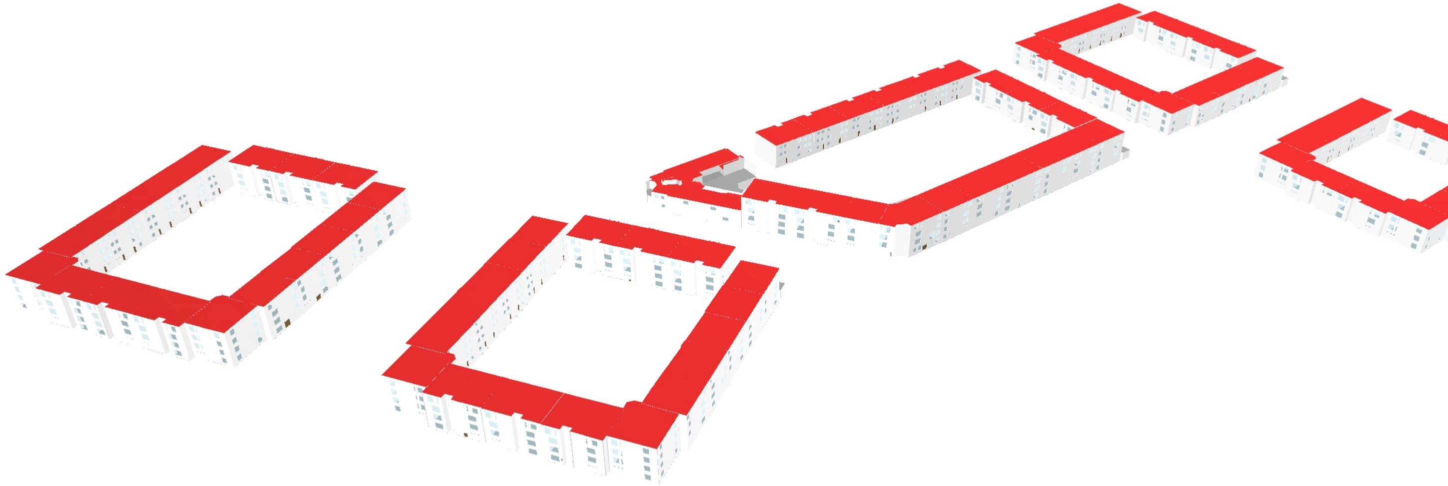
A data-driven approach to add openings to 3D BAG LoD2 building model

Yitong Xia

5445825

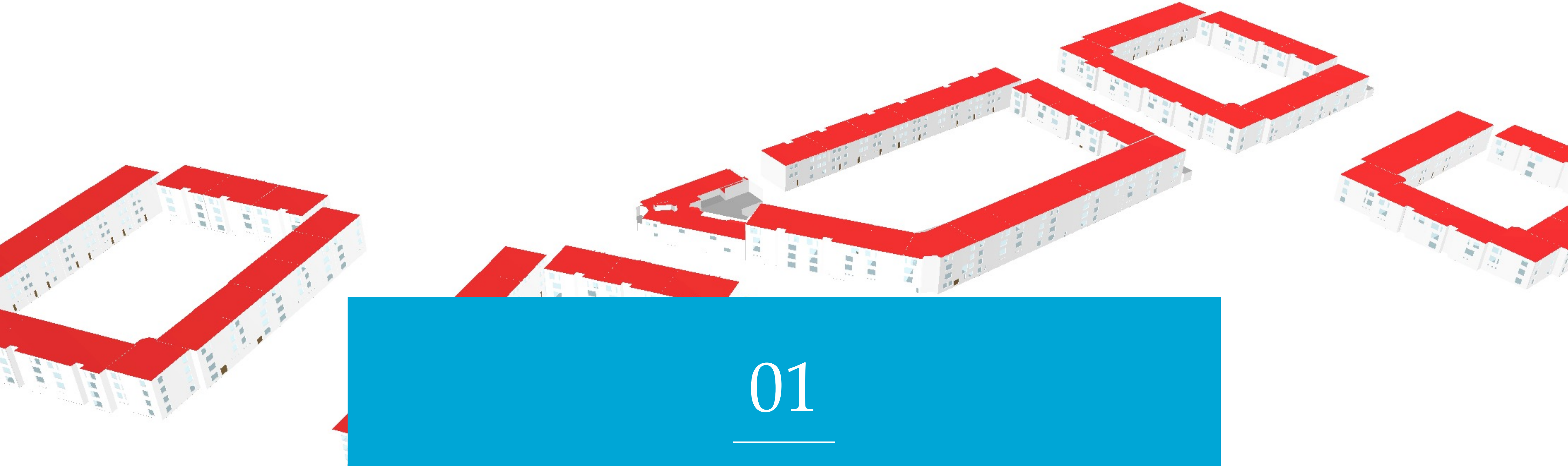
MSc Geomatics for the Built Environment

TUDelft



Content

- Research background & Objective
- Related Work
- Methodology
- Results & Analysis
- Conclusions



01

Research background & Objective

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. [2022]; Singh et al. [2013]].

Advantages of 3D city model:

- simulate realistic environments better.
- improve the accuracy of the results and their interpretation.

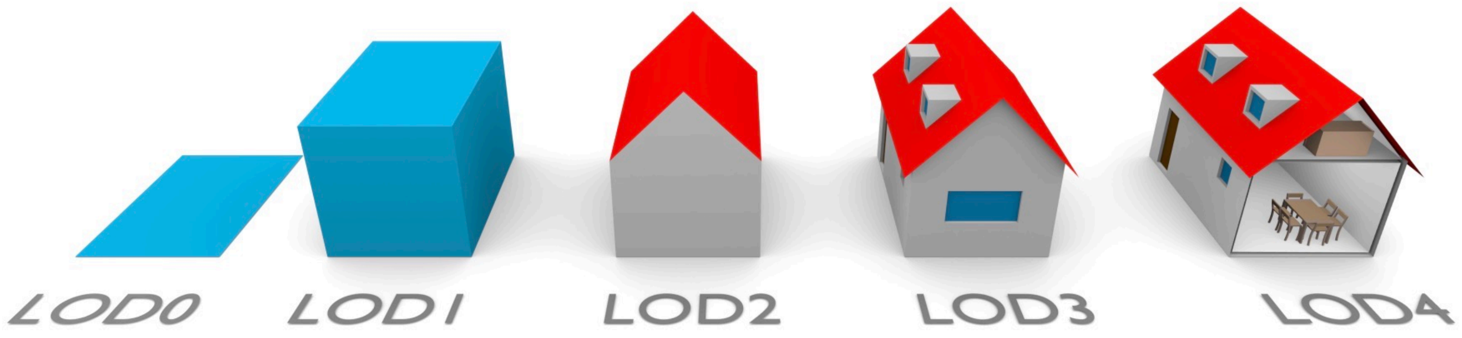
TUDelft

Fig 1. Examples of 3D city models in different LOD, [Biljecki, [2017]]

Level of Detail

One of the important characteristics of the 3D city model is the level of detail (LOD).

- an indication of how thoroughly a 3D city model has been modeled,
- describe the geometric detail of a model, primarily of buildings.



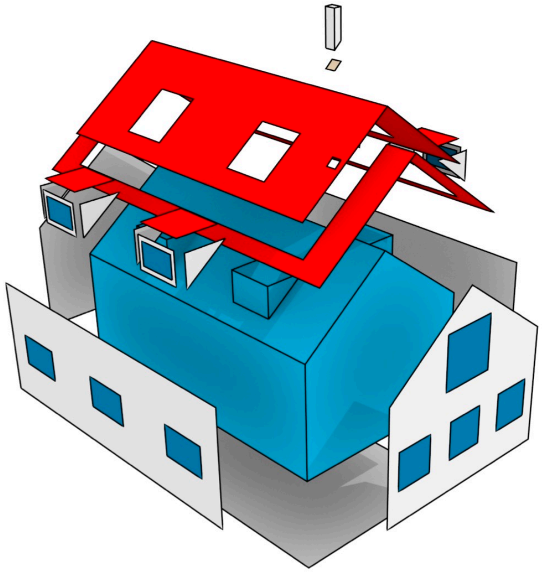


Fig 2. Definition of Level of Detail, [Biljecki, [2017]]

TU Delft

LOD2 versus LOD3

The comparison between LOD2 and LOD3:

the LOD3 model is much more detailed, especially the openings included in LOD3 are beneficial for many applications, for instance:

- Illumination analysis
- Heat loss estimating
- Rescue route planning

The generation of LOD3 3D city model is always a worthy topic to be discussed.

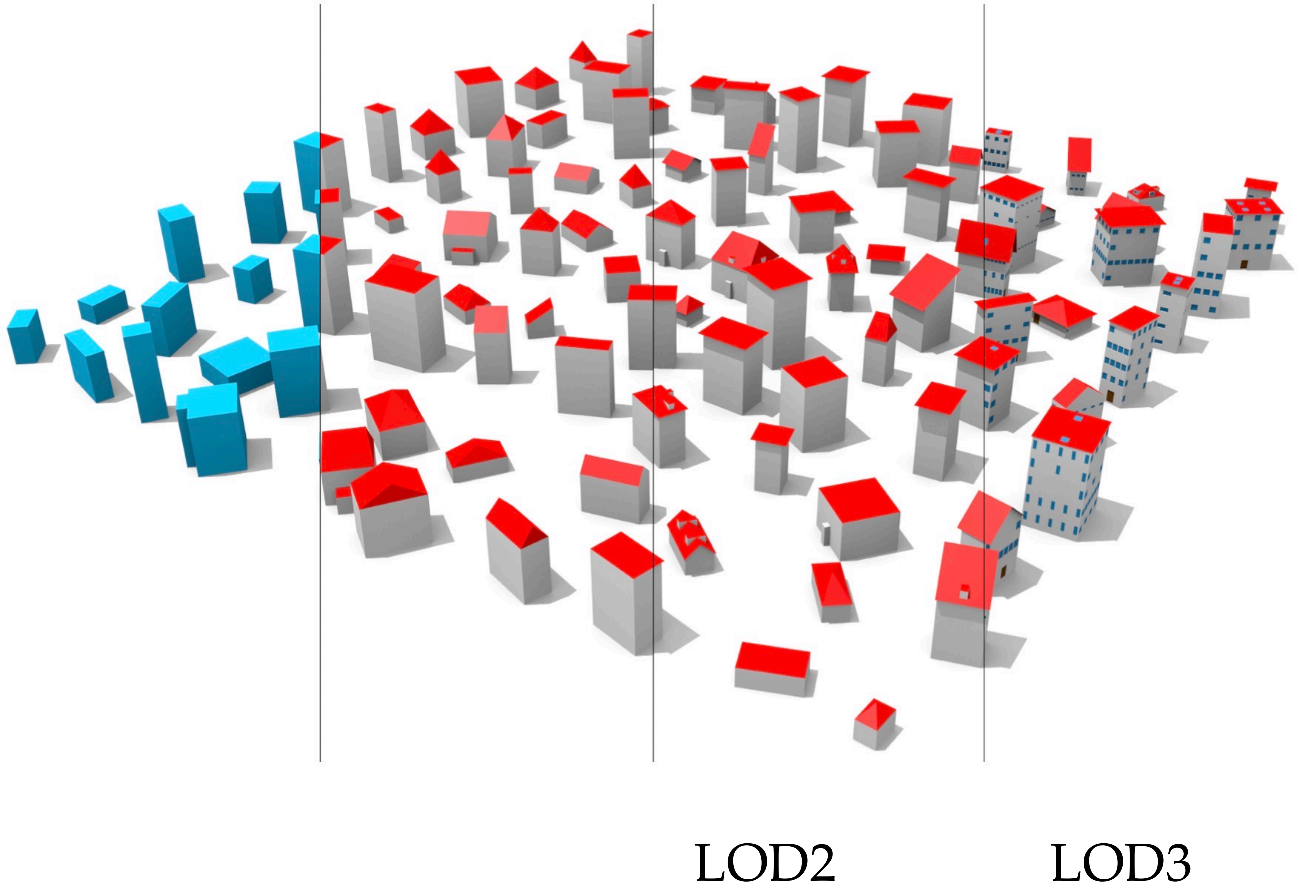


Fig 3. Comparison between LOD2 and LOD3, [Biljecki, [2017]]

TU Delft

Introduction of 3D BAG

The 3D BAG is an up-to-date 3D building model data set covering the entire Netherlands, which contains 3D models at multiple levels of detail (LOD0, LOD1.2, LOD1.3 and LOD2.2, seeing Fig 4.).

The data sources of 3D BAG:

- the building data from the Register of Buildings and Addresses (BAG);
- the airborne laser scanning (LiDAR) AHN3.

NO LOD3 models at present.

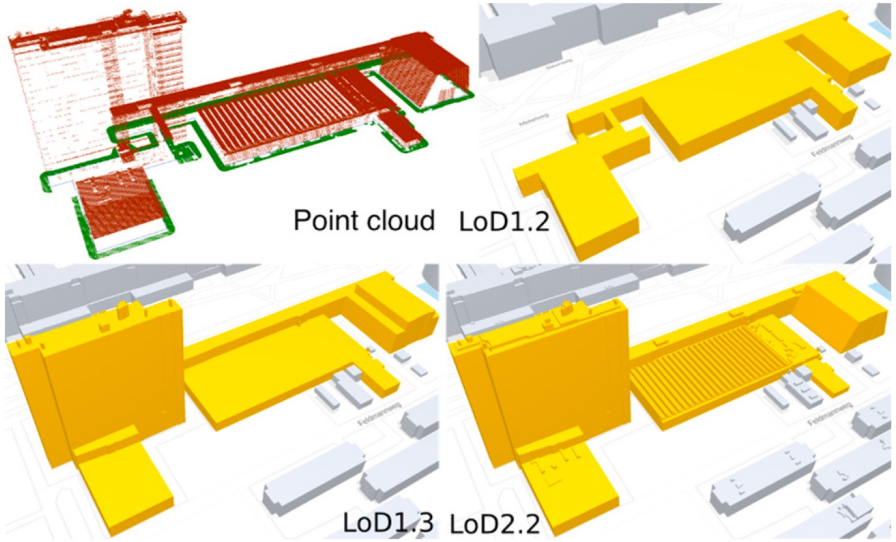
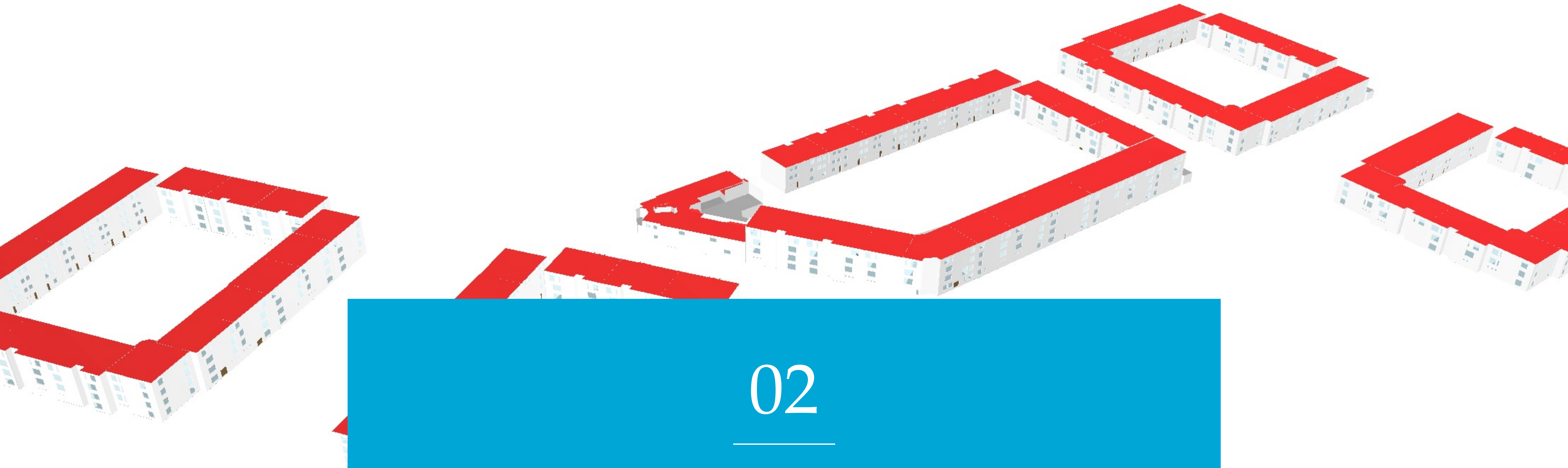


Fig 4. Current LOD of 3D BAG, [Peters, [2022]]

Research objective

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

- 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?



02

Related work

LOD2 building model reconstruction

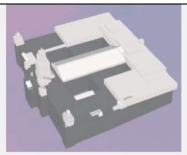
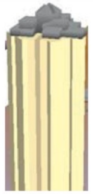
Building ID in the topographic map	Reconstructed model (colors are for distinguishing only)	Photo (collected from the Internet or taken by the authors)
11**374		
11**486		
11**535		
11**845		

Fig 5. 3D reconstruction with ALS point clouds and topographic maps utilizing mSTEP method, [Chen et al. [2018]]

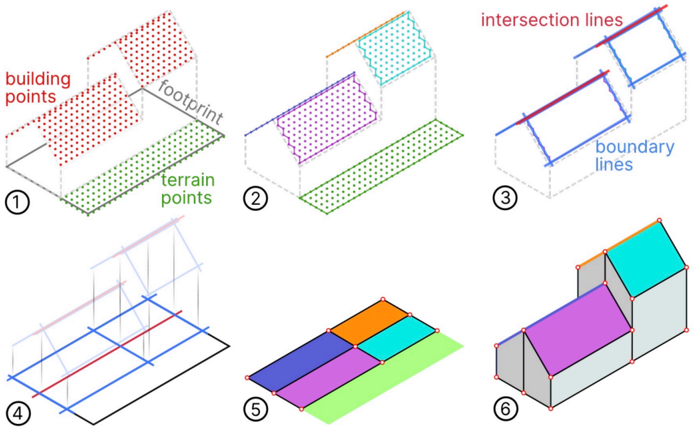


Fig 6. 3D reconstruction by combining cadastral data and ALS point clouds, [Peters et al. [2022]]

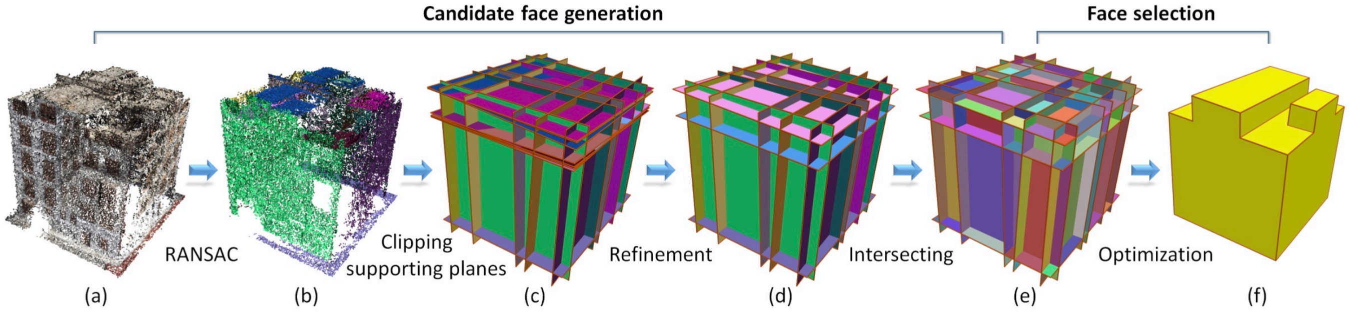


Fig 7. Polyfit: Polygonal surface reconstruction using various point clouds, [Nan and Wonka, [2017]]

TU Delft

LOD3 building model reconstruction

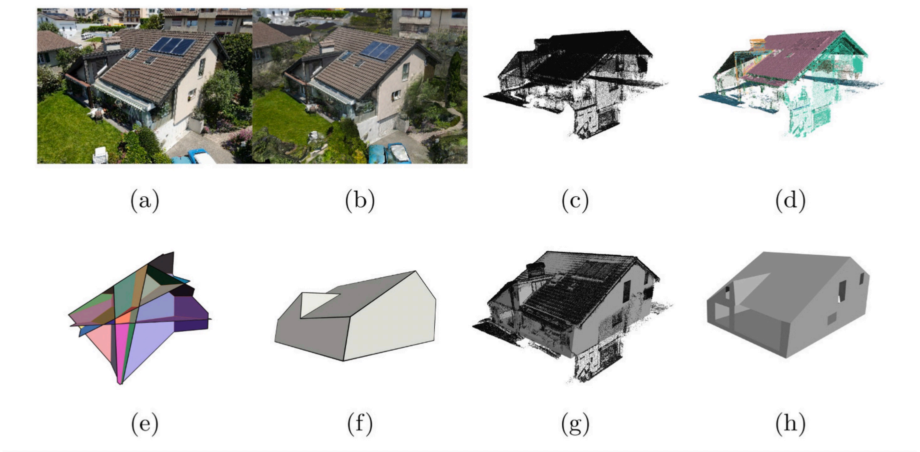


Fig 8. LOD3 model generation using SfM and semantic segmentation using images, [Pantoja-Rosero B et al. [2022]]

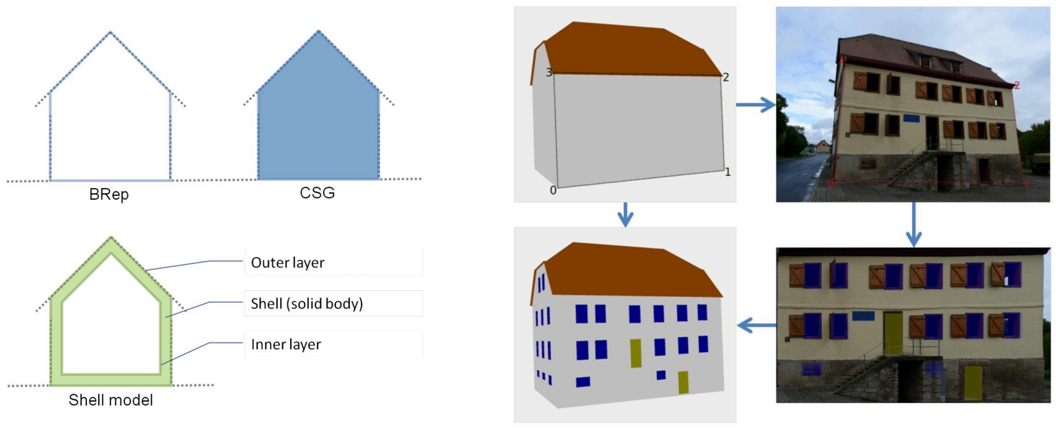


Fig 9. LOD3 model generation using "Shell" model using UAV imagery, [Huang et al. [2020]]

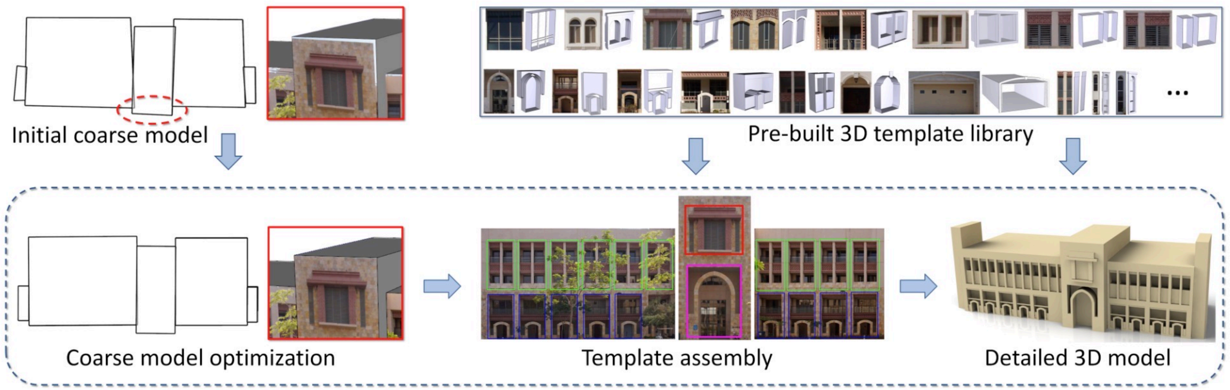


Fig 10. LOD3 model generation by assembling 3D template on coarse models, [Nan et al. [2015]]

Façade element detection and extraction

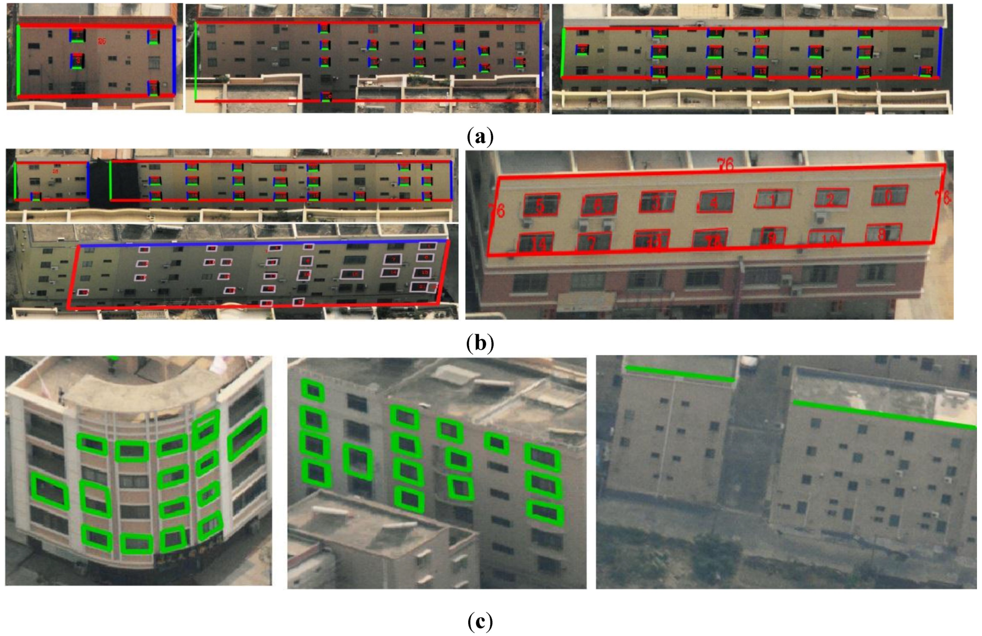


Fig 11. Façade identification employing edge detection, region growing, and Hough transforms, Yang et al. [2015]].

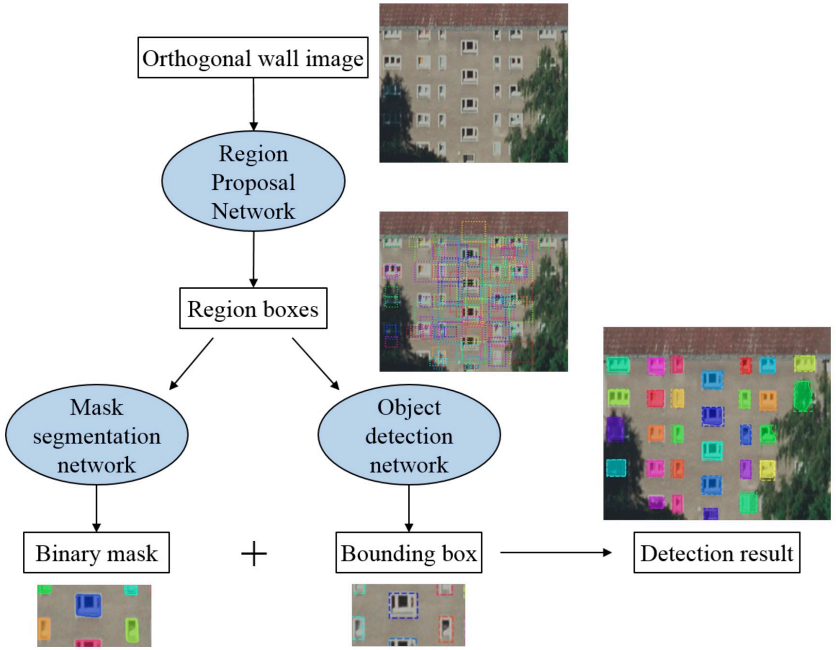
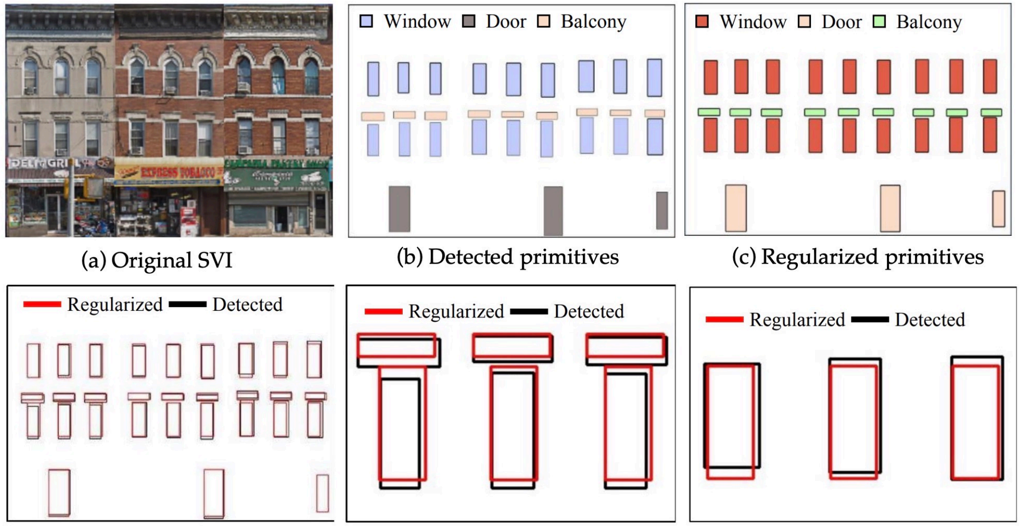


Fig 12. Façade elements extraction using deep learning techniques including Faster R-CNN and Mask R-CNN, [Wang [2021], Zhang et al.[2020]].

TU Delft

Façade element layout regularization



(a) Original SVI (b) Detected primitives (c) Regularized primitives
(d) Regularized primitives (e) and (f) Enlarged demonstrations

Fig 13. Using binary linear programming (BIP) to regularize the façade element, [Hu et al. [2020]]

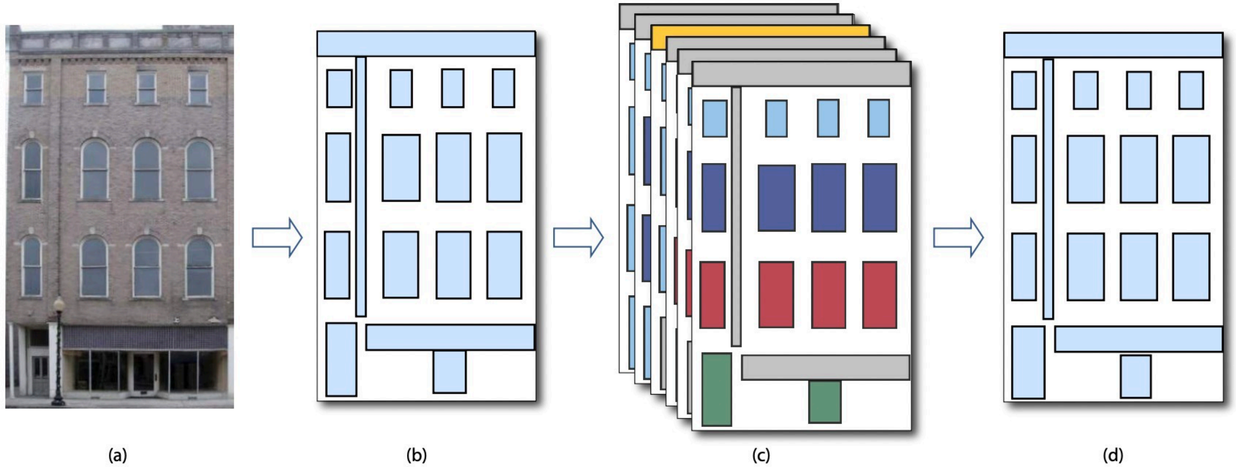
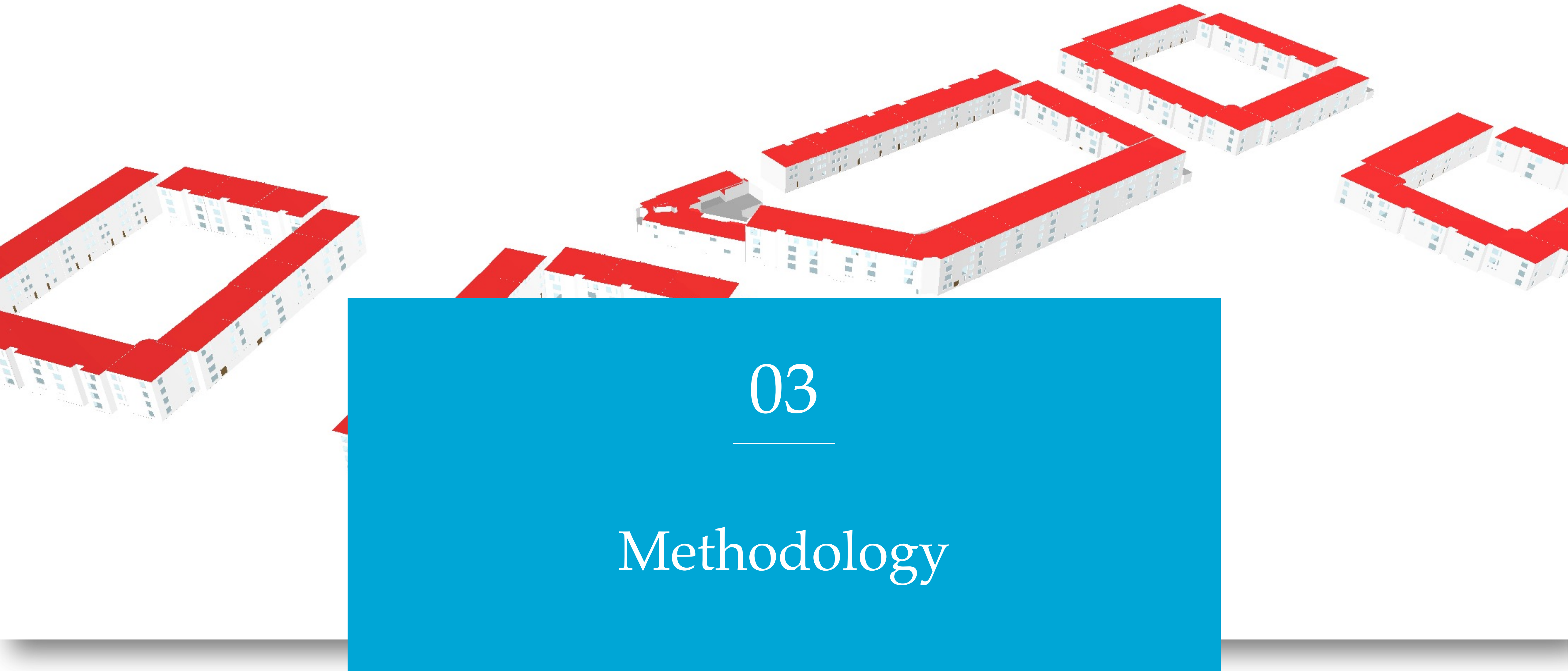


Fig 14.Using constraint detection algorithm to implement the layout regularization, [Jiang et al. [2016]]



03

Methodology

Methodology: Overview

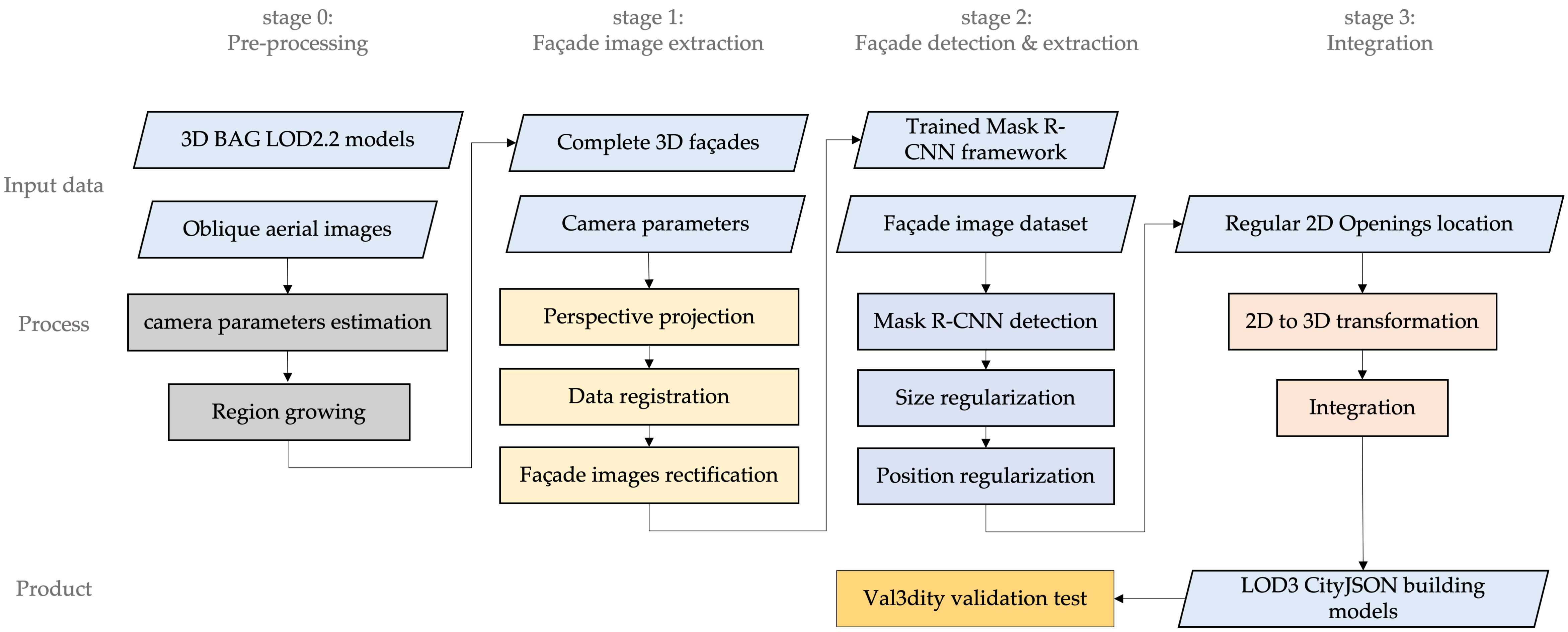
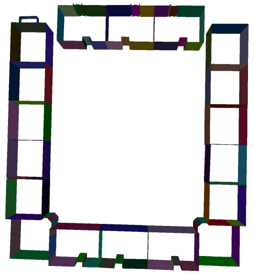


Fig 15. General workflow of the pipeline

Data pre-processing

- Camera parameters estimation for perspective projection in the 1st stage: Pix4D
- For better extraction of complete façade texture imagery: Region growing algorithm



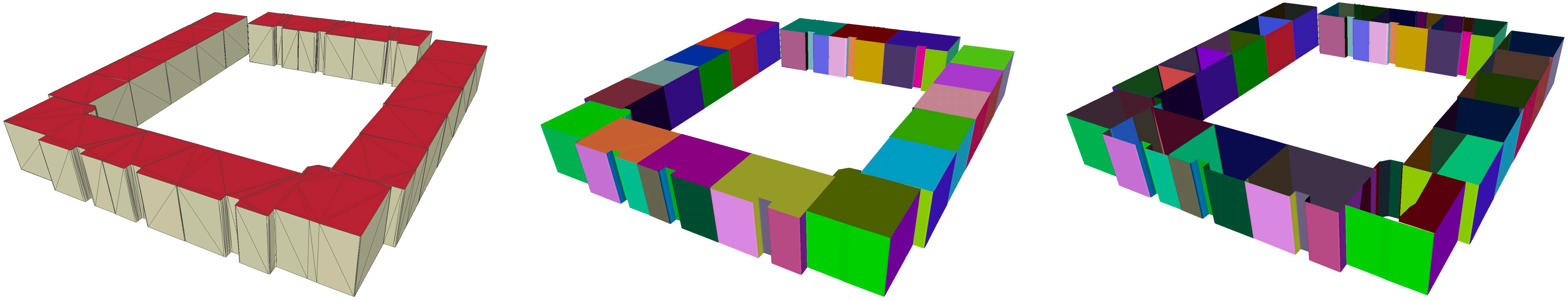


Fig 16(a). Original 3D BAG building

Fig 16 (b). Co-planar surface merging result

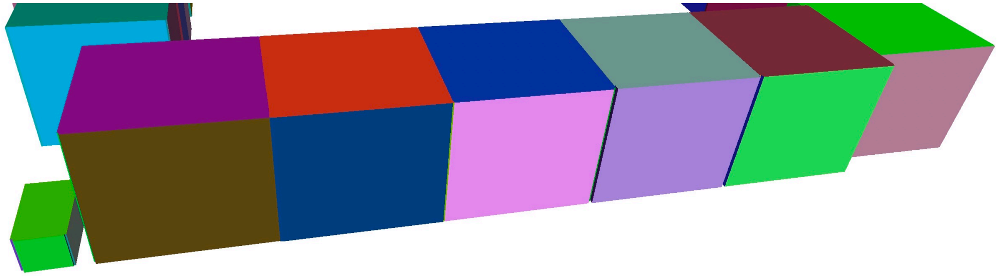
Fig 16 (c). Wallsurface only

Input 3D building model of the following pipeline

Stage 1: Façade texture images extraction

Purpose: to find the corresponding texture image of individual 3D façade from 2D oblique aerial images.

+



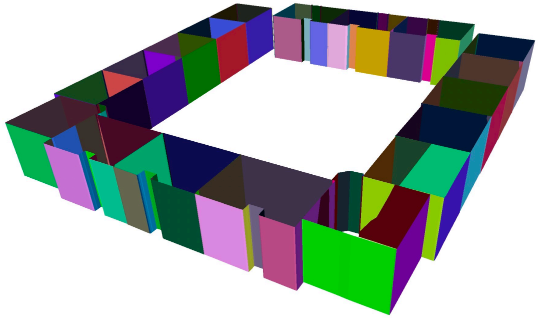
automatically

extract

Visible façade texture image are extracted as many as as possible.

TU Delft

Stage 1 overview: Façade texture images extraction



Input data:

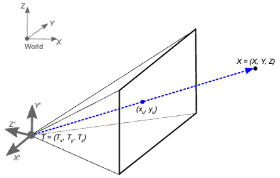
Complete 3D façades

Camera parameters

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & f_y \\ 0 & 0 & 1 \end{bmatrix} \qquad T = (T_x, T_y, T_z)$$

$$R = R_x(\omega)R_y(\phi)R_z(\kappa)$$

$$P = K[R| - Rt]X$$



Perspective projection

Data registration → Regression function

Projection result optimization

Façade images rectification

Extracted façade texture images

Perspective projection

To project a 3D scene (point) onto a 2D image space, utilizing camera intrinsic parameters and extrinsic parameters. (x_u, y_u) can be obtained using the following equations:

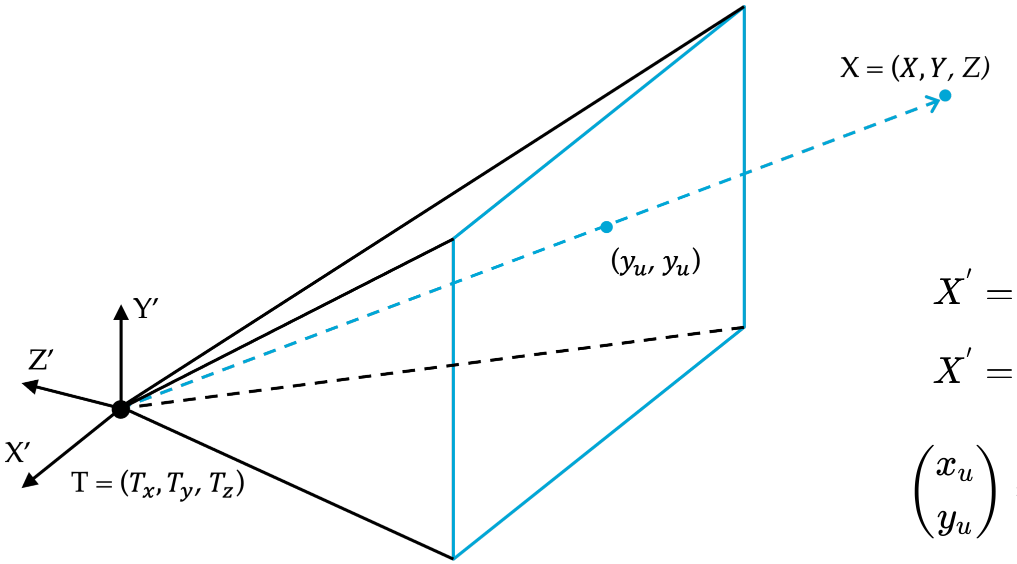


Fig 17. Perspective projection

$$X' = (X', Y', Z')$$

$$X' = R^T(X - T)$$

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

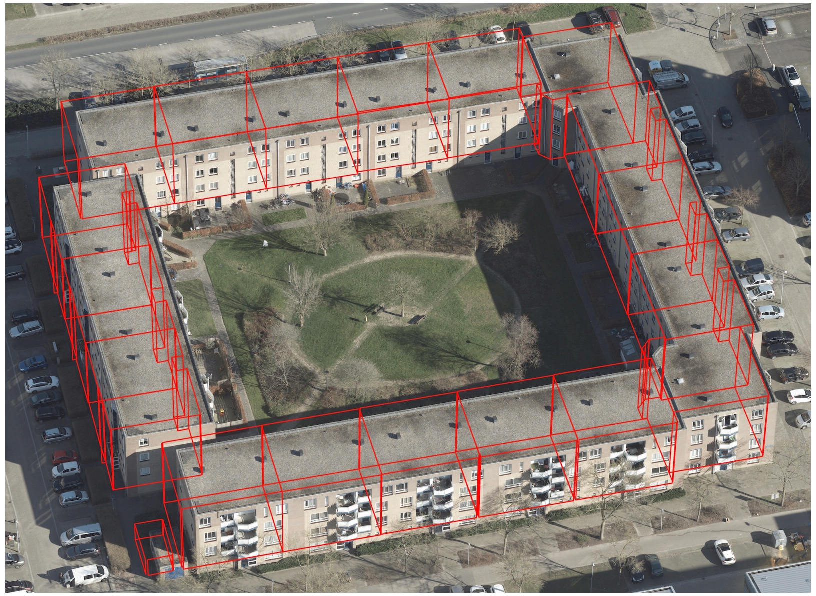
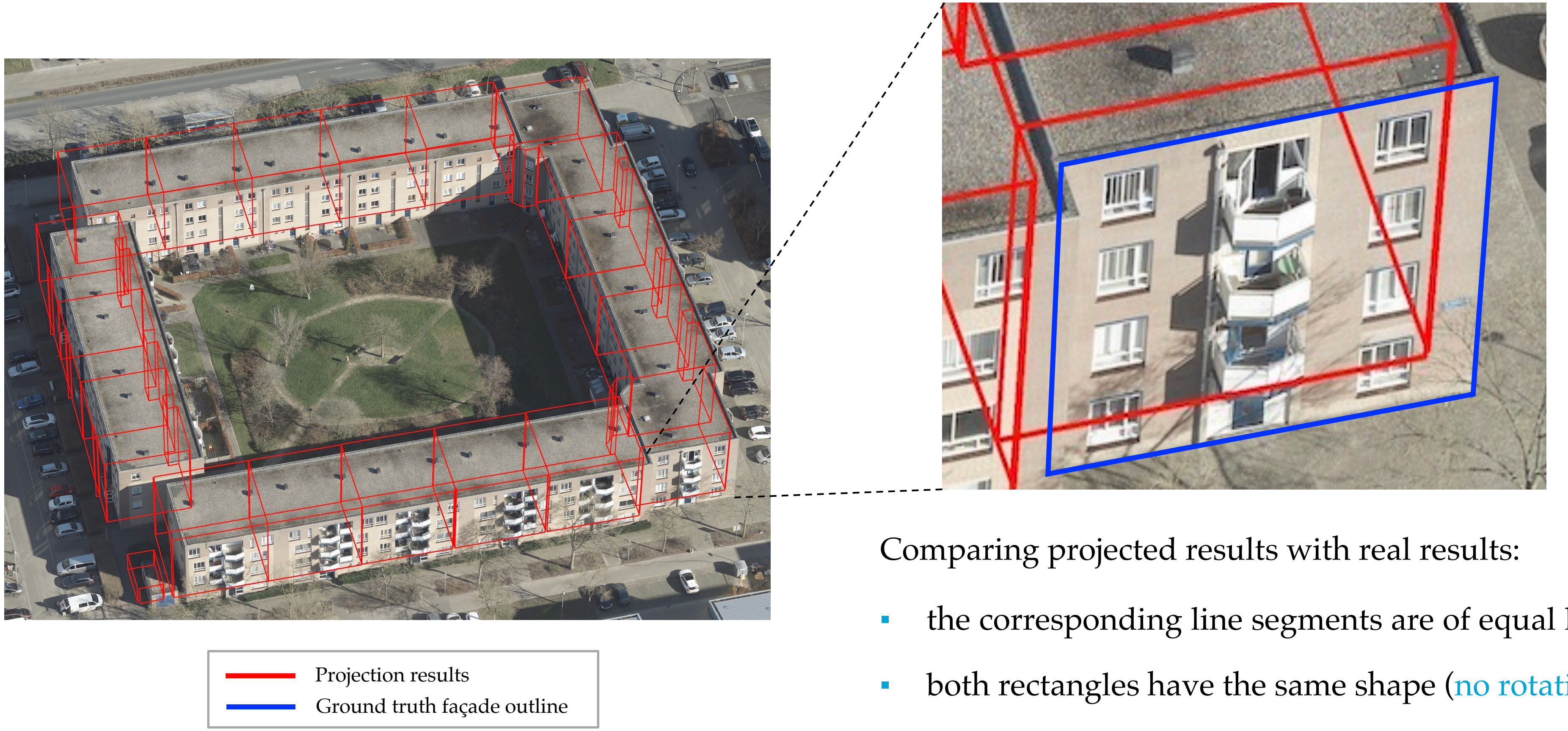


Fig 18. Perspective projection result

Data registration: model determination

There’s offset between projection result and ground truth result:



Comparing projected results with real results:

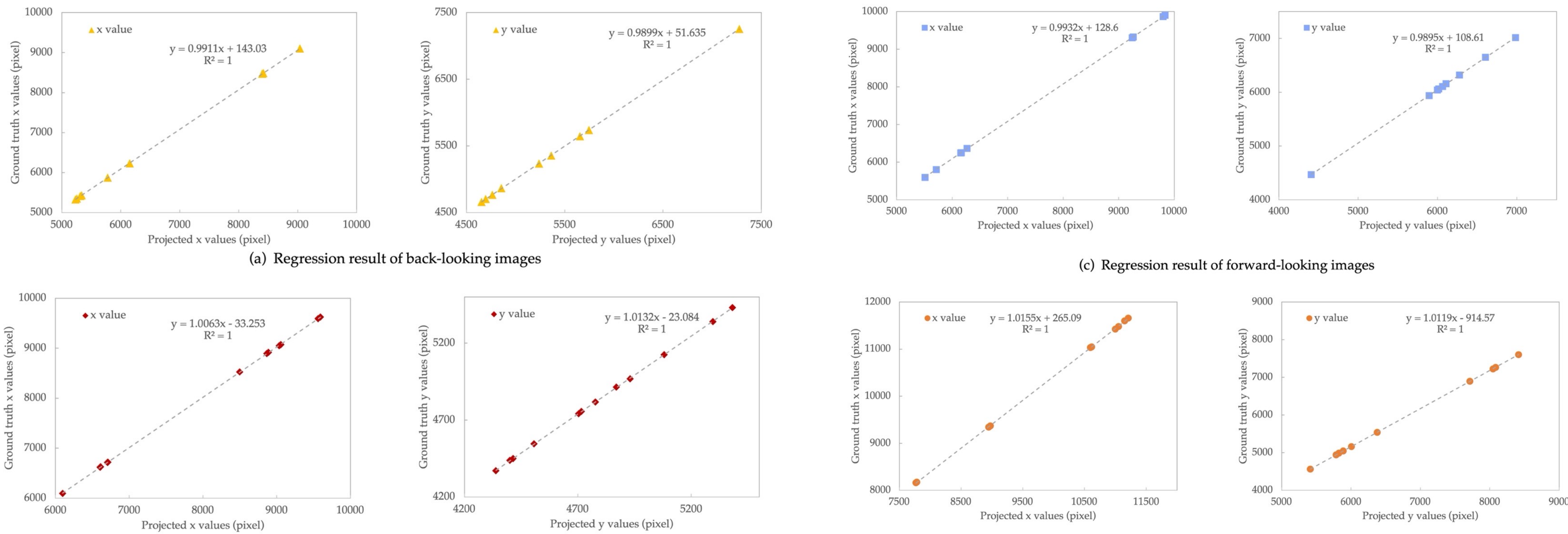
- the corresponding line segments are of equal length (no scaling);
- both rectangles have the same shape (no rotation);
- only the offset in position (need translation).

TU Delft

Fig 19. Comparison of projection result and the ground truth

Data registration

Based on the linear relationship between projection points and ground truth points, least squares regression is utilized to obtain the translation value (offsets) in X and Y respectively.



(a) Regression result of back-looking images

(c) Regression result of forward-looking images

(b) Regression result of right-looking images

(d) Regression result of left-looking images

Fig 20. The **linear relationship** between projection results and ground truth value

Projection results optimization

The optimized results are highly consistent with the ground truth after using the regression function to "move"* the projection results:

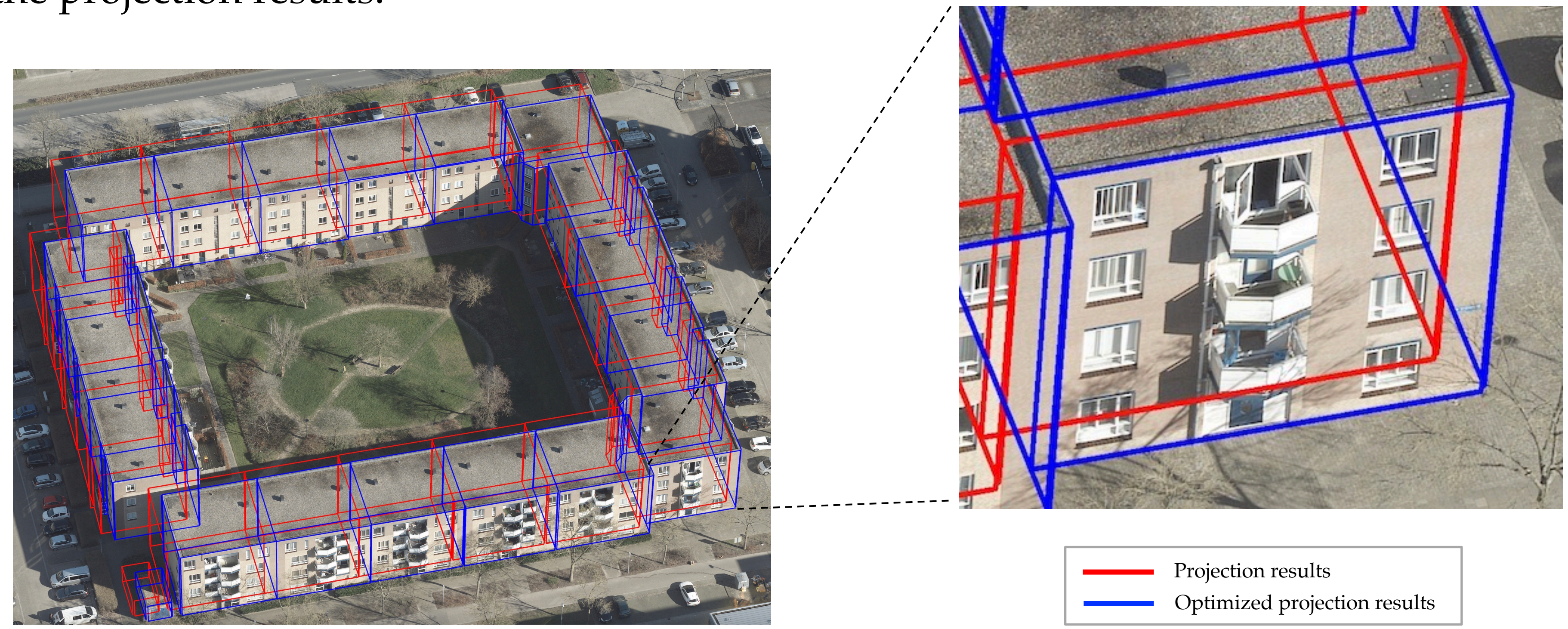


Fig 21. Comparison of projection result and the optimized result

*"move" means to translate the projection results with calculated offsets using regression function.

Texture images rectification and extraction

Perspective transformation: a mathematical mapping that allows the conversion of points and shapes in a 2D plane to another self-defined 2D plane from a different perspective.

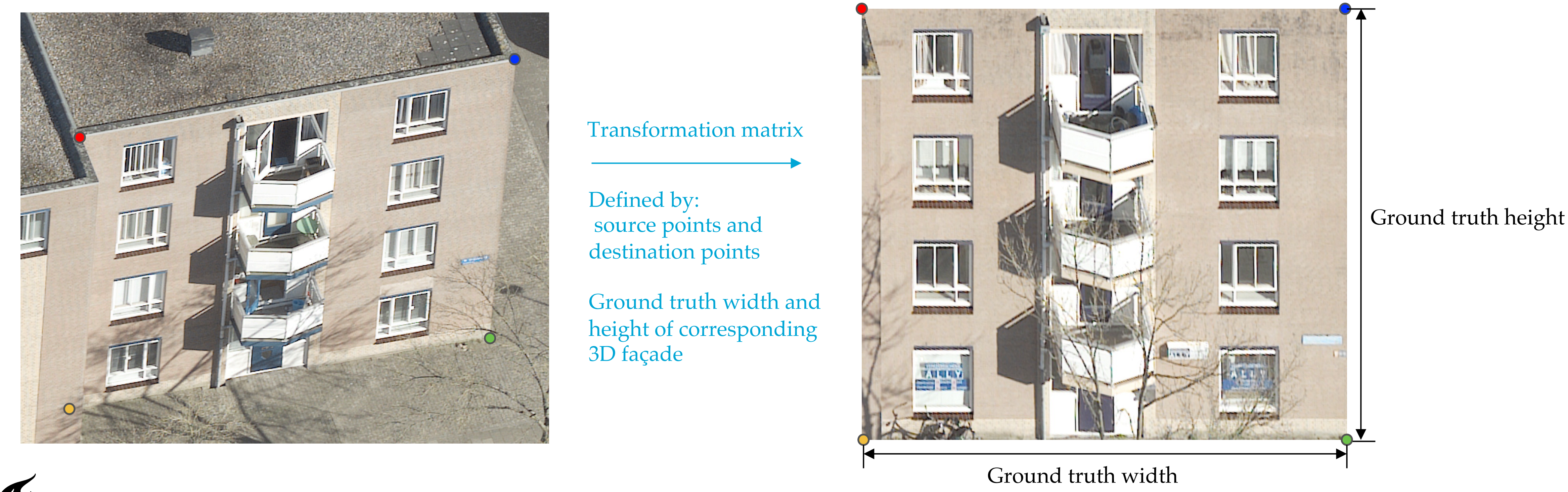


Fig 22. Perspective transformation process

Stage 2: openings detection & extraction

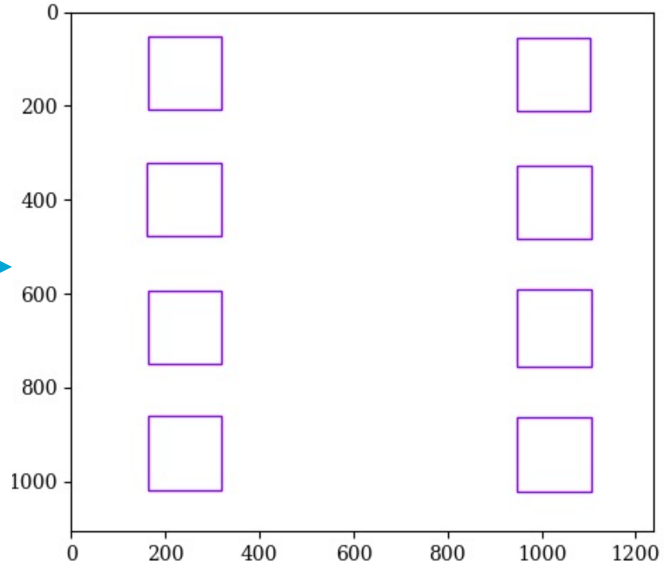
Purpose: to detect and extract openings (windows, doors) automatically, and optimize their location to make the arrangement of façade openings more aesthetically pleasing.

Extracted façade texture image

Automatic openings detection

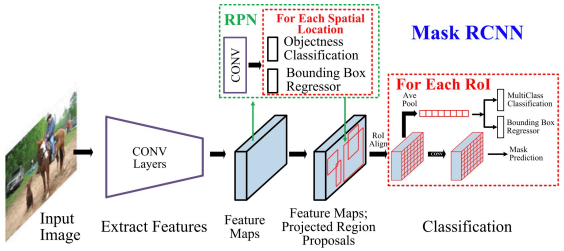
b. Openings prediction results

Layout optimization



c. Openings layout optimization

Stage 2 overview: openings detection & extraction



Trained Mask R-CNN framework

Façade texture image dataset

↓

Mask R-CNN prediction

↓

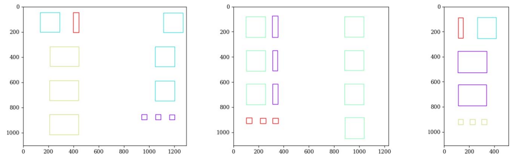
Openings extraction result
(2D pixel coordinates)

↓

Size regularization

↓

Position regularization



↓

Regularized 2D openings location

TU Delft

Openings detection & extraction using Mask R-CNN

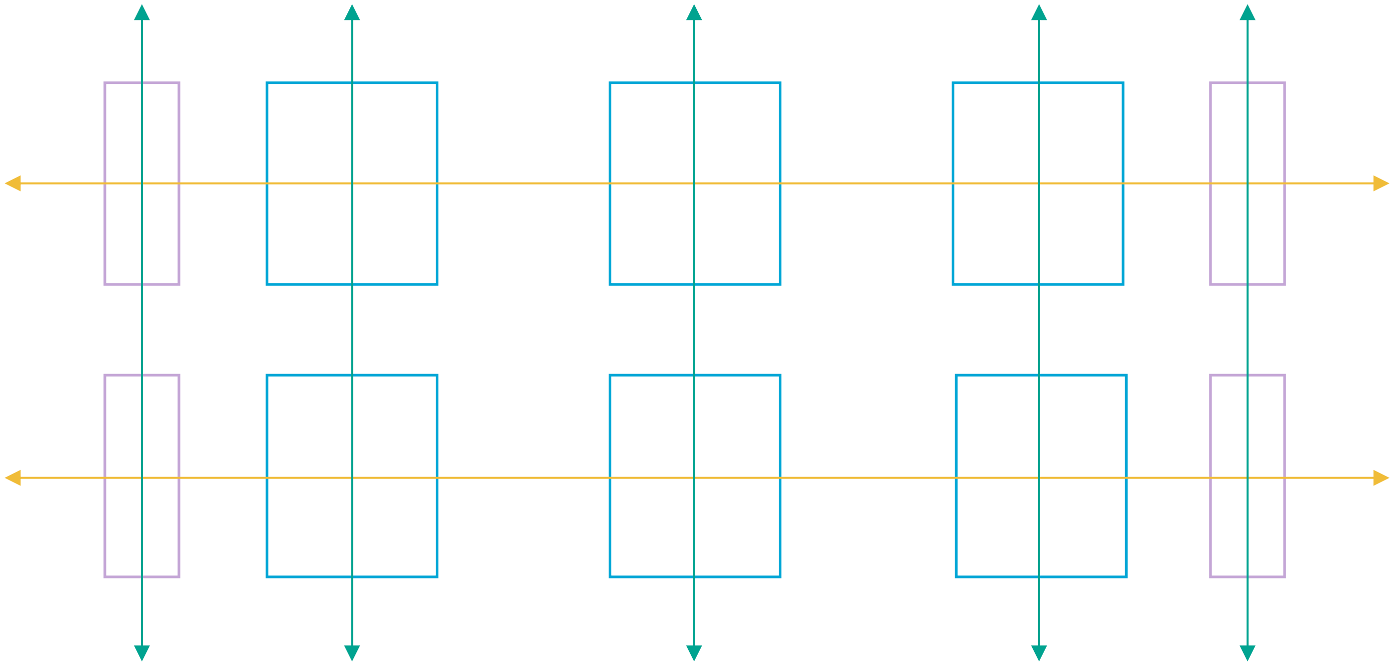
Mask R-CNN model with backbone ResNet-101 FPN is utilized to detect the openings:

The output contains the segmentation result, the bounding box, and the confidence score of the prediction result.

Fig 23. Mask R-CNN detection results

TU Delft

Openings layout optimization



- Openings with initially similar sizes should be adjusted to have identical size;
- Openings originally positioned horizontally and vertically should be adjusted to align horizontally and vertically;

TU Delft

Position regularization

Two-step adjustment is applied to adjust the X and Y coordinates of each centroid (take horizontal adjustment as example):

1. Sort the centroids in ascending order based on y value;
2. Determine the horizontal relationship by calculating the difference value between current centroid (c_{iy}) and the next centroid ($c_{(i+1)y}$): if difference exceeds the threshold, the next centroid is treated as the start of a new horizontal row;
3. Compute the average $\overline{c_y}$ of current horizontal group and replace with the new average value.

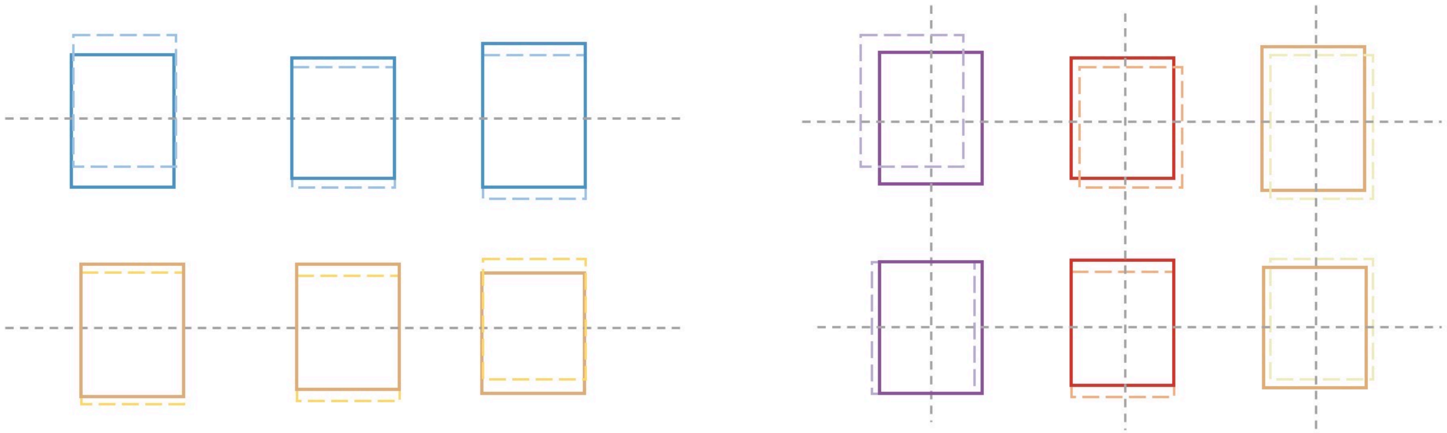


Fig 24 (a) Step 1: horizontal adjustment

Fig 24 (b) Step 2: vertical adjustment

TUDelft

Size regularization

- Unsupervised DBSCAN algorithm is firstly employed for classifying windows based on their height and width, with eps = 50 and minimal sample = 1.
- Replace the original size with a calculation of the average length and width of each group.

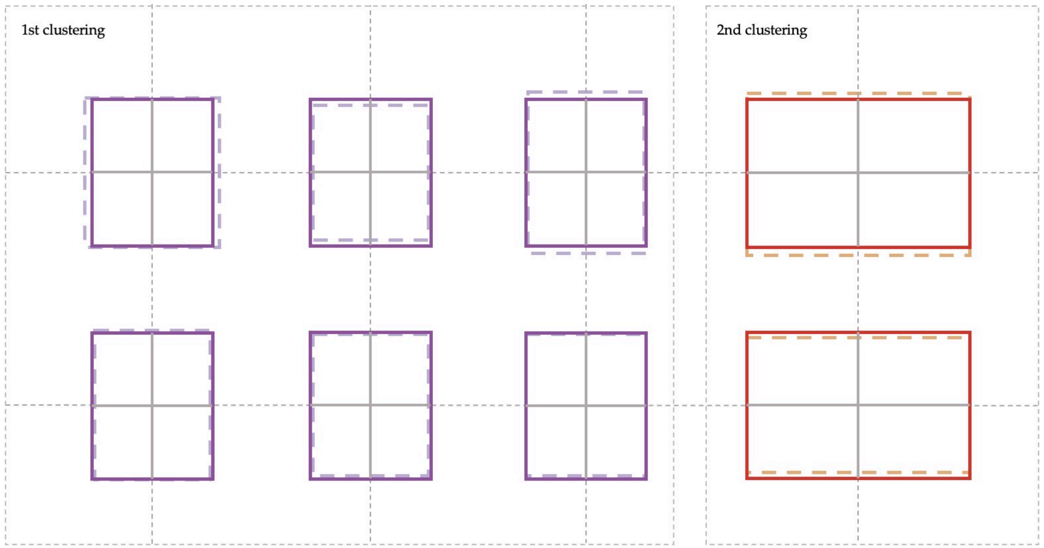
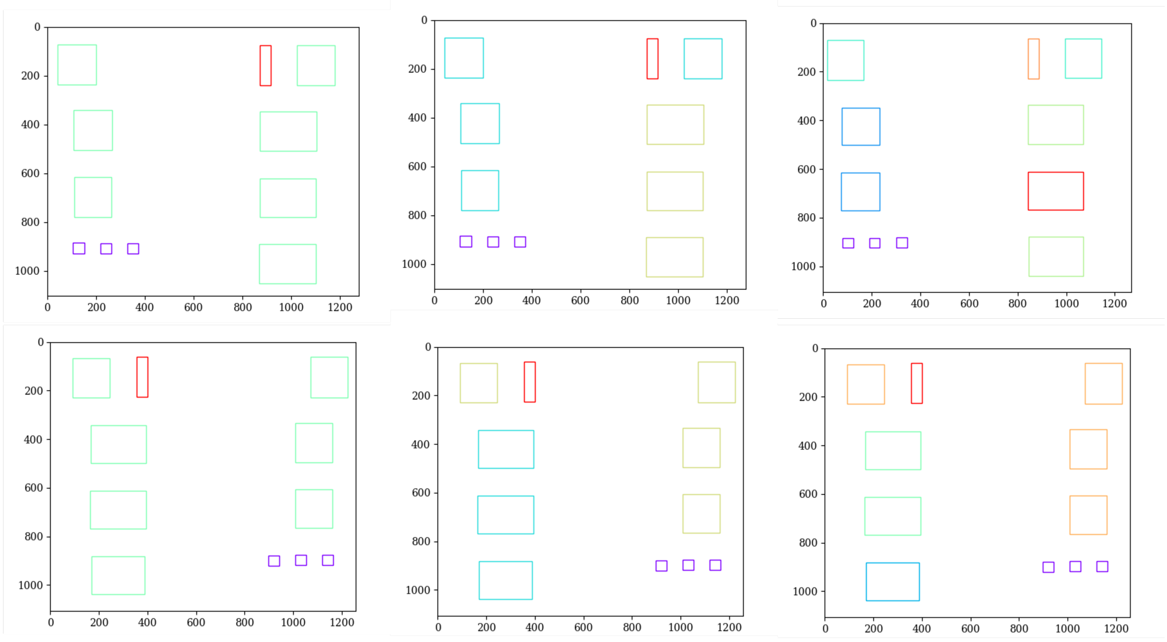


Fig 25. Size regularization



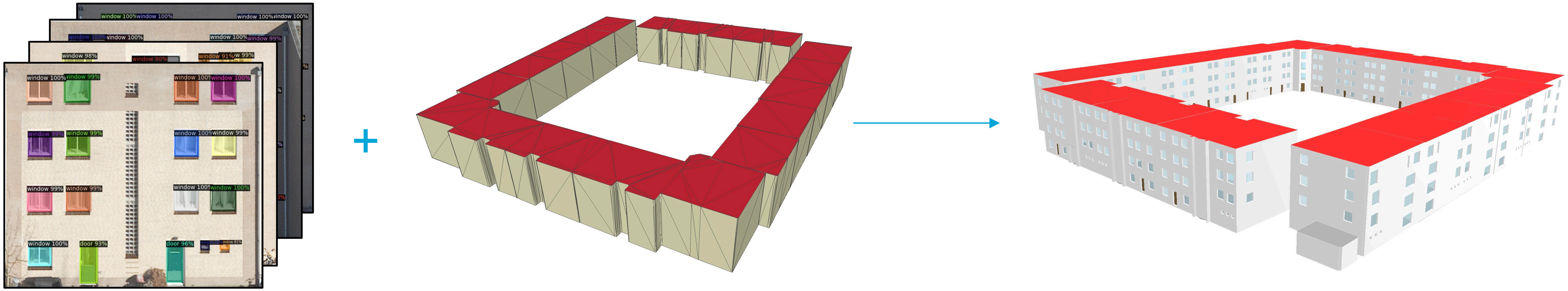
Eps = 70 *Eps* = 50 *Eps* = 10

Fig 26. Experiments with various eps

We tried several values of *eps* with the aim of obtaining the best automatic clustering results, which proved that *eps* = 50 is the optimal.

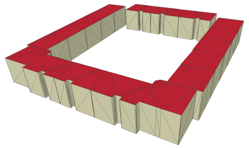
Stage 3 overview: Final integration

Purpose: to convert the 2D extracted openings to 3D ones, then integrate the 3D openings and 3D façade to obtain the final LOD3 building models.



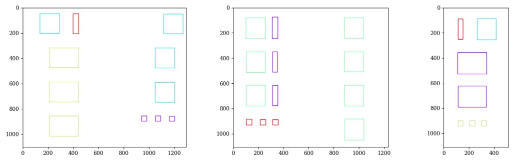
TU Delft

Stage 3 overview: Final integration

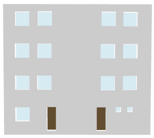


3D BAG LOD2.2 model

Regularized 2D Openings location

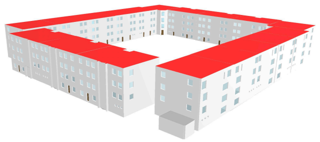


2D-to-3D transformation



Integration of 3D openings and 3D building model

LOD3 CityJSON building models

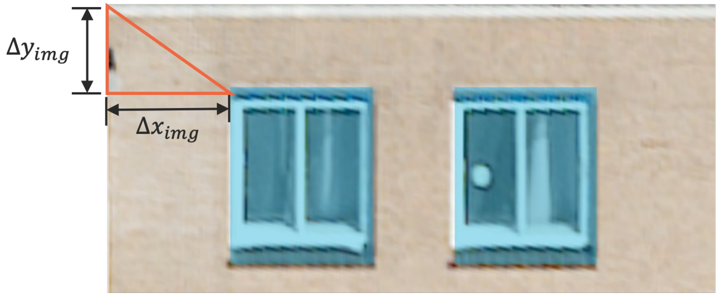


Conversion of 2D openings to 3D

step ① : Compute length of Δy_{3D} and Δx_{3D} :

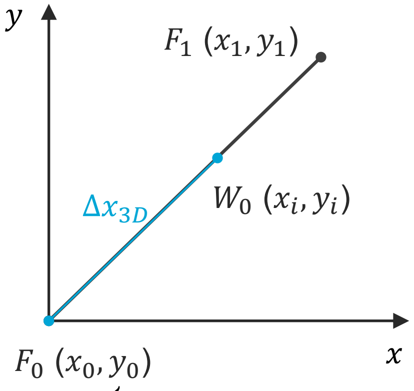
$$\frac{\Delta y_{3D}}{\Delta y_{img}} = \frac{H_{3D}}{H_{img}} \longrightarrow \Delta y_{3D} = \frac{H_{3D}}{H_{img}} \times \Delta y_{img} \longrightarrow z_i = z_0 - \Delta y_{3D}$$

$$\frac{\Delta x_{3D}}{\Delta x_{img}} = \frac{H_{3D}}{H_{img}} \longrightarrow \Delta x_{3D} = \frac{H_{3D}}{H_{img}} \times \Delta x_{img}$$



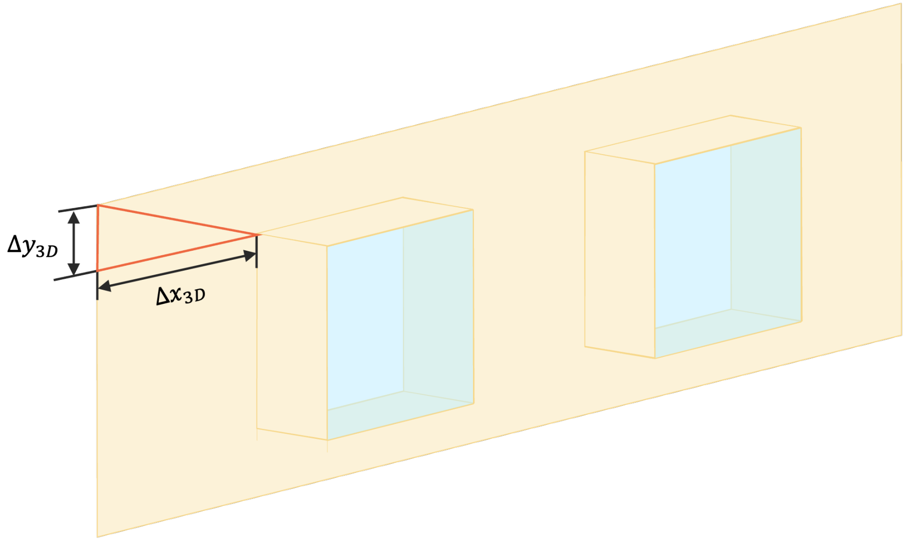
2D space

step ② : Compute 3D coordinates x_i, y_i:



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

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



3D space

TU Delft

Fig 27. Calculation of 3D coordinates by Principle of Similar Triangles

Integration of openings and 3D building model

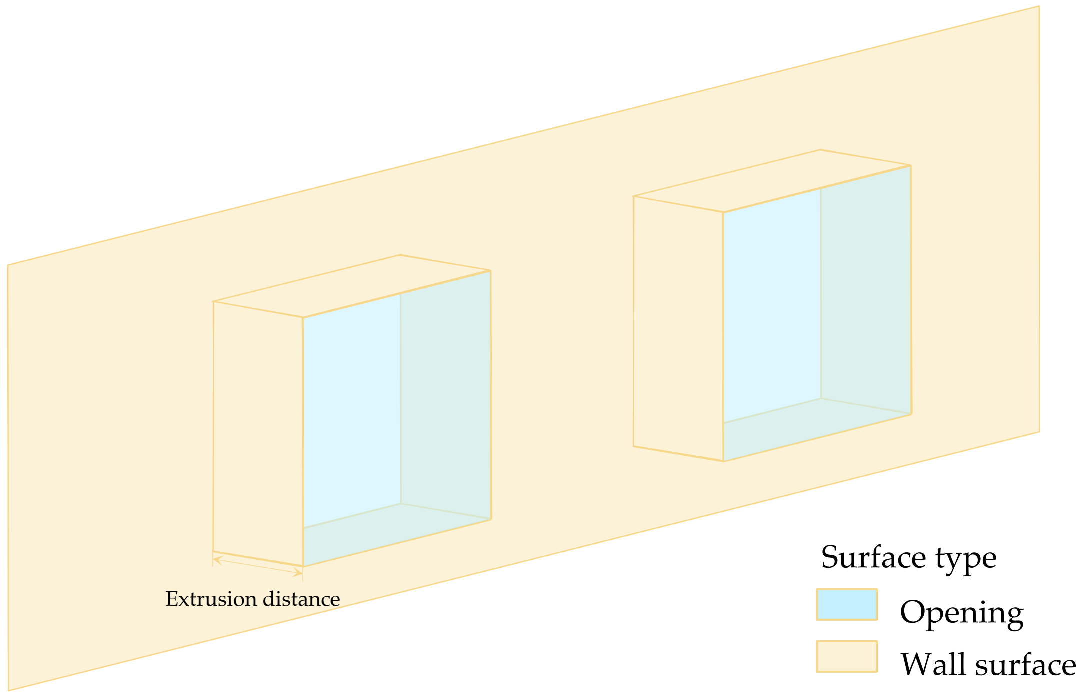


Fig 28. Detailed ideal façade structure

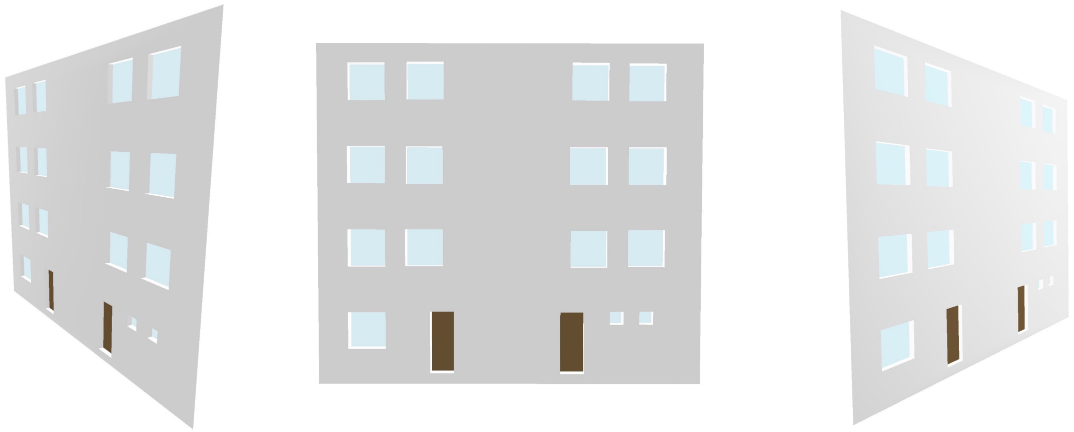


Fig 29. Implementation result (visualizing in Azul)

04

Results & Analysis

Research area and datasets

- The experimental area for this research is located near Almere Centrum.
- The imagery dataset has 30% side overlap and 60% forward overlap.

Fig 30. Research area: Almere Centrum

Research area and datasets

- The selected 3D city model of this research is 3D BAG (https://3dbag.nl/en/download).

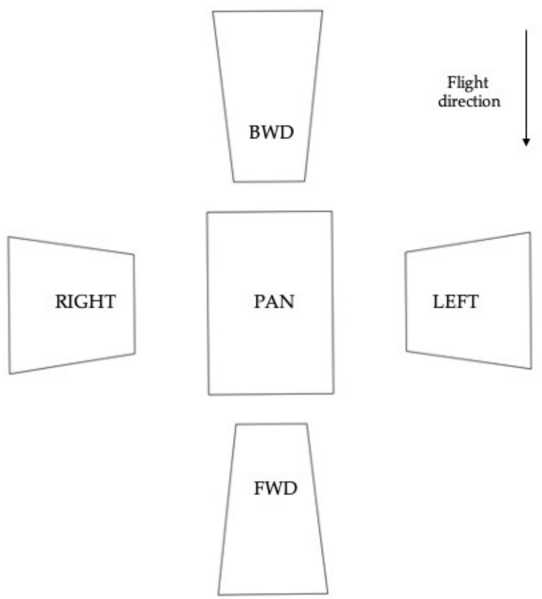


Fig 31. Five view camera system

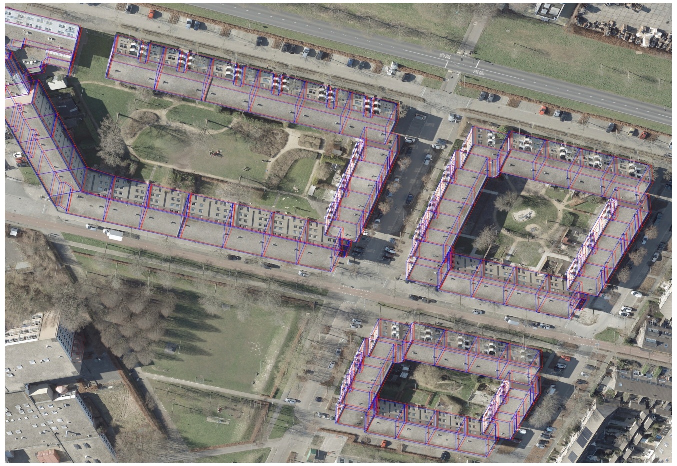
Fig 32. 3D BAG LOD 2.2 data in Almere Centrum

TUDelft

General applicability of the registration model

(a) right-looking

(b) left-looking



(c) back-looking

(d) forward-looking

Preliminary projection result
Optimized result

Fig 33. The result of using the same set of registration models on different buildings/images

TU Delft

Mask R-CNN detection

Table 1: comparison of ResNet-50 and ResNet-101 (iteration: 2,000)

Backbone	Type	AP(%)	AP windows(%)	AP Doors(%)
ResNet-50	segmentation	72.4	56.3	56.6
ResNet-50	bbox	71.5	64.2	55.4
ResNet-101	segmentation	**73.2**	**65.7**	56.2
ResNet-101	bbox	**72.3**	**65.7**	55.1

Table 2: comparison of different iteration time: 2,000 versus 5,000)

Iteration	Type	AP(%)	AP windows(%)	AP Doors(%)
5000	segmentation	**75.5**	**67.9**	**61.7**
5000	bbox	**72.8**	65.1	58.0
2000	segmentation	73.2	65.7	56.2
2000	bbox	72.3	65.7	55.1

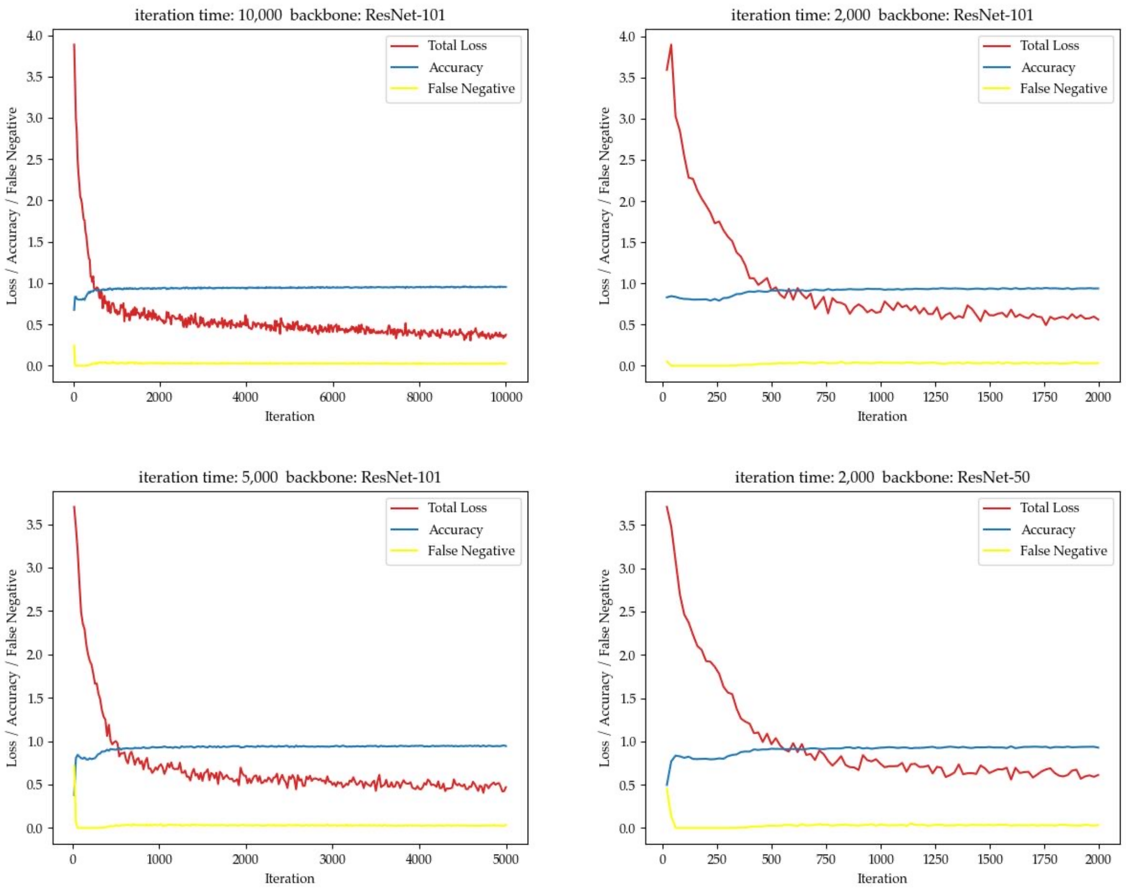


Fig 34. Comparison of total loss, accuracy and FN in different condition

Error analysis of Mask R-CNN detection

- **Misclassifications**: Doors are mistakenly predicted as windows;
- **Obstructions**: trees, large balconies;
- The physical state of the windows.

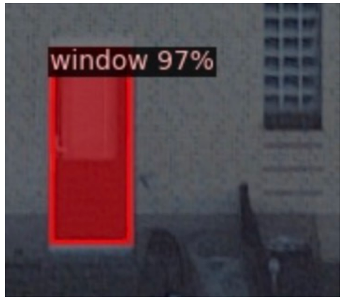


Fig 35 (a). wrong prediction result

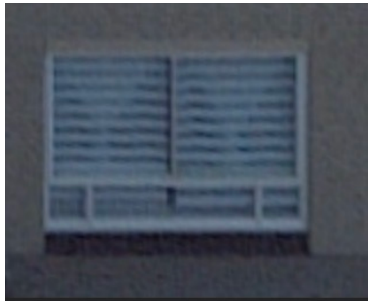


Fig 35 (b). texture of glass influence the result

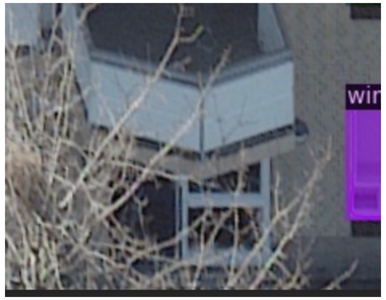


Fig 35 (c). large balcony/obstruction

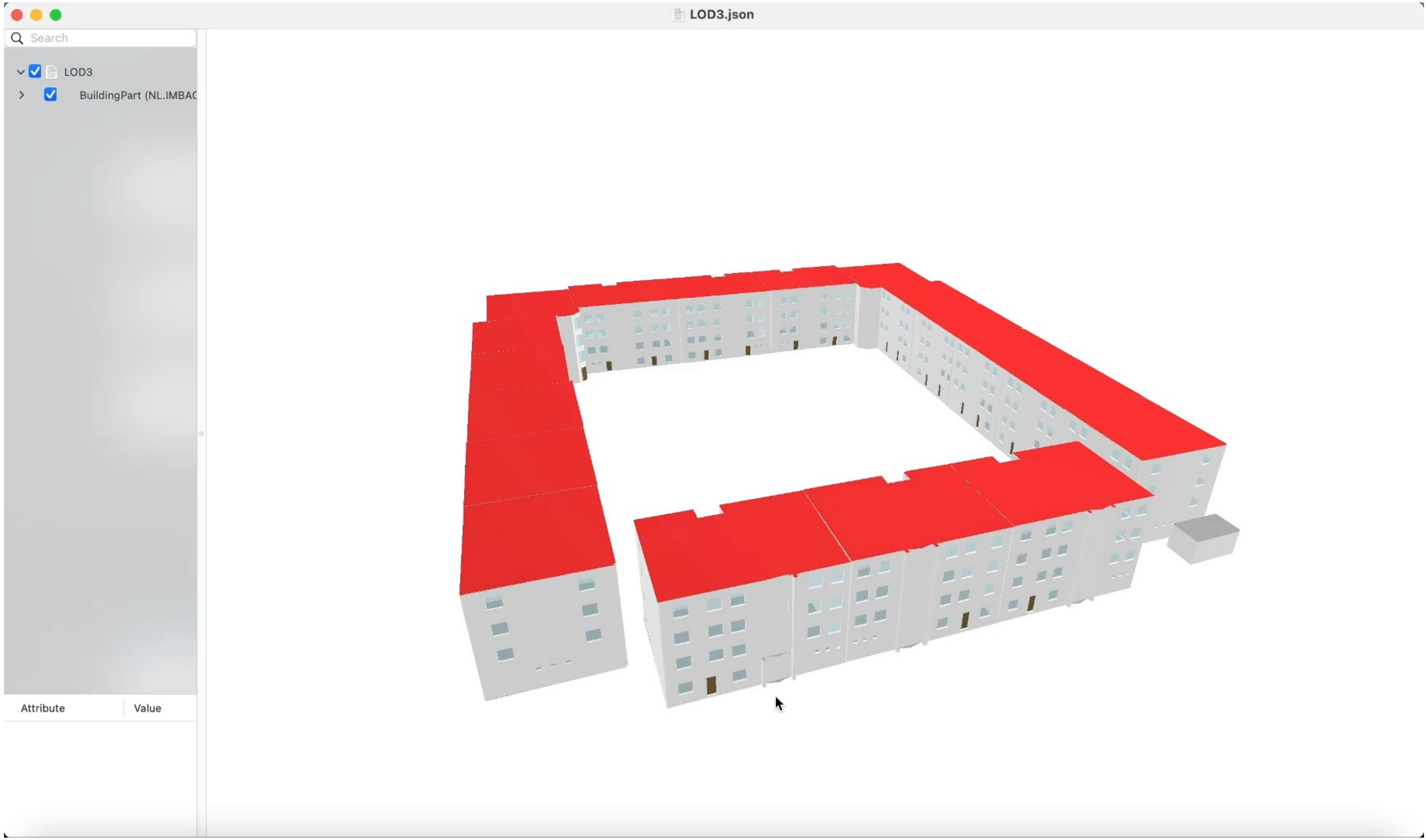
Fig 35 (d). large balcony/obstruction

Fig 35 (e). obstruction

Fig 35 (f). obstruction

TU Delft

LOD3 3D building model



LOD3 3D building model

Fig 36. the LOD3 model and the ground truth images

- Realistic
- Aesthetically pleasing
- Inside structure can be captured as well

TU Delft

Larger area test

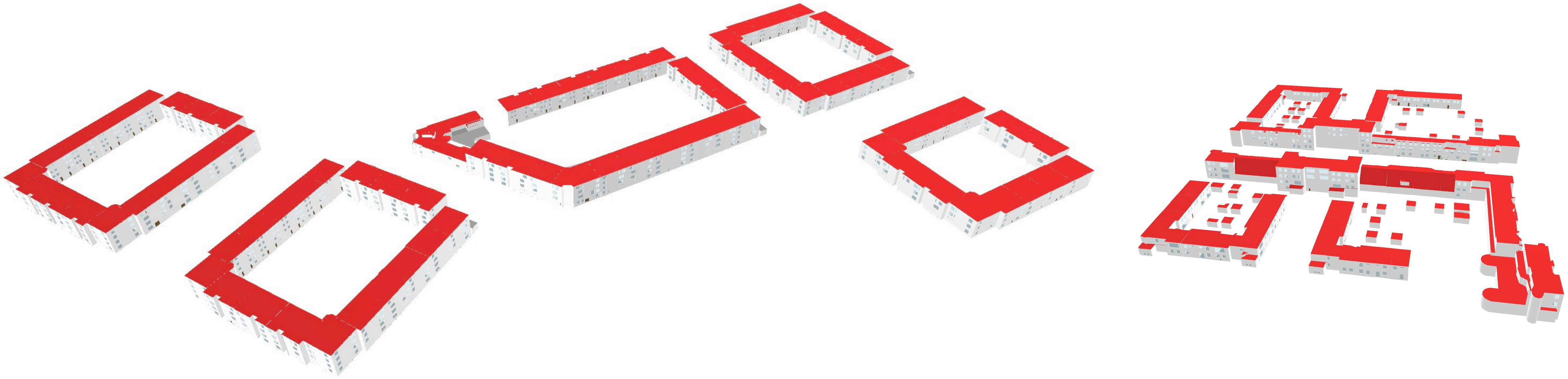


Fig 37. Testing the pipeline on a larger scale and different area in Almere

Failed case: the Amsterdam dataset

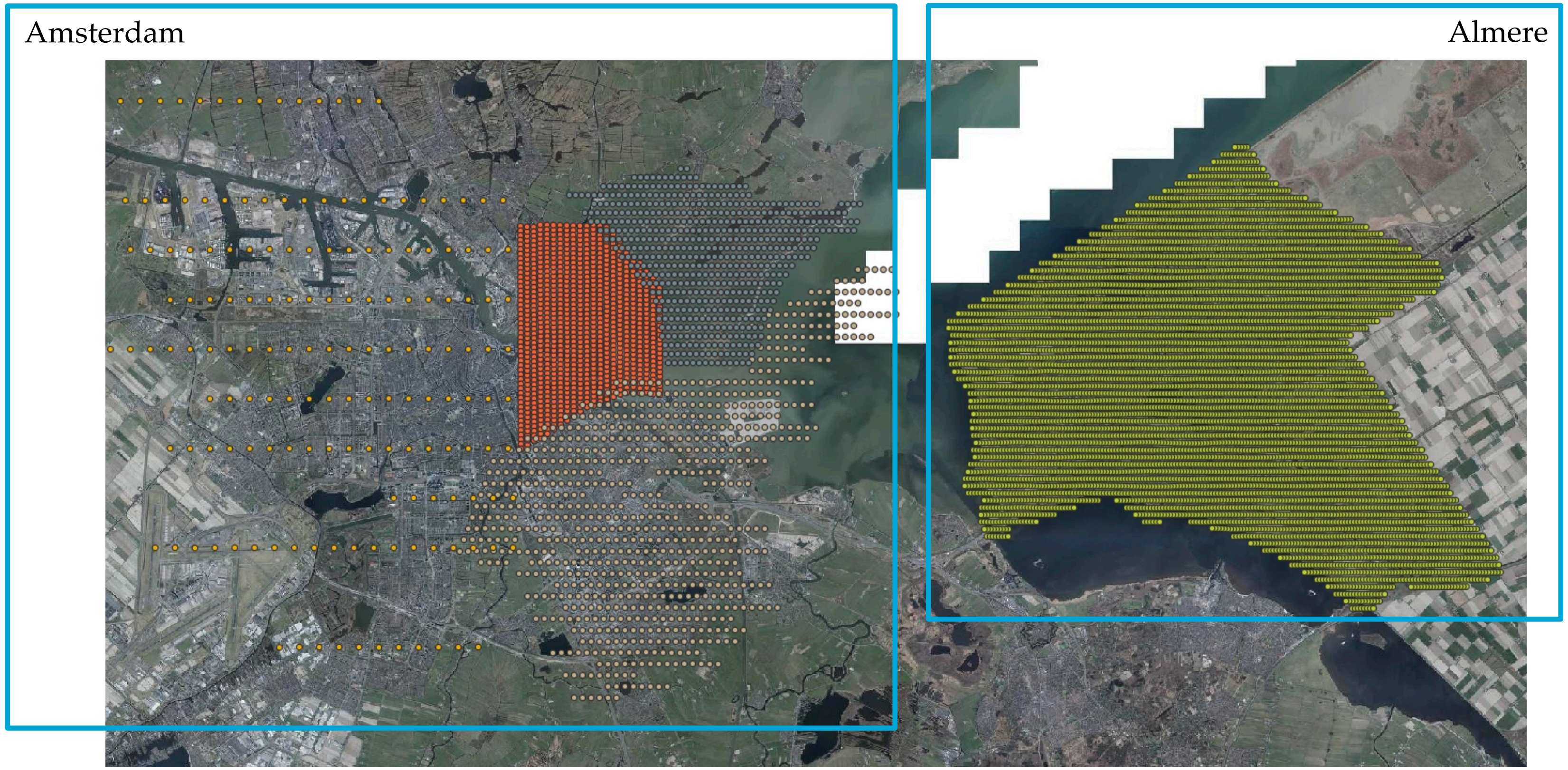
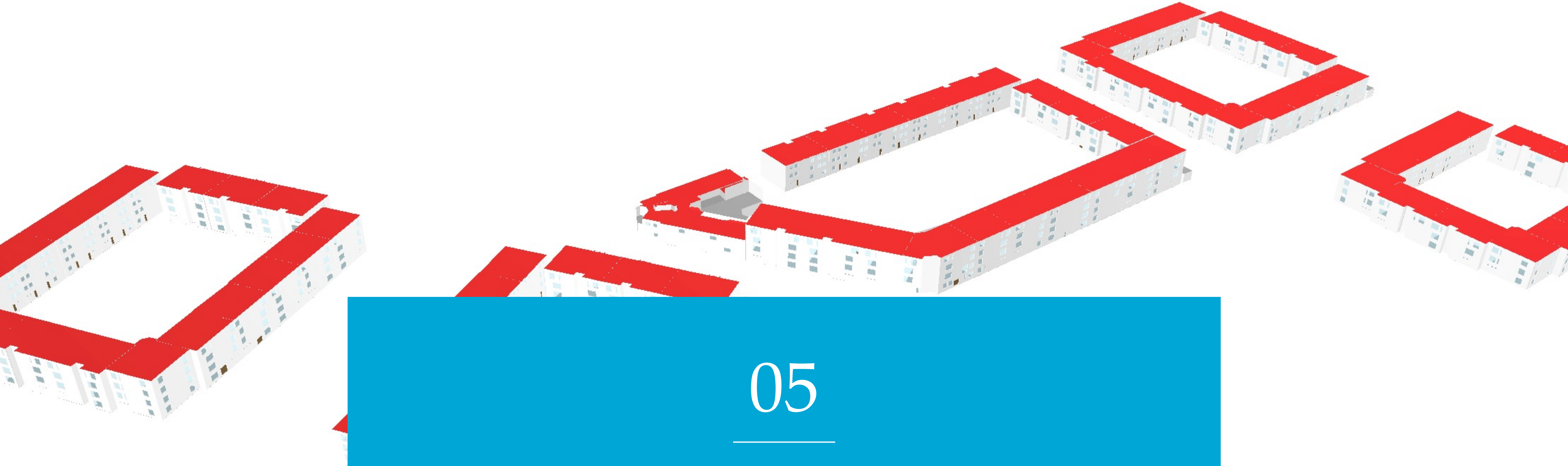


Fig 38. Comparison of data volume (left: Amsterdam dataset, right: Almere dataset)

Insufficient overlap and inadequate images lead to failure of camera parameter estimation.

At least 60% overlap is needed.

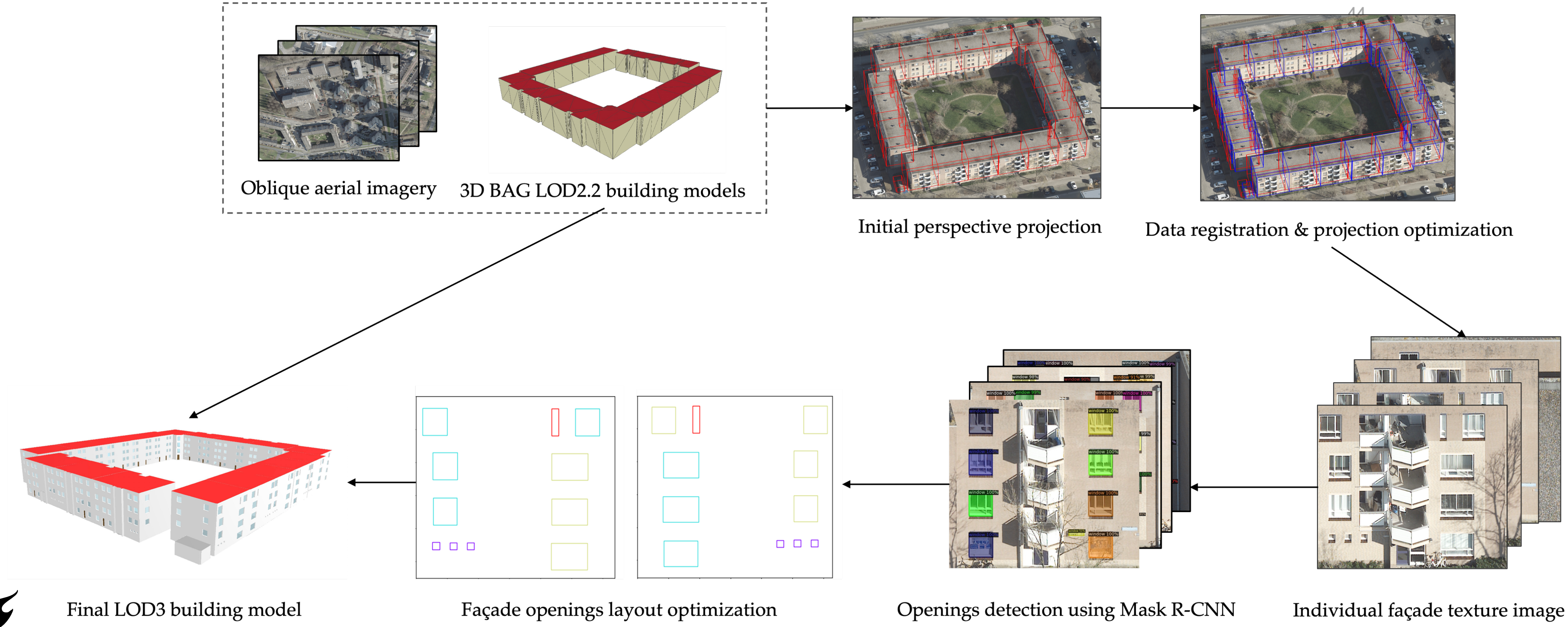


05

Conclusions

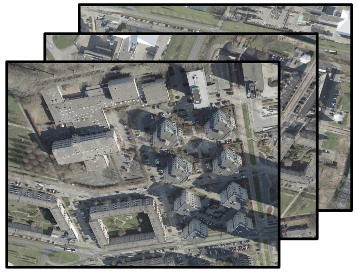
Conclusions

Objective: *Upgrade the 3D BAG LOD2.2 building model to LOD3 by extracting openings from oblique aerial images.*



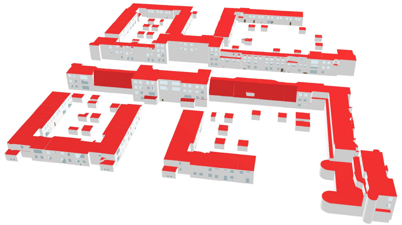
The uniqueness and distinction

Unique data source



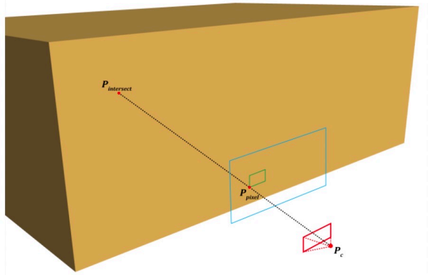
Street view imagery ☺ Oblique aerial imagery

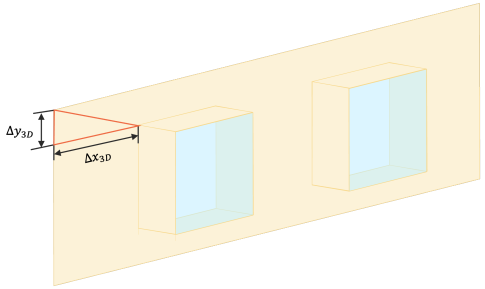
The capability of Larger region reconstruction



Single building reconstruction ☺ Larger region reconstruction

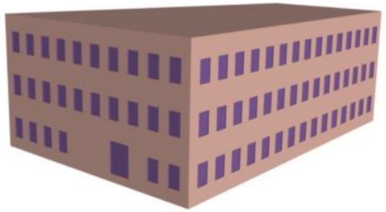
Different 2D-to-3D Conversion method

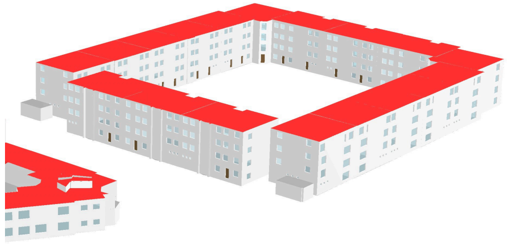




Back projection ☺ Principle of Similar Triangles

Inside structures can be fully captured





Only outward façade can be captured ☺ Both Inside and outside structures can be captured

Limitations

- The process largely relies on the quality of the imagery data;
- No fully automated data registration process;
- Unsatisfactory detection of doors: missing/misclassifications.

Future work

- Combining different data sources: street view imagery and oblique aerial imagery;
- Full automation of data registration;
- Enhance openings detection, especially the detection of the doors.

Thank you for your attention

Yitong Xia

TU Delft