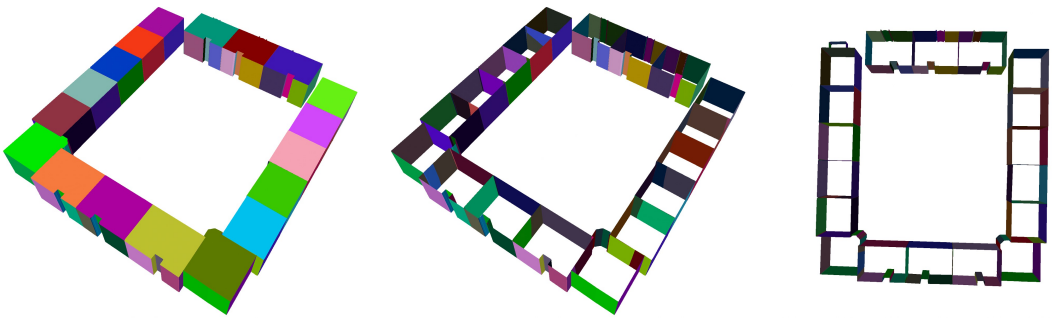


Figure 6.1: Surface of 3D building models reduction

6.1.2 3D building model projection, registration, and rectification

Upon completion of the previously described pre-processing steps, we have amassed adequate data to proceed with the next stage of the pipeline. In the façade extraction stage, the projection matrix for each image is constructed utilizing the camera matrix, rotation matrix, and translation vectors, enabling the perspective projection. Façade extraction is conducted on a face-by-face basis. This study does not consider the visibility of façades in particular images; therefore, all façades will be projected onto the image for each image capture direction. Figure 6.2 presents shows the result of projecting the 3D façades of the buildings in the experimental area onto four 2D images with different orientations by perspective projection.

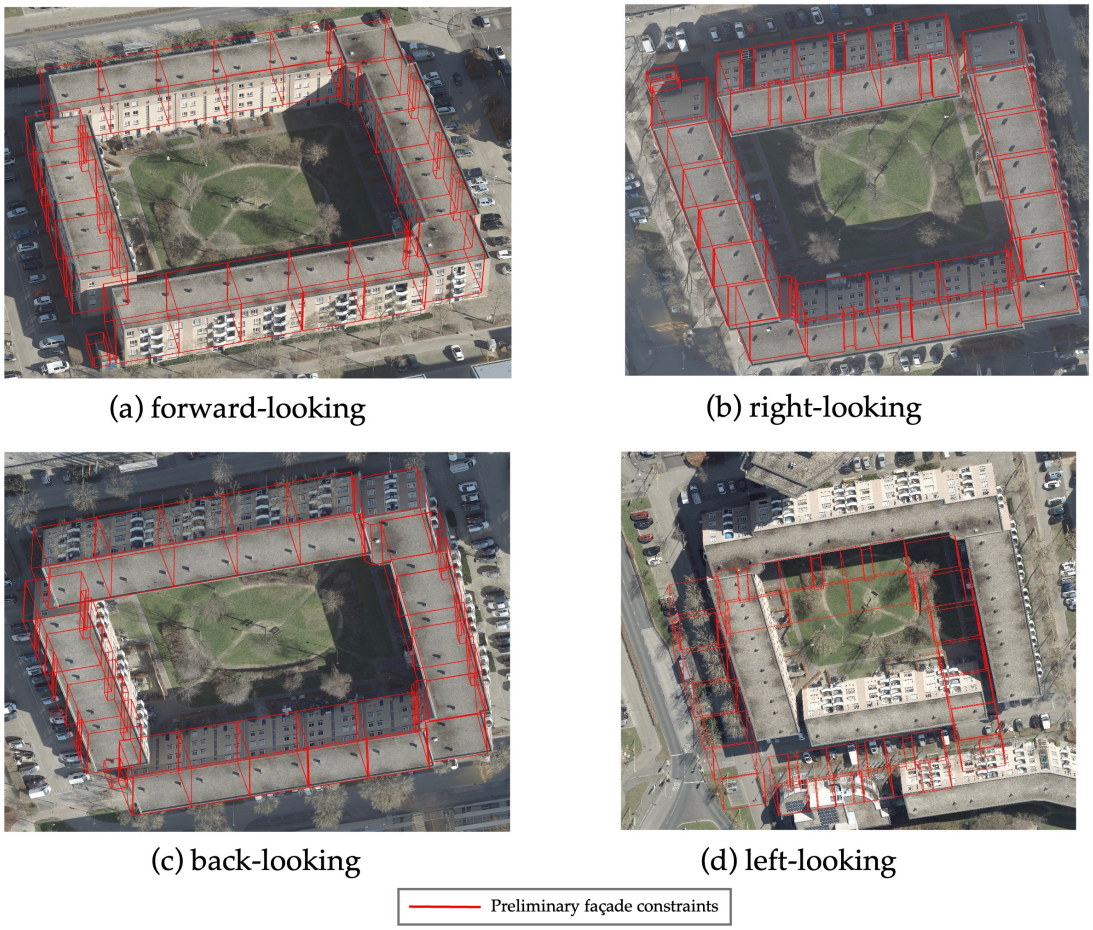


Figure 6.2: Initial projection result using 3D BAG façades

The differences in projection errors for buildings within the same range across different orientations are notable. For instance, the projection errors for left-looking images are significantly larger than those observed in the other three orientations. Initially, these offsets and differences between offsets were speculated to be a consequence of inaccuracies in the

camera parameter estimation stage. These offsets were effectively eliminated by the designed registration model (refer to Figure 4.7), by utilizing a regression equation derived using least squares regression. The optimized projection results in four orientations are shown respectively in Figure 6.3, where the red line is the preliminary projection result and the blue line is the optimized result. It can be observed that the offsets are eliminated in all four directions of the images, and the blue line fits very well with the real boundary of the façade.

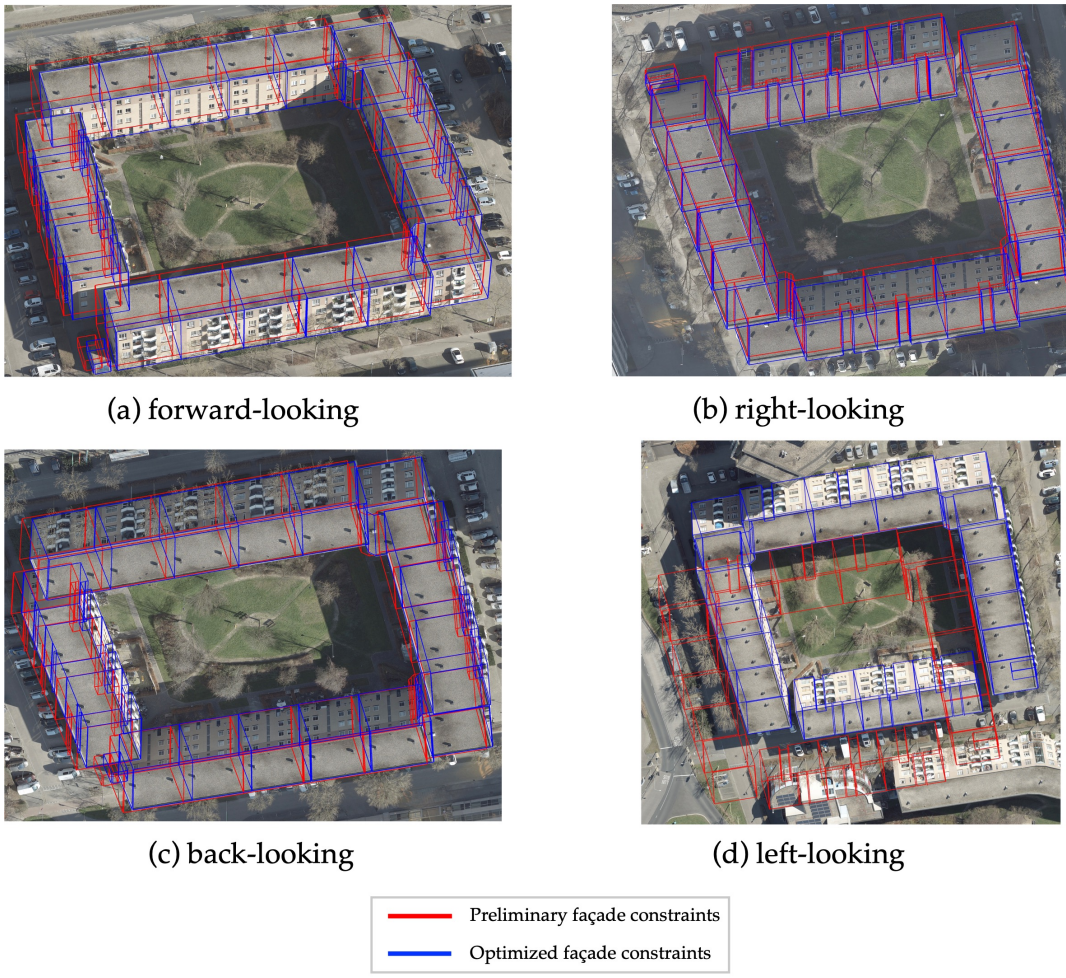


Figure 6.3: Optimized façade extraction constraints

The optimized façade extraction constraints obtained in 6.3 are utilized to extract and correct façade images. Since the 3D façade and 2D façade images maintain a strong connection throughout the process, it is straightforward to correct the true aspect ratio of the images based on the 3D façade. After the extraction and rectification, the extracted façade image results from four directions exhibit a distinct pattern: Regular image results can be obtained from photos taken facing the camera orientation and those taken facing the opposite direc-

tion of the camera orientation. However, in the other two orientations, no results can be extracted. Figure 6.4 lists two extractions, a façade facing the forward-looking camera and a façade facing the left-looking camera. Therefore, it is evident that selecting the correct façade from the 4 extraction results and performing openings extraction is not a complicated task, since the information cannot be extracted from the other 3 images at all. Statistical analysis was done for the projection and extraction results, counting the façades that are directly visible on each oblique aerial image facing the camera, the façade images that can be extracted through this pipeline, and the percentage of matches between the two, as shown in 6.1. It is evident that this method can extract 100% of the façades that can be seen in an orientation and filter out the façades that cannot be seen. Moreover, utilizing oblique aerial images as the data source not only the outer façades can be extracted, but even the inner façades can be extracted as well. This is where oblique aerial images have the advantage of capturing part of the interior façade of a building, the part that cannot be captured by SVI.

Figure 6.4: Special pattern of façade extraction result

Table 6.1: Statistics on the number of façade extractions

Image direction	expected façade number	actual façade number	completeness
Back-looking	12	12	100%
Left-looking	20	20	100%
Forward-looking	13	13	100%
Right-looking	12	12	100%
Total	57	57	100%

6.2 Openings detection results

6.2.1 Analysis of openings detection result

The factors contributing to extraction failure can be summarized as follows:

- Obstructions in front of the façade, such as trees, can impede the accurate identification of architectural features.
- Large balconies on the façade may obstruct the detection of windows and doors situated below, leading to inadequate recognition.
- The physical state of the windows, such as the presence of multiple glass panes, can cause issues in the extraction process. For instance, if a window consists of two panes of glass, the window's edge might lead to the perception of two separate windows instead of a single unit.

Figure 6.5: Successful openings detection result using Mask R-CNN

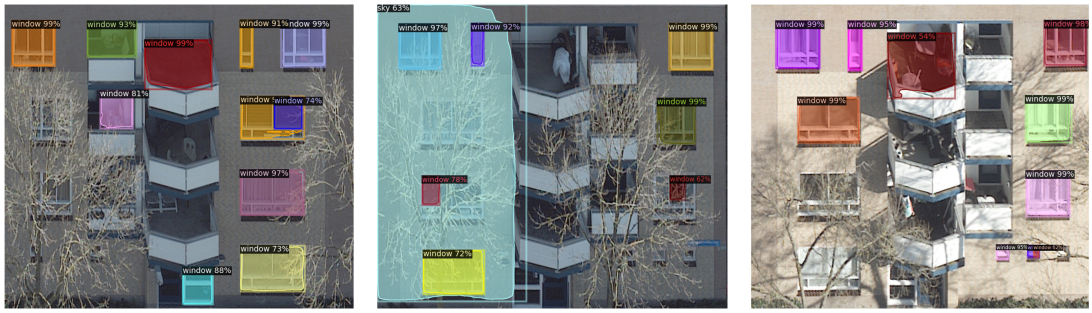


Figure 6.6: Unsuccessful extraction cases using Mask R-CNN

6.3 Openings Layout optimization results

Figure 6.7 shows an initial detection and extraction result with Mask R-CNN, and the result by applying the openings layout optimization algorithm, with different colors representing different clustering results obtained through DBSCAN algorithm. It demonstrates that DBSCAN effectively groups openings based on their dimensions, successfully distinguishing them by their length and width.

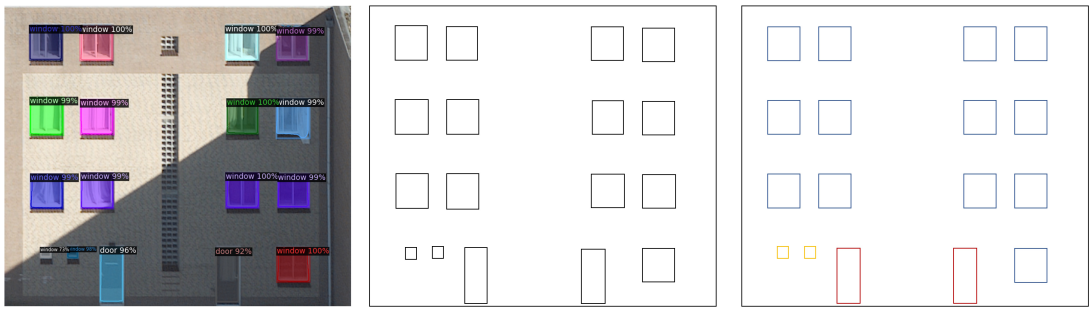


Figure 6.7: Optimized layout of the openings

6.4 LOD3 building model reconstruction results

As outlined by Chapter 4, the reconstruction target consists of 18 adjacent buildings in the Almere area, with input oblique aerial images captured from four different directions. The entire pipeline is divided into three subtasks and their corresponding outputs. The first stage generates rectified façade images, the second stage produces openings detection results in the COCO format, and the third stage yields the final LOD3 model. The resulting LOD3 model is represented as a CityJSON file, which successfully passed the val3dity validation test. During the creation of this CityJSON file, semantic information associated with each surface is maintained, thus enhancing the overall completeness of the final 3D city model.

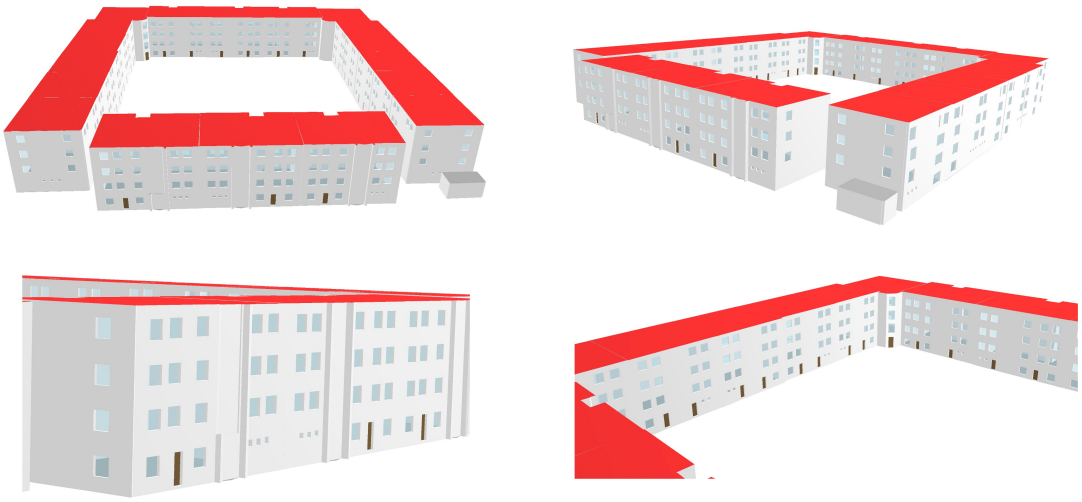


Figure 6.8: Resulting LOD3 models

This representation ensures the model's effective utilization in various urban planning and analysis applications. The resulting model is visualized in Azul, displaying both overall and detailed results in Figure 6.8. The openings exhibit regular arrangement on the façade, with the structure of the openings recessed inward. Connected polygons between the windows and exterior walls maintain watertightness, featuring Wallsurface semantics.

Traditional reconstruction methods employing street view images typically capture only the outward-facing façade of a building, making it challenging to obtain detailed information on inward-facing façades as well as roofs. This may result in incomplete LOD3 models. The proposed method utilizes building models generated by TLS to address the issue of incomplete preliminary LOD1 or LOD2 models, while also leveraging oblique aerial images to capture both interior and exterior building features. Since there is no need to rebuild the model, this approach is highly effective for upgrading existing large-scale LOD2 models to LOD3 models.

We have conducted performance evaluations of this pipeline on a larger building dataset, which is the buildings located in the oblique aerial imagery involved in the camera parameters estimation before. As illustrated in Figure 6.9, the process demonstrates robustness and efficiency. Moreover, the produced results align well with the stipulations of Level of Detail 3 (LOD3) building model standards. The consistent performance and compliance of our approach underscore its utility and effectiveness in practical applications.

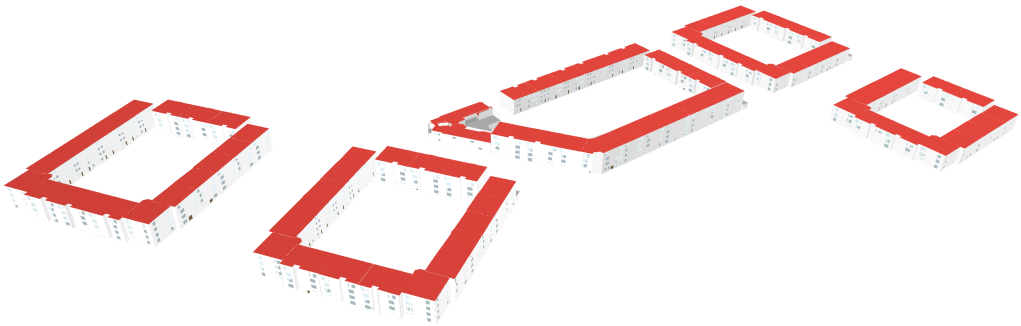


Figure 6.9: Pipeline test on a larger dataset

6.5 Impact of oblique aerial data quality

In this research, we employed oblique aerial imagery data sourced from Amsterdam and Rotterdam. During the camera parameters estimation phase, we discerned that data quality remains an essential determinant for the success of our pipeline.

7 Conclusions and future work

This chapter is a review and summary of the entire study and provides answers to the research questions and research objectives defined at the beginning of this thesis. The limitation of this pipeline is also presented in 7.3.

7.1 Research overview

Firstly, to achieve the objective of this research ***Upgrade the 3D BAG LOD2.2 building model to LOD3 by extracting openings information from oblique aerial images.***, we present a pipeline to automatically generate LOD3 building models by leveraging the power of photogrammetry and deep learning techniques. The pipeline requires the data pre-processing first, then employs perspective projection and least squares regression to extract accurate façade images from oblique aerial images, and utilizes Mask R-CNN to detect and extract openings from façade images. These extracted openings are then seamlessly integrated into the 3D BAG LOD2 building model and extruded to a specific depth, ultimately resulting in the watertight and valid LOD3 model.

How to identify all the façade texture images of 3D façades from oblique aerial images, and maximize the number of extractable façades?

To effectively extract 3D façades from 2D oblique aerial images without specific camera information and establish a connection between them, our approach involves estimating the camera parameters for each image using specialized software. Then, perspective projection is performed on the 3D façades individually. This process is conducted sequentially for oblique aerial images captured from four different directions, thereby obtaining a comprehensive set of the 3D façade - 2D façade texture image pairs. To maximize the number of extractable façades, we project all façades onto images in each direction and subsequently perform openings detection. This method ensures a more complete extraction and mapping of 3D façades to their corresponding 2D aerial images.

How to address the potential systematic errors between 3D BAG and oblique aerial images?

Systematic errors indeed exist between the 3D BAG and oblique aerial images. To address this issue, we employ a data registration process. By using a translation registration model and applying the least squares regression method, we obtain a linear relationship between the projected results and ground truth values, as well as the translation values. It is important to note that the translation model varies in each direction. By calculating the regression

functions for all four directions, we achieve the projection results for the final projected façade and image high alignment, effectively minimizing the impact of systematic errors. This method is universally applicable and can be readily applied to other images involved in the camera parameter estimation process. Utilizing the same set of registration models allows for consistent and effective registration across various image datasets, enhancing the applicability and robustness of the method.

How can openings be detected and extracted from façade texture images?

In this study, the Mask R-CNN model with ResNet-101 and FPN as the backbone is utilized to automatically detect and extract openings. We combined 800 images from the Amsterdam façade dataset with 30 manually annotated façade images from our study for training and used 90 images for validation. After 5000 iterations of model training and with a confidence threshold of 0.8, the average precision for window detection reached 75%. We also compared the performance of ResNet-101 and ResNet-50, ultimately selecting ResNet-101 due to its slightly better performance.

How to optimally integrate extracted 2D openings with 3D building models?

To address this problem, we leverage the inherent characteristics of the data and utilize a correspondence-based approach between the 2D image and 3D façade coordinates, rather than relying on photogrammetry-based back projection, which may reintroduce systematic errors. We establish a correspondence between the coordinates of the four corner points in both 2D and 3D spaces. By calculating the relative pixel coordinates of the openings in the 2D images, using the known corner points as the origin, we can determine their 3D coordinates on the corresponding 3D façades according to the scaling relationship. To ensure the watertightness of the resulting model, we extrude the openings inward from the façade to a specified depth and generate four new polygons to fill the gaps between the façade and the openings. The resulting LOD3 model successfully passes the val3dity test.

7.2 Contributions

This study contributes to the upgrading of the LOD2 building models to the LOD3 building models on a large scale:

- **Façade texture imagery extraction.** This study substantiates the efficacy of our proposed perspective projection technique for façade texture image extraction. This method ensures maximal capture of each façade texture image, thereby enhancing the quality of our results.
- **Data registration between various data sources.** By leveraging LSR, we have effectively minimized the errors between distinct data sources. The universality of the regression model enables efficient application in large-scale reconstruction, significantly reducing the need for manual error correction and eliminating process redundancy.

- **LOD3 building model generation in a larger area**: This research introduces a novel method to convert 2D openings into 3D using the principle of similar triangles and integrates them into LOD2 building models as recessed openings. The resulting watertight LOD3 building models not only demonstrate the effectiveness of our approach but also effectively circumvent the issues of data offset from different sources. Given its high automation level, our methodology holds significant promise for large-scale LOD3 building model generation. This could potentially aid in the nationwide generation of LOD3 models for the 3D Base Register of Addresses and Buildings (3D BAG) in the Netherlands.
- **Inward and outward facade reconstruction**: This process enables the extraction and reconstruction of facade texture images from the outward and inward aspects of the building. It is not feasible when using street view imagery, which generally allows for external reconstruction only.

7.3 Limitations

We also found some limitations of this process:

- Although balconies are important façade elements, the focus of this study is on openings (including windows and doors), and therefore, balconies are not identified. In future research, the inclusion of balconies could be considered to provide a more comprehensive integration of façade elements.
- Oblique aerial images have limitations: image quality significantly influences the results, and the imaging angle may cause occlusions between façade elements. For instance, prominent balconies can obstruct windows and doors, reducing the accuracy of the extracted LOD3 model. Obstructions, such as trees, can also impact opening extraction results.
- The Mask R-CNN training dataset is crucial for accurate opening extraction, as the façade types and styles in the dataset impact the results, especially in the Netherlands where the façade style is very diverse. To apply the model on a broader scale, the training dataset can be collected by the city and should include most façade types of the city.
- This study did not account for potential missed openings during the detection step while performing the openings layout optimization process. In future research, a more sophisticated model could be employed to explore and optimize façade layouts by adding potential missed openings.
- One of its limitations is the requirement for manual intervention in the data registration process. The existing registration model does not yet support complete automation, which leaves room for future advancements in this step.

7.4 Future works

Based on the limitations we discussed above, we propose some future works in this section, which could improve the quality of the resulting LOD3 building models:

- **Integration of oblique aerial imagery and SVI**: By leveraging these two types of imagery—captured from diverse perspectives—we can obtain a more detailed structure of building façades. This holistic approach not only addresses the occlusion issue but also enhances both the quantity and quality of the extracted openings, thereby refining the resulting LOD3 building model.
- **Incorporation of more LOD3 model elements**: A promising direction for future research would be the integration of several existing techniques for LOD3 model generation. This could involve combining the current methods of openings, dormers, windows, and chimneys detection [Apra [2022]] and balcony detection into a unified pipeline. This comprehensive approach could result in the generation of more intricate and detailed LOD3 building models, thereby advancing their accuracy and fostering greater realism in built environment simulations.
- **Optimization of visible façade detection**: In this research, we proceeded with the extraction of all façades present in a single image, followed by a filtering process based on the detectability of openings. In future work, this process can be improved by deploying an algorithm designed to detect façades that can be captured by a specific camera position and orientation. Such an approach would have the added benefit of narrowing down the scope of opening detection and thereby enhancing the overall efficiency of the pipeline.
- **Automation of data registration**:

Bibliography

Akmalia, R., Setan, H., Majid, Z., Suwardhi, D., and Chong, A. (2014). Tls for generating multi-lod of 3d building model. 18(1):012064.

AlHalawani, S., Yang, Y.-L., Liu, H., and Mitra, N. J. (2013). Interactive facades analysis and synthesis of semi-regular facades. 32(2pt2):215–224.

Amsterdam (2020). Amsterdam facade dataset. Accessed on April 1, 2023.

Apra, I. (2022). Semantic segmentation of roof superstructures.

Arayici, Y. (2007). An approach for real world data modelling with the 3d terrestrial laser scanner for built environment. *Automation in construction*, 16(6):816–829.

Arroyo Ohori, G. (2020). azul: A fast and efficient 3d city model viewer for macos. *Transactions in GIS*, 24(5):1165–1184.

Batty, M., Chapman, D., Evans, S., Haklay, M., Kueppers, S., Shiode, N., Smith, A., and Torrens, P. M. (2001). Visualizing the city: communicating urban design to planners and decision-makers.

Biljecki, F. (2017). Level of detail in 3d city models.

Biljecki, F., Ledoux, H., and Stoter, J. (2016). An improved lod specification for 3d building models. *Computers, Environment and Urban Systems*, 59:25–37.

Biljecki, F., Stoter, J., Ledoux, H., Zlatanova, S., and Çöltekin, A. (2015). Applications of 3d city models: State of the art review. *ISPRS International Journal of Geo-Information*, 4(4):2842–2889.

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

Brooks, J. (2019). COCO Annotator. https://github.com/jsbroks/coco-annotator/.

Bruno, N. and Roncella, R. (2019). Accuracy assessment of 3d models generated from google street view imagery. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-2/W9:181–188.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J., Layton, R., Vanderplas, J., Joly, A., Holt, B., and Varoquaux, G. (2013). Api design for machine learning software: experiences from the scikit-learn project.

Bullinger, S., Bodensteiner, C., and Arens, M. (2021). 3d surface reconstruction from multi-date satellite images.

Chen, K., Lu, W., Xue, F., Tang, P., and Li, L. H. (2018). Automatic building information model reconstruction in high-density urban areas: Augmenting multi-source data with architectural knowledge. *Automation in Construction*, 93:22–34.

Dobson, D. (2023). Floor count from street view imagery using learning-based façade parsing.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, page 226–231. AAAI Press.

Geiger, A., Benner, J., and Haefele, K. H. (2015). Generalization of 3d ifc building models. pages 19–35.

Gruen, A., Schubiger, S., Qin, R., Schrotter, G., Xiong, B., Li, J., Ling, X., Xiao, C., Yao, S., and Nuesch, F. (2019). Semantically enriched high-resolution lod 3 building model generation. *The International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42:11–18.

Haala, N., Rothermel, M., and Cavegn, S. (2015a). Extracting 3d urban models from oblique aerial images. In *2015 Joint Urban Remote Sensing Event (JURSE)*, pages 1–4.

Haala, N., Rothermel, M., and Cavegn, S. (2015b). Extracting 3d urban models from oblique aerial images. In *2015 Joint Urban Remote Sensing Event (JURSE)*, pages 1–4. IEEE.

Habbecke, M. and Kobbelt, L. (2012). Automatic registration of oblique aerial images with cadastral maps. In Kutulakos, K. N., editor, *Trends and Topics in Computer Vision*, pages 253–266, Berlin, Heidelberg. Springer Berlin Heidelberg.

He, K., Gkioxari, G., Dollar, P., and Girshick, R. (2017). Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

He, Y., Liao, W., Hong, H., and Huang, X. (2023). High-precision single building model reconstruction based on the registration between osm and dsm from satellite stereos. *Remote Sensing*, 15(5).

Hensel, S., Goebbels, S., and Kada, M. (2019). Facade reconstruction for textured lod2 citygml models based on deep learning and mixed integer linear programming. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 4.

Hu, H., Wang, L., Zhang, M., Ding, Y., and Zhu, Q. (2020). Fast and regularized reconstruction of building facades from street-view images using binary integer programming. *arXiv preprint arXiv:2002.08549*.

Huang, H., Michelini, M., Schmitz, M., Roth, L., and Mayer, H. (2020). Lod3 building reconstruction from multi-source images. *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 43.

Jiang, H., Nan, L., Yan, D.-M., Dong, W., Zhang, X., and Wonka, P. (2016). Automatic constraint detection for 2d layout regularization. *IEEE Transactions on Visualization and Computer Graphics*, 22(8):1933–1944.

Kakooei, M. and Baleghi, Y. (2023). Fusion of vertical and oblique images using intra-cluster-classification for building damage assessment. *Computers and Electrical Engineering*, 105:108536.

Leberl, F., Irschara, A., Pock, T., Meixner, P., Gruber, M., Scholz, S., and Wiechert, A. (2010). Point clouds: Lidar versus 3d vision. *Photogrammetric Engineering & Remote Sensing*, 76(10):1123–1134.

Ledoux, H. (2018). val3dity: validation of 3d gis primitives according to the international standards. *Open Geospatial Data, Software and Standards*, 3(1):1–12.

Ledoux, H. and Meijers, M. (2011). Topologically consistent 3d city models obtained by extrusion. *International Journal of Geographical Information Science*, 25:557–574.

Li, W., Meng, L., Wang, J., He, C., Xia, G.-S., and Lin, D. (2021). 3d building reconstruction from monocular remote sensing images. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 12528–12537.

Li, Y., Hu, Q., Wu, M., Liu, J., and Wu, X. (2016). Extraction and simplification of building façade pieces from mobile laser scanner point clouds for 3d street view services. *ISPRS International Journal of Geo-Information*, 5(12).

Liu, C., Kim, K., Gu, J., Furukawa, Y., and Kautz, J. (2019). Planercnn: 3d plane detection and reconstruction from a single image.

Liu, H., Xu, Y., Zhang, J., Zhu, J., Li, Y., and Hoi, S. C. H. (2020a). Deepfacade: A deep learning approach to facade parsing with symmetric loss. *IEEE Transactions on Multimedia*, 22(12):3153–3165.

Liu, L., Ouyang, W., Wang, X., Fieguth, P., Chen, J., Liu, X., and Pietikäinen, M. (2020b). Deep learning for generic object detection: A survey. *International journal of computer vision*, 128:261–318.

Micusik, B. and Kosecka, J. (2009). Piecewise planar city 3d modeling from street view panoramic sequences. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2906–2912. IEEE.

Nan, L., Jiang, C., Ghanem, B., and Wonka, P. (2015). Template assembly for detailed urban reconstruction. In *Computer Graphics Forum*, volume 34, pages 217–228. Wiley Online Library.

Nan, L., Sharf, A., Zhang, H., Cohen-Or, D., and Chen, B. (2010). Smartboxes for interactive urban reconstruction.

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

Nan, L. and Wonka, P. (2017b). Polyfit: Polygonal surface reconstruction from point clouds. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Oesau, S., Verdie, Y., Jamin, C., Alliez, P., Lafarge, F., Giraudot, S., Hoang, T., and Anisimov, D. (2023). Shape detection. In *CGAL User and Reference Manual*. CGAL Editorial Board, 5.5.2 edition.

Oniga, V.-E., Breaban, A.-I., Pfeifer, N., and Diac, M. (2022). 3d modeling of urban area based on oblique uas images – an end-to-end pipeline. *Remote Sensing*, 14(2).

Overby, J., Bodum, L., Kjems, E., and Iisoe, P. (2004). Automatic 3d building reconstruction from airborne laser scanning and cadastral data using hough transform. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 34(01).

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

Pantoja-Rosero, B., Achanta, R., Kozinski, M., Fua, P., Perez-Cruz, F., and Beyer, K. (2022). Generating lod3 building models from structure-from-motion and semantic segmentation. *Automation in Construction*, 141:104430.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., and Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. In *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Peters, R., Dukai, B., Vitalis, S., van Liempt, J., and Stoter, J. (2022a). Automated 3d reconstruction of lod2 and lod1 models for all 10 million buildings of the netherlands. *Photogrammetric Engineering and Remote Sensing*, 88(3):165–170.

Peters, R., Dukai, B., Vitalis, S., van Liempt, J., and Stoter, J. (2022b). Automated 3d reconstruction of lod2 and lod1 models for all 10 million buildings of the netherlands. *Photogrammetric Engineering & Remote Sensing*, 88(3):165–170.

PIROTTI, F., Zanchetta, C., Previtali, M., Della Torre, S., et al. (2019). Detection of building roofs and facades from aerial laser scanning data using deep learning. *ISPRS ANNALS OF THE PHOTOGRAMMETRY, REMOTE SENSING AND SPATIAL INFORMATION SCIENCES*, 42(2):975–980.

Pu, S., Vosselman, G., et al. (2006). Automatic extraction of building features from terrestrial laser scanning. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 36(5):25–27.

Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement.

Remondino, F. and Gerke, M. (2015). *Oblique aerial imagery: a review*, volume 15.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R., editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc.

Singh, S. P., Jain, K., and Mandla, V. R. (2013). Virtual 3d city modeling: Techniques and applications. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XL-2/W2:73–91.

Su, K., Li, J., and Fu, H. (2011). Smart city and the applications. In *2011 International Conference on Electronics, Communications and Control (ICECC)*, pages 1028–1031.

Torii, A., Havlena, M., and Pajdla, T. (2009). From google street view to 3d city models. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 2188–2195.

Wang, L. (2022). Detailed facade reconstruction for manhattan-world buildings.

Wang, Y., Chen, Q., Zhu, Q., Liu, L., Li, C., and Zheng, D. (2019). A survey of mobile laser scanning applications and key techniques over urban areas. *Remote Sensing*, 11(13).

Wen, X., Xie, H., Liu, H., and Yan, L. (2019). Accurate reconstruction of the lod3 building model by integrating multi-source point clouds and oblique remote sensing imagery. *ISPRS International Journal of Geo-Information*, 8(3).

Willenborg, B., Sindram, M., and Kolbe, T. H. (2018). *Applications of 3D City Models for a Better Understanding of the Built Environment*, pages 167–191. Springer International Publishing, Cham.

Wu, B., Xie, L., Hu, H., Zhu, Q., and Yau, E. (2018). Integration of aerial oblique imagery and terrestrial imagery for optimized 3d modeling in urban areas. *ISPRS Journal of Photogrammetry and Remote Sensing*, 139:119–132.

Wu, Y., Kirillov, A., Massa, F., Lo, W.-Y., and Girshick, R. (2019). Detectron2. https://github.com/facebookresearch/detectron2.

Yang, X., Qin, X., Wang, J., Wang, J., Ye, X., and Qin, Q. (2015a). Building façade recognition using oblique aerial images. *Remote Sensing*, 7(8):10562–10588.

Yang, X., Qin, X., Wang, J., Wang, J., Ye, X., and Qin, Q. (2015b). Building façade recognition using oblique aerial images. *Remote Sensing*, 7(8):10562–10588.

Ying, X. (2019). An overview of overfitting and its solutions. *Journal of Physics: Conference Series*, 1168(2):022022.

Zhang, X., Lippoldt, F., Chen, K., Johan, H., Erdt, M., Zhang, X., Lippoldt, F., Chen, K., Johan, H., and Erdt, M. (2019). A data-driven approach for adding facade details to textured lod2 citygml models. pages 294–301.

Colophon

This document was typeset using LaTeX, using the KOMA-Script class `scrbook`. The main font is Palatino.