Detailed Facade Reconstruction for Manhattan-world Buildings

Linjun Wang

Supervisors:
Liangliang Nan
Nail Ibrahimli

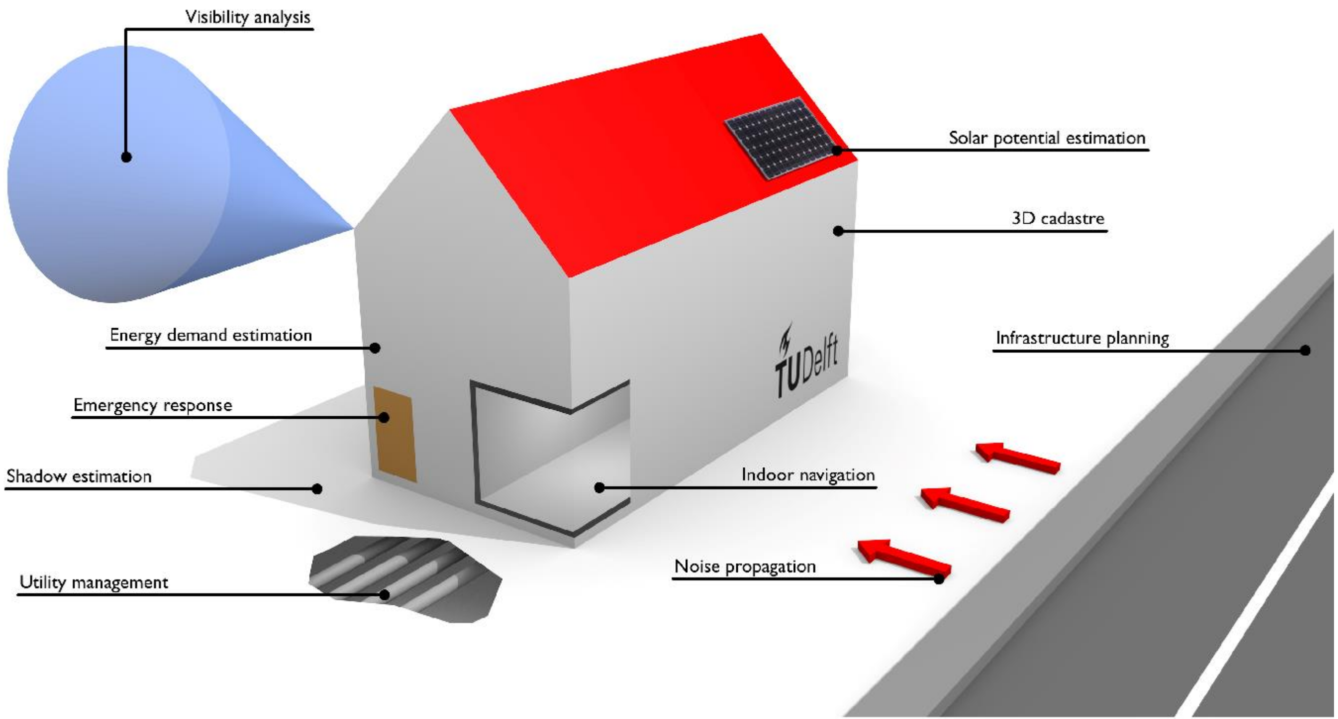
Contents

- Introduction
- Related work
- Methodology
- Results and evaluation
- Conclusions

Contents

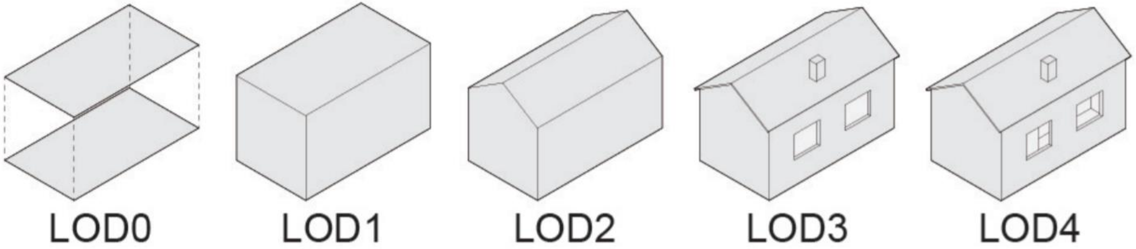
- Introduction
- Related work
- Methodology
- Results and evaluation
- Conclusions

Introduction: 3D building models

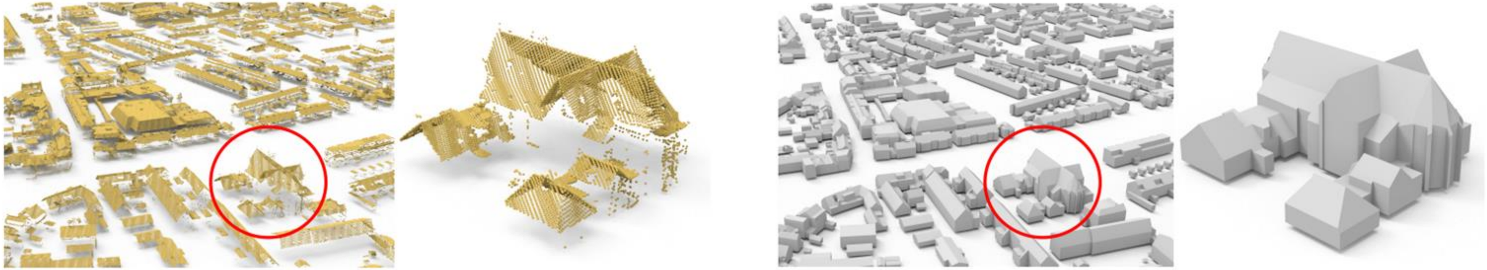


Applications of 3D building models (Biljecki et al.,2015)

Introduction: Level of Detail (LOD)



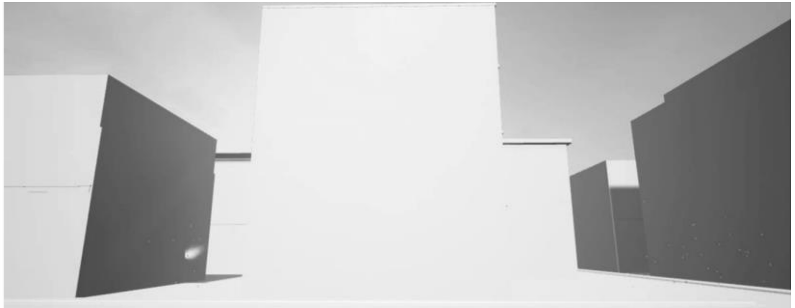
Example of a building modeled at different LODs (Peronato et al.,2016)



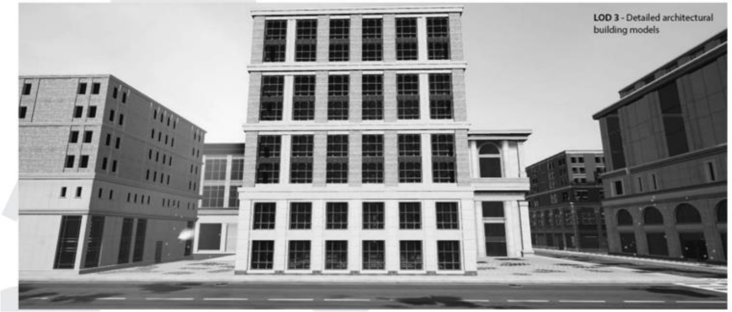
The LOD2 models can be easily obtained using existing reconstruction pipelines (Huang et al.,2022)

Introduction: Level of Detail (LOD)

LOD3 models with fine facade details are crucial for many applications, such as virtual reality and urban simulation.



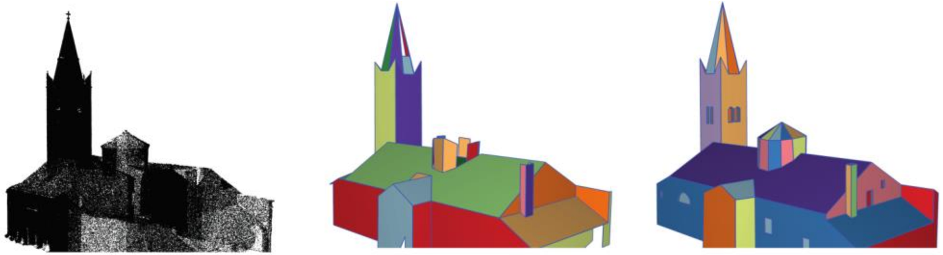
Coarse model in the urban scene



LOD3 model in the urban scene

Introduction: LOD3 model reconstruction

Create model directly from the point cloud



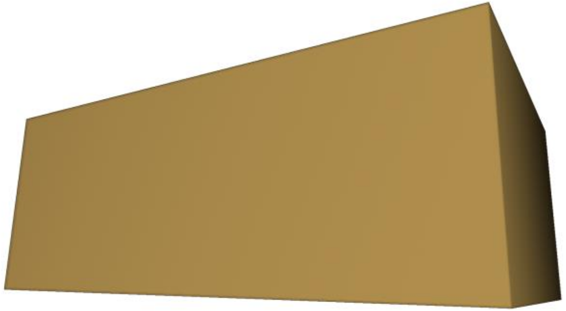
From the input point cloud to the refined model after automatic reconstruction and manual editing (Arikan et al., 2013).

Need to pay many manual efforts to refine the model, which is very time-consuming.

Introduction: Inspiration and research question

Create model based on images

Piecewise planar facades in 3D ↔ Facade photos



Obtain information from the facade images. Reduce the problem to 2D.

Research question:

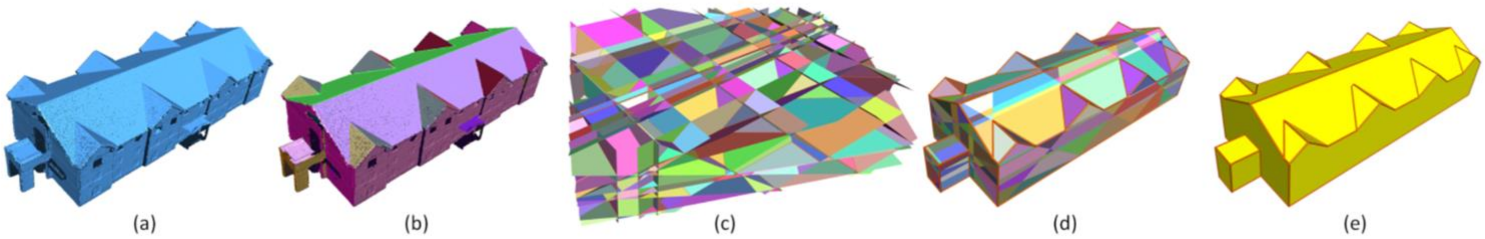
How can we bring more facade details into the image-based urban reconstruction pipeline?

Contents

- Introduction
- **Related work**
- Methodology
- Results and discussion
- Conclusions

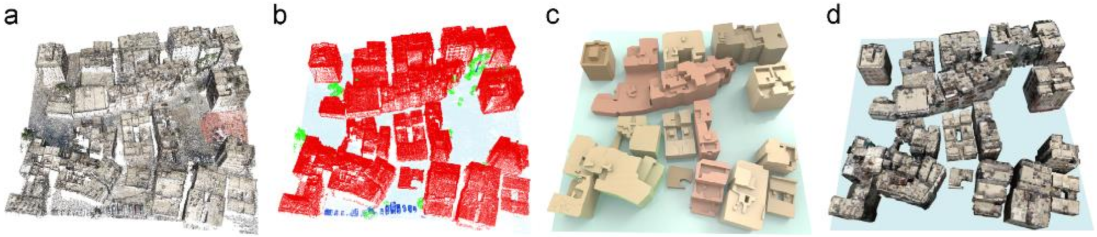
Related work: Coarse model reconstruction

Single building model reconstruction



Polyfit (Nan and Wonka, 2017)

Urban scene reconstruction



Reconstructing building mass models from UAV images (Li et al., 2016)

Related work: Detailed facade reconstruction

Photogrammetry-based reconstruction

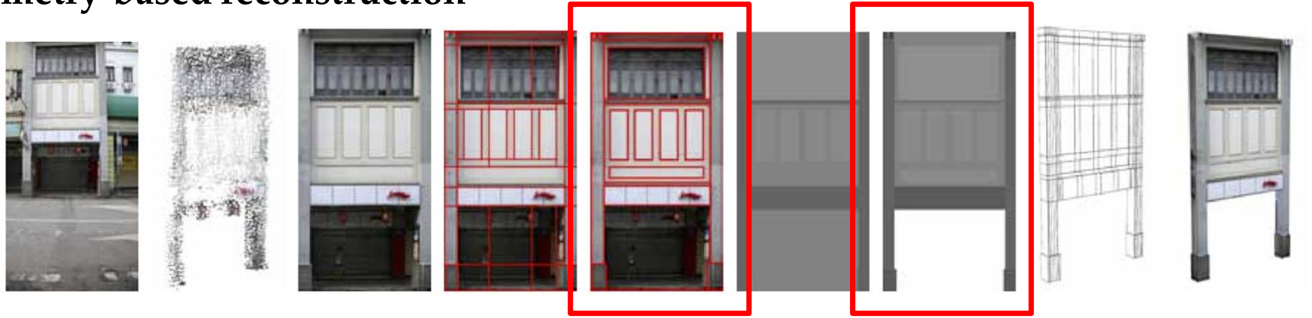
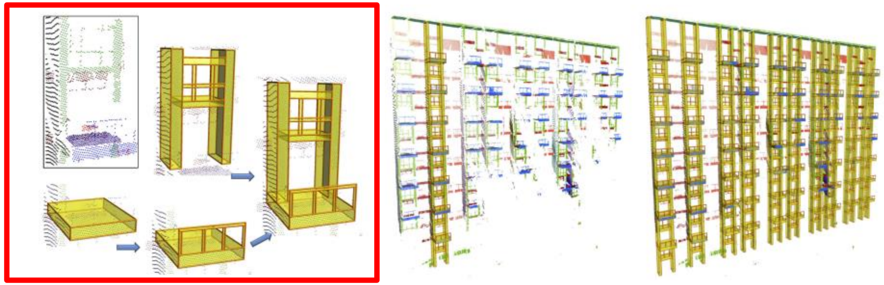


Image-based Facade Modeling (Xiao et al. 2008)

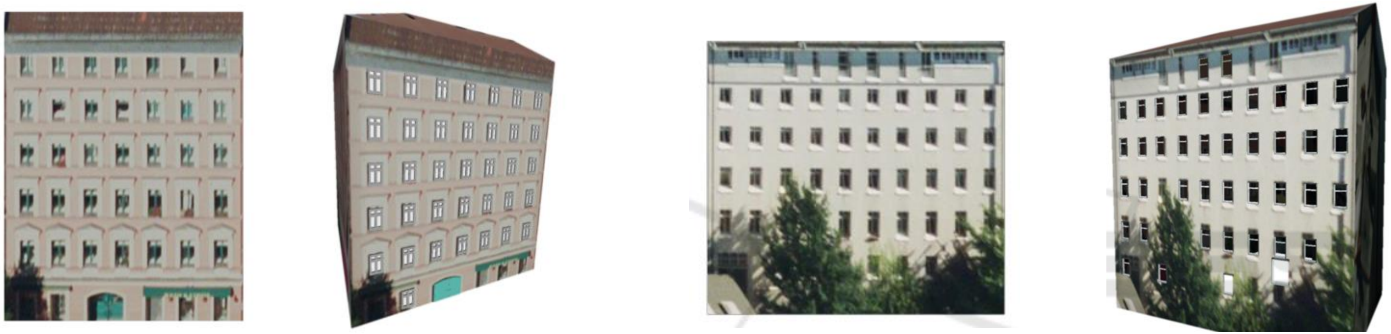
Primitive-based reconstruction



SmartBoxes for Interactive Urban Reconstruction(Nan et al. 2010)

Related work: Detailed facade reconstruction

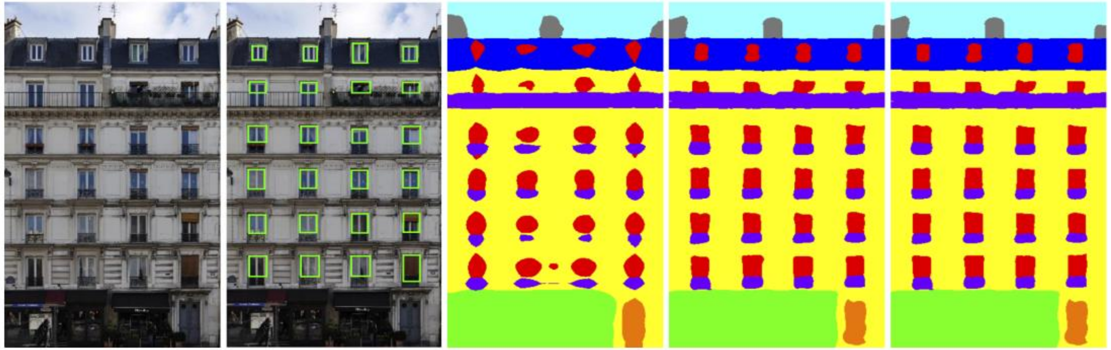
Extend a model from a lower LOD to LOD3 based on the images of the buildings



Enrich the LOD2 CityGML model with facade details using texture images (Zhang et al., 2019)

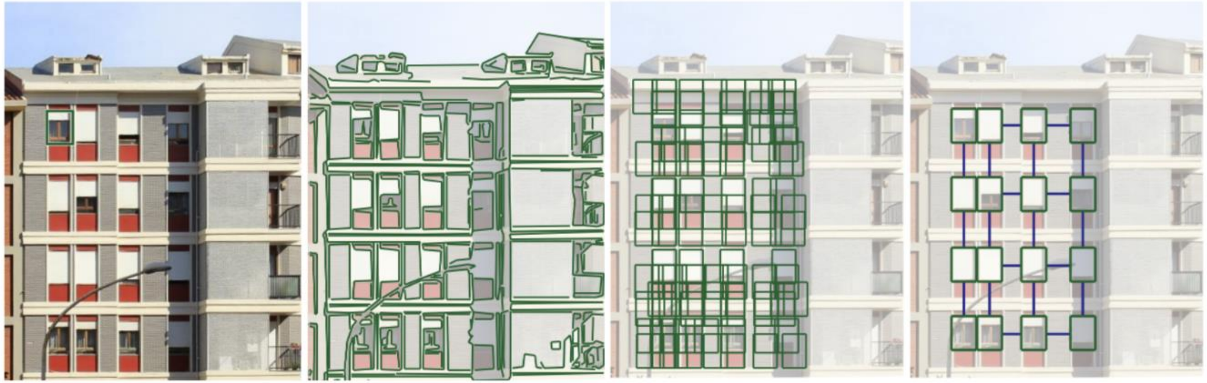
Related work: Facade parsing

Pixel-level:



DeepFacade (Liu et al., 2019)

Subdivision:

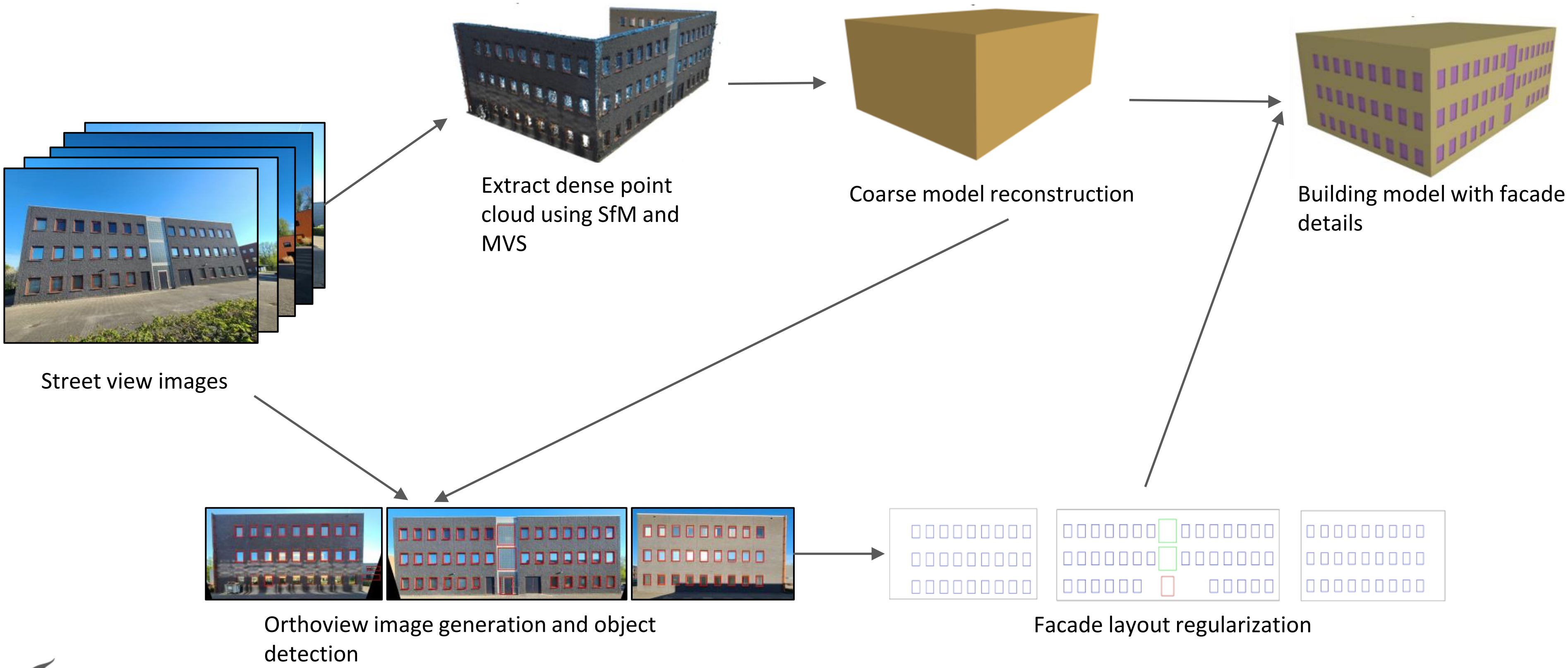


Interactive Facades Analysis and Synthesis of Semi-Regular Facades (Alhalawani et al., 2013)

Contents

- Introduction
- Related work
- Methodology
- Results and discussion
- Conclusions

Methodology: Overview



Street view images

Extract dense point cloud using SfM and MVS

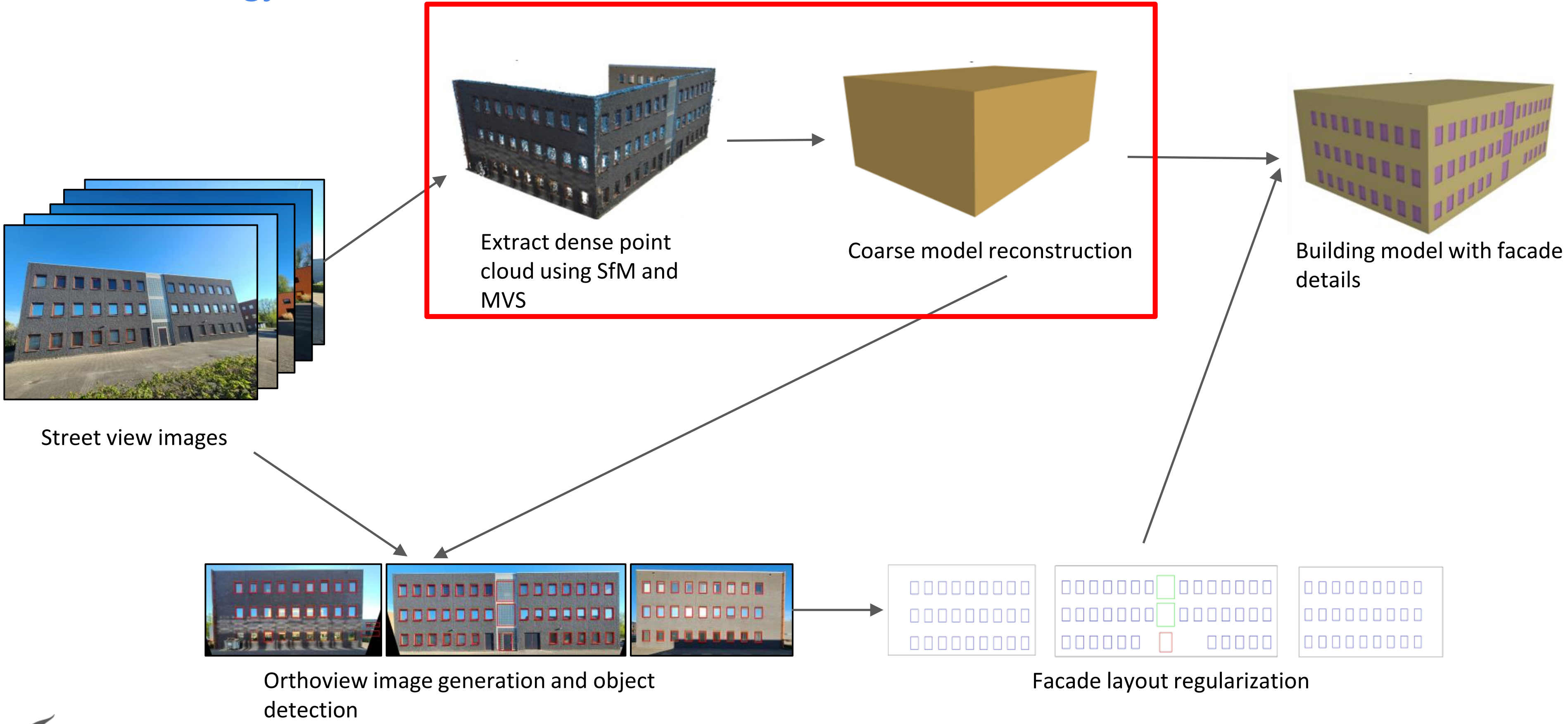
Coarse model reconstruction

Building model with facade details

Orthoview image generation and object detection

Facade layout regularization

Methodology: Overview



Street view images

Extract dense point cloud using SfM and MVS

Coarse model reconstruction

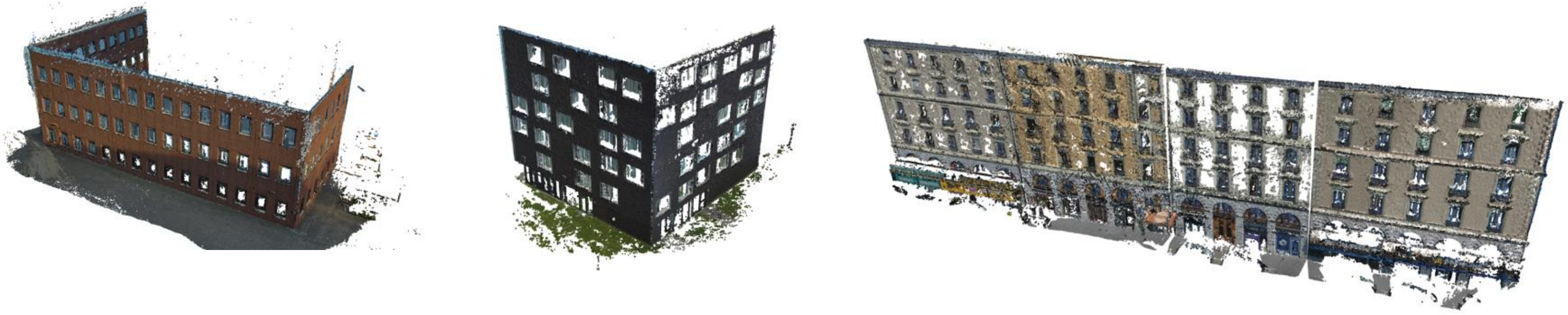
Building model with facade details

Orthoview image generation and object detection

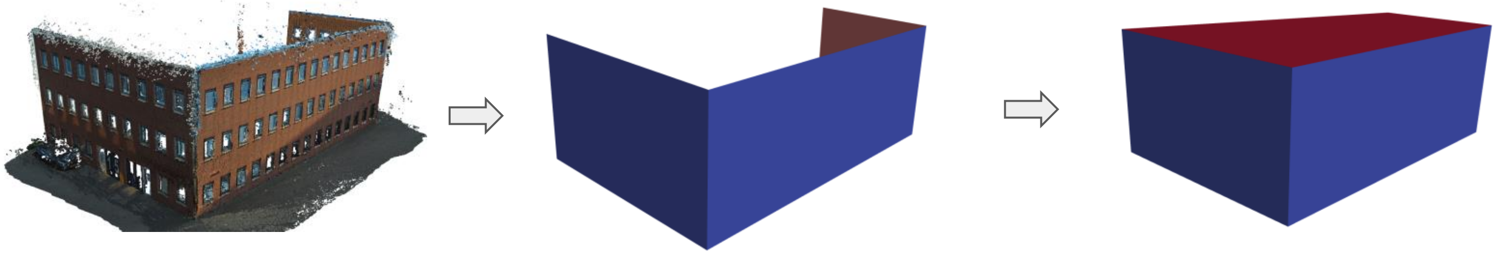
Facade layout regularization

Methodology: Coarse model reconstruction

The point clouds extracted from street view images may not contain all the facades around the buildings:



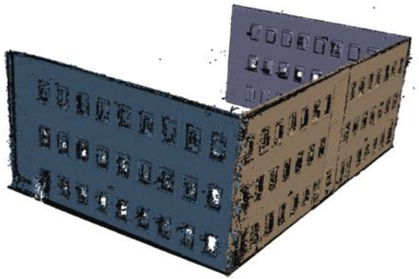
Reconstruct the walls first. Then complete the building blocks as closed models:



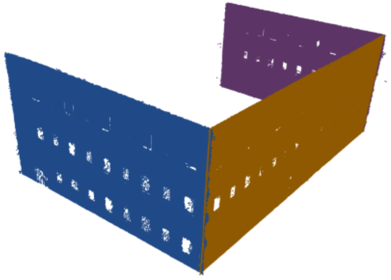
Methodology: Coarse model reconstruction

1. Initialization

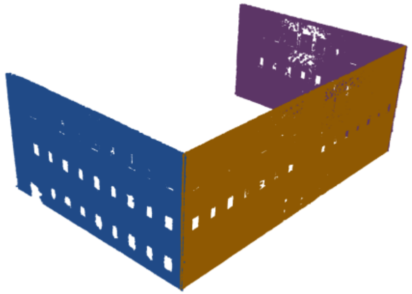
Input point cloud



RANSAC



Project points to the corresponding plane

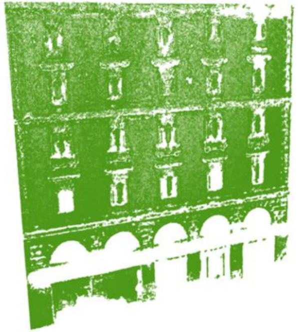


Remove outliers for every face using the **statistical outlier removal algorithm (Rusu, PhD thesis, 2009)**

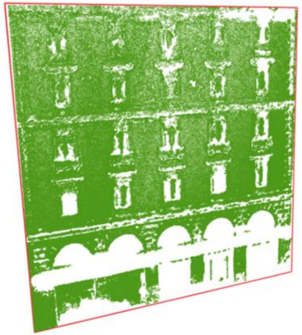
Methodology: Coarse model reconstruction

2. One face situation

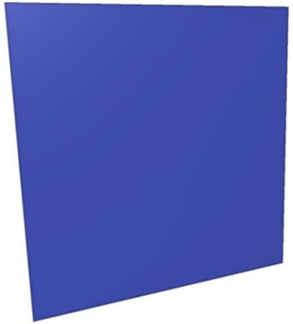
Input point cloud



Initialization result



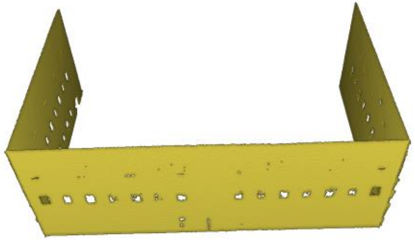
Obtain the **minimum area bounding box(Freeman et al., 1975)**



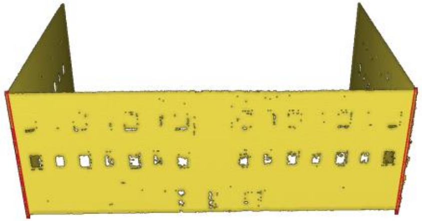
Reconstructed facade face

Methodology: Coarse model reconstruction

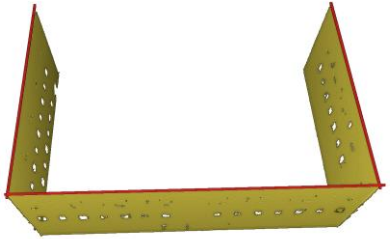
3. Multi-face situation



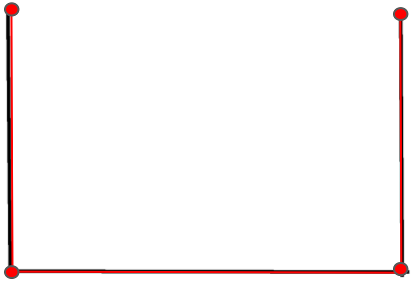
Initialization result



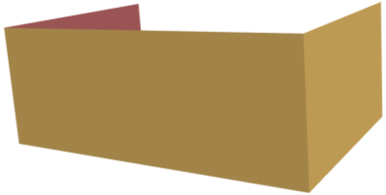
Find the intersection lines between planes and project the points to the lines



Project the points to the bottom plane

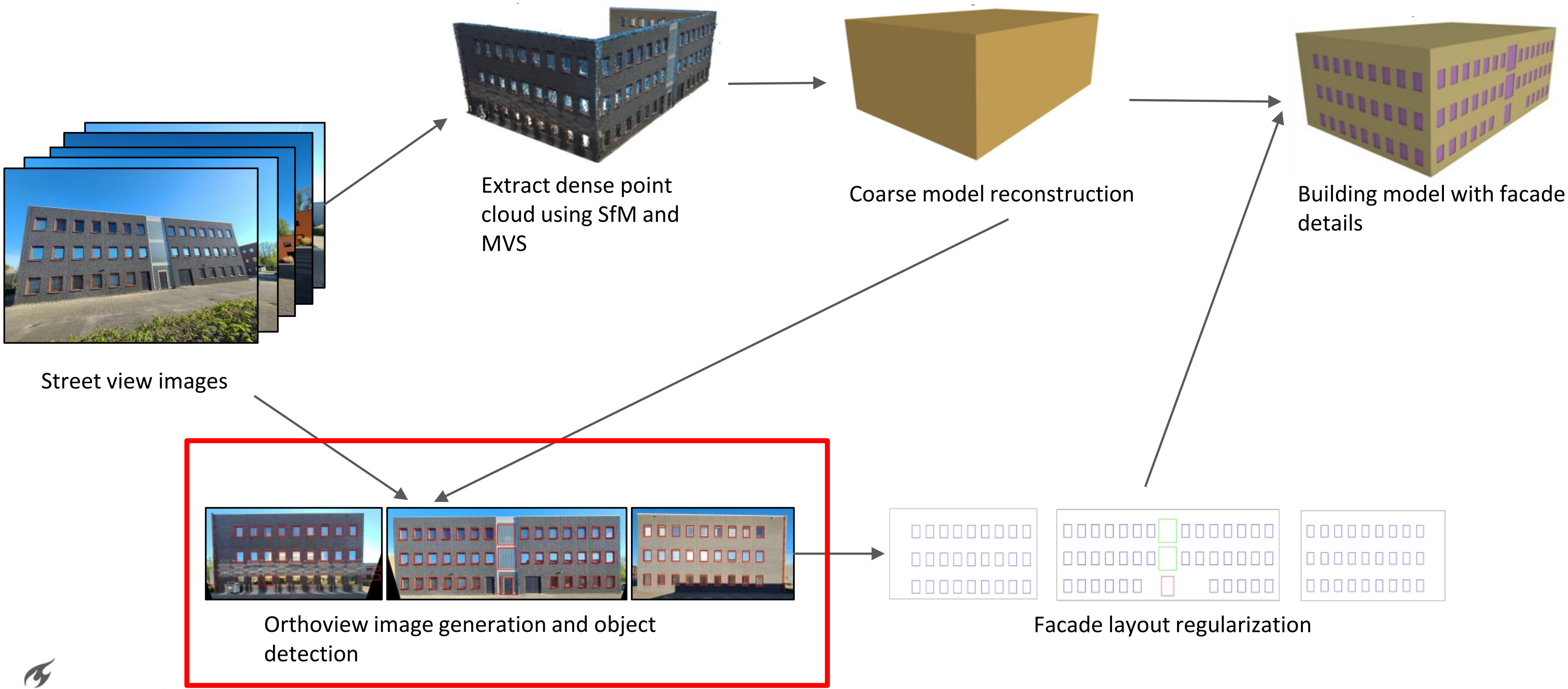


Construct the footprint graph based on the projected lines



Extrude the footprint to get the reconstructed facade faces

Methodology: Overview



Street view images

Extract dense point cloud using SfM and MVS

Coarse model reconstruction

Building model with facade details

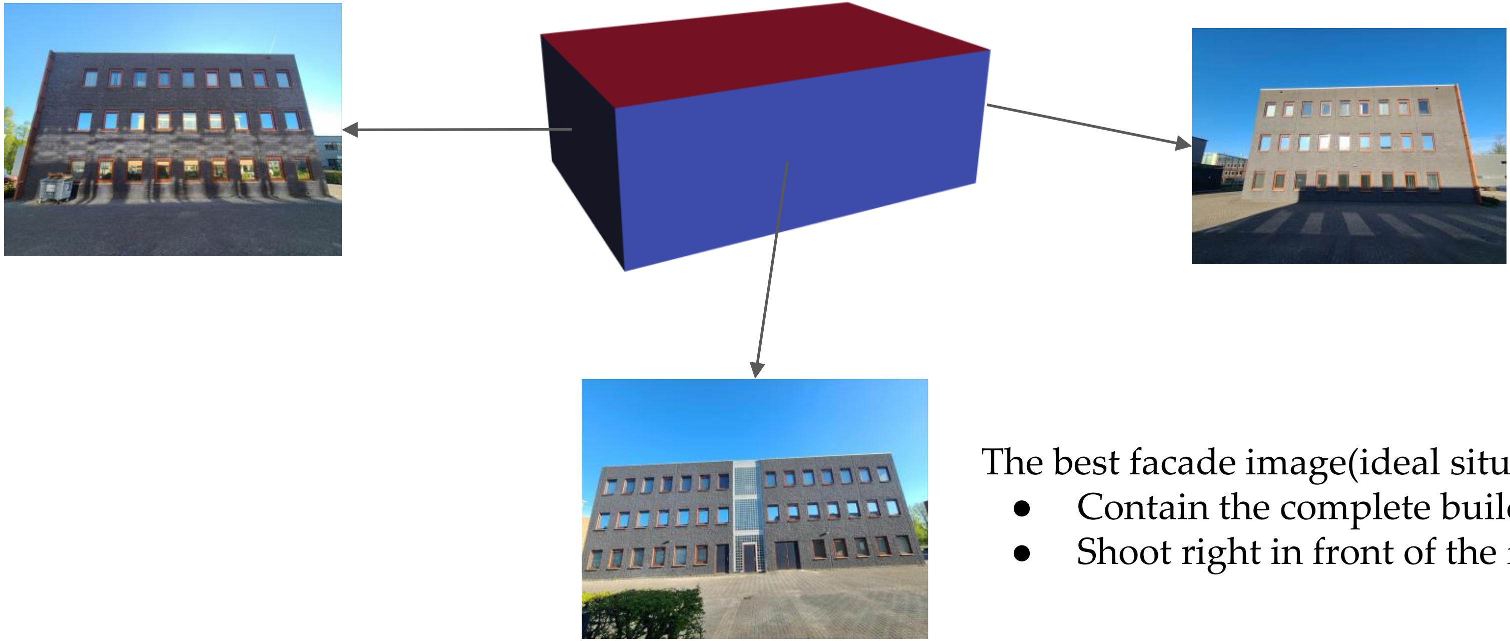
Orthoview image generation and object detection

Facade layout regularization

Methodology: Orthoview image generation

1. Automatic facade image selection

Need to select the best facade image for every facade face:



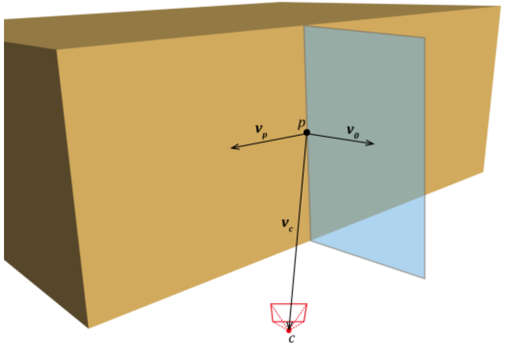
The best facade image(ideal situation):

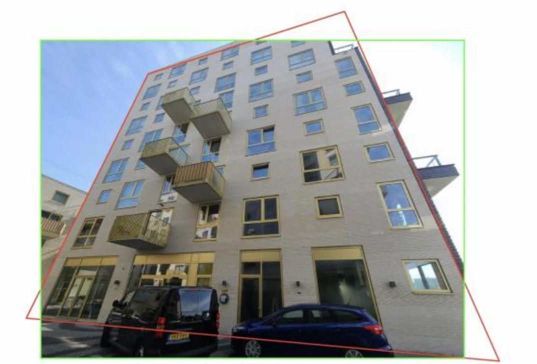
- Contain the complete building facade
- Shoot right in front of the facade

Methodology: Orthoview image generation

1. Automatic facade image selection

The camera should be as close to the central orthogonal plane (the blue plane in the left figure) as possible. The image should contain as much facade area as possible.





Four vectors:

V0: the facade face's normal vector
Vp: the central orthogonal plane's normal vector
Vc: vector from the face centroid to the camera center
Vf: forward direction of the camera

Two polygons:

Pf: the projected facade polygon on the image plane (red)
Pi: the image rectangle (green)

(1) The angle between **Vf** and **V0** should be larger than 90 degrees.
(2) The percentage $area(P_i \cap P_f)/area(P_f)$ should be larger than 0.9.

→ **Must be met**

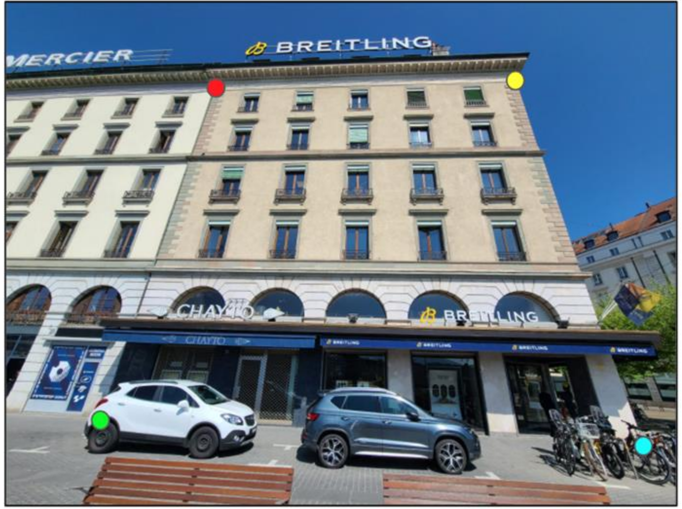
(3) The angle between **Vc** and **Vp** should be as close to 90 degrees as possible.

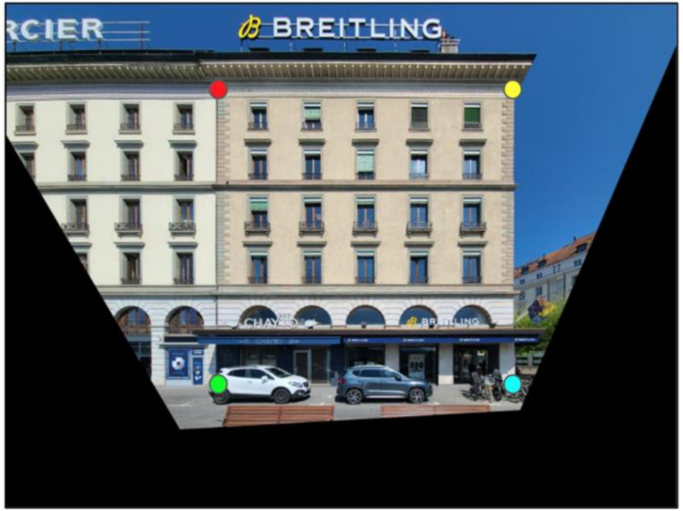
→ **Compare and find the smallest one**

Methodology: Orthoview image generation

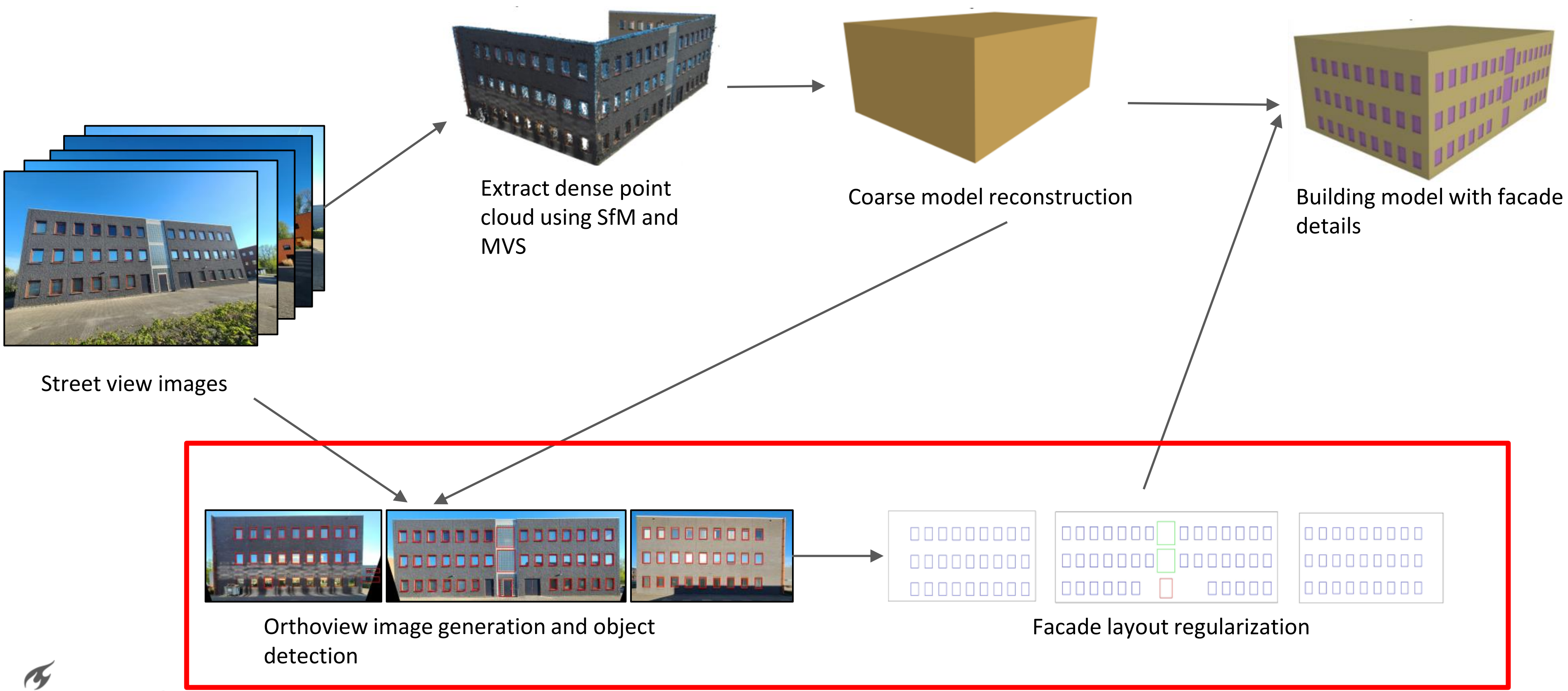
2. Facade image rectification

Perspective Transformation: Given the four corners of the rectangle in the original image and the destination image, we can obtain the transformation matrix.





Methodology: Overview



Street view images

Extract dense point cloud using SfM and MVS

Coarse model reconstruction

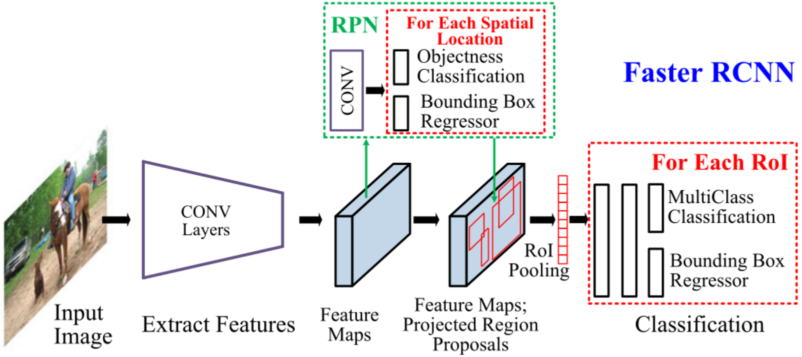
Building model with facade details

Orthoview image generation and object detection

Facade layout regularization

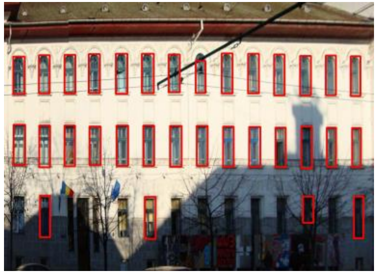
Methodology: Facade element detection and regularization

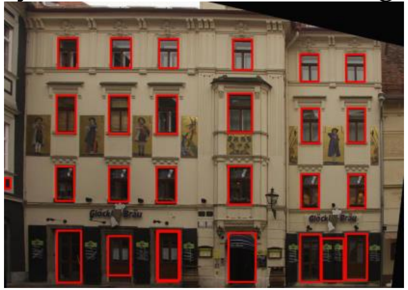
1. Facade elements detection based on Faster R-CNN (Ren et al., 2015)

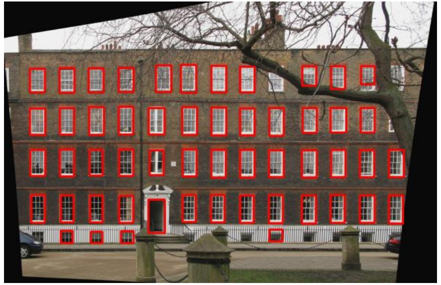


High-level framework of Faster R-CNN (Liu et al., 2018).

We apply the Faster R-CNN model with the ResNet50 FPN backbone. The detected bounding boxes can match pretty well with the rectangular shape of most facade elements.



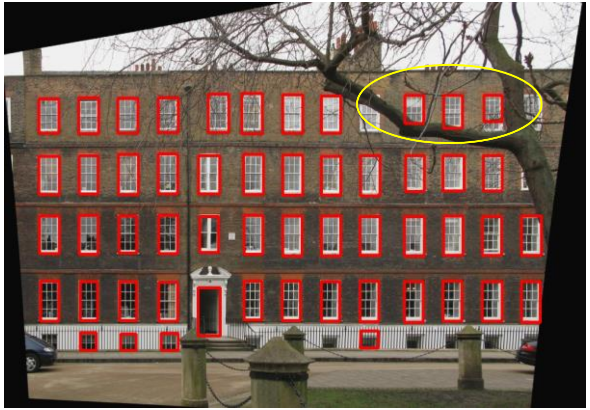




Methodology: Facade element detection and regularization

2. Facade layout regularization

The detection result may have errors such as missing or misplaced facade elements:

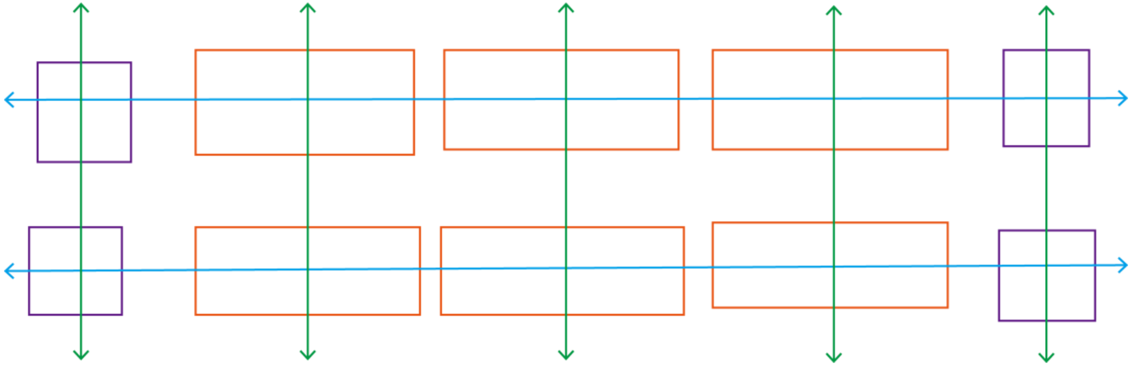


In order to derive more convincing models, we need to define some architectural principles to regularize the facade layout.

Methodology: Facade element detection and regularization

2. Facade layout regularization

Three constraints:



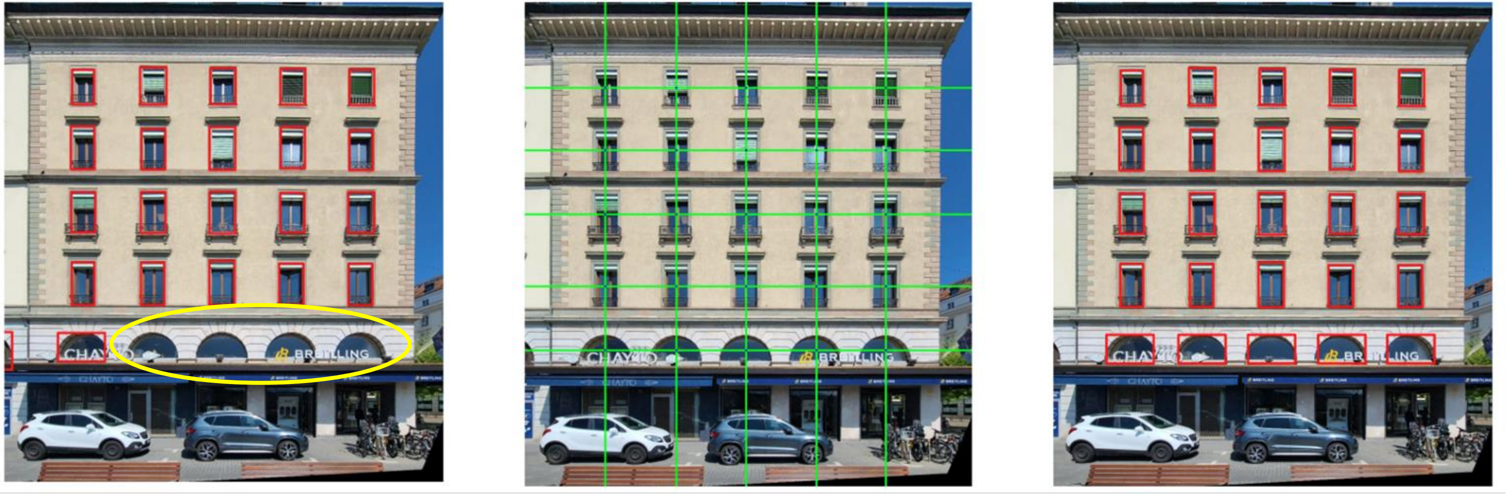
- Boxes along the blue line should be *horizontally aligned*.
- Boxes along the green line should be *vertically aligned.*
- Boxes with similar sizes should have the *same size.*

Boxes under different constraints will be divided into different constraint groups.

Methodology: Facade element detection and regularization

2. Facade layout regularization

We can add missing boxes based on the global regularity.



Treat the layout as a grid. The average coordinate value for every row and column can be calculated.

Methodology: Facade element detection and regularization

2. Facade layout regularization

Regularization: Solve an optimization problem to change the boxes' locations and sizes.

Energy term for changes in element locations: $\boldsymbol{E_l} = \sum_{i=1}^{n}(x_i^* + \frac{w_i^*}{2} - x_i - \frac{w_i}{2})^2 + (y_i^* + \frac{h_i^*}{2} - y_i - \frac{h_i}{2})^2$

Energy term for changes in element sizes: $\boldsymbol{E_s} = \sum_{i=1}^{n}(w_i^* - w_i)^2 + (h_i^* - h_i)^2$

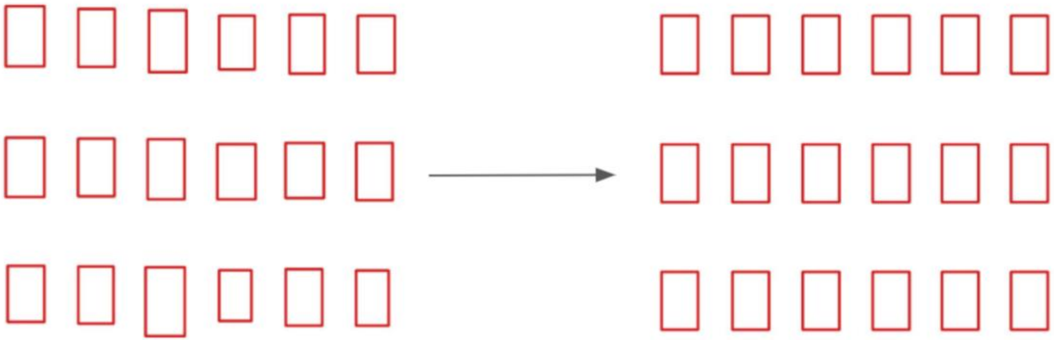
Overall objective function to be minimized: $\boldsymbol{w} \cdot \boldsymbol{E_l} + \boldsymbol{E_s}$

Where:

- *(xi, yi, wi, hi)*: location and size of the initial bounding box
- *(x*, y*, w*, h*)*: location and size of the changed bounding box
- *w*: a weight to balance the two terms

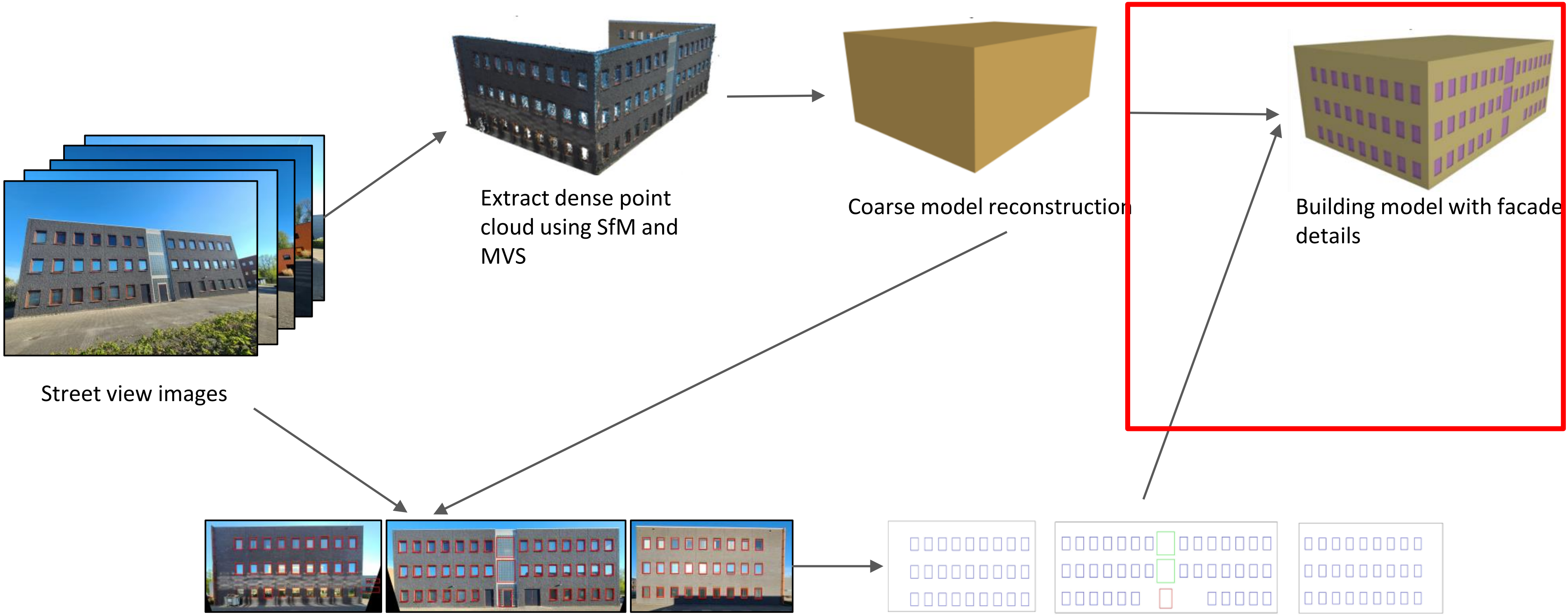
Methodology: Facade element detection and regularization

2. Facade layout regularization



Original and regularized layout of 18 detected windows

Methodology: Overview



Street view images

Extract dense point cloud using SfM and MVS

Coarse model reconstruction

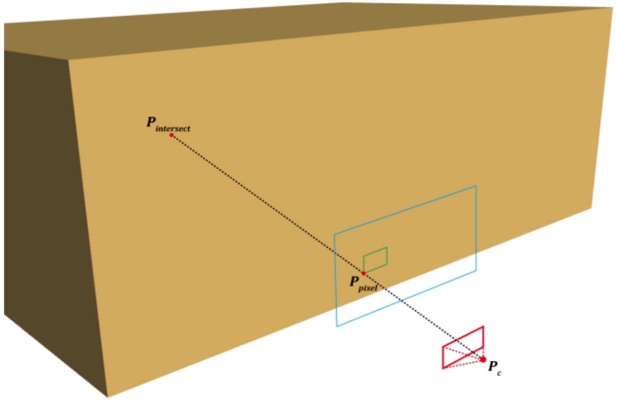
Building model with facade details

Orthoview image generation and object detection

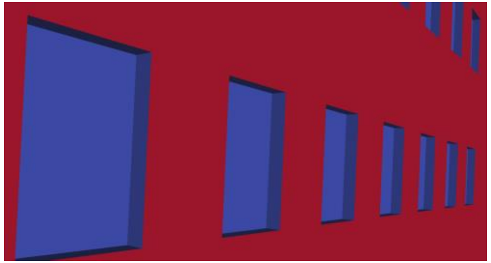
Facade layout regularization

Methodology: Detailed facade reconstruction

Based on the camera parameters and the 3D facade face, we can project the bounding boxes from 2D to 3D.



After the projection of the rectangles, we can extrude or intrude the rectangles to add the final facade elements to the model.



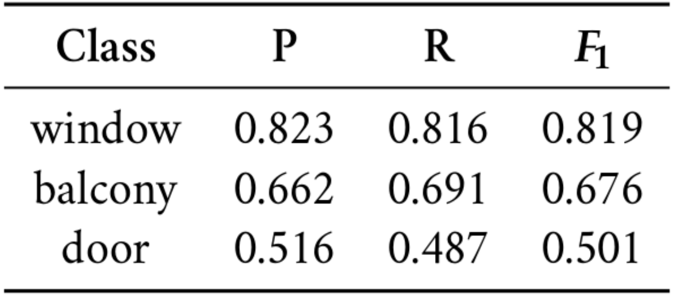
The depth of extrusion/intrusion: The median value of the distance between the points inside the rectangle and the facade plane.

Contents

- Introduction
- Related work
- Methodology
- **Results and evaluation**
- Conclusions

Results & evaluation: Object detection evaluation

The evaluation metrics are: precision(P), recall(R), F1 score.



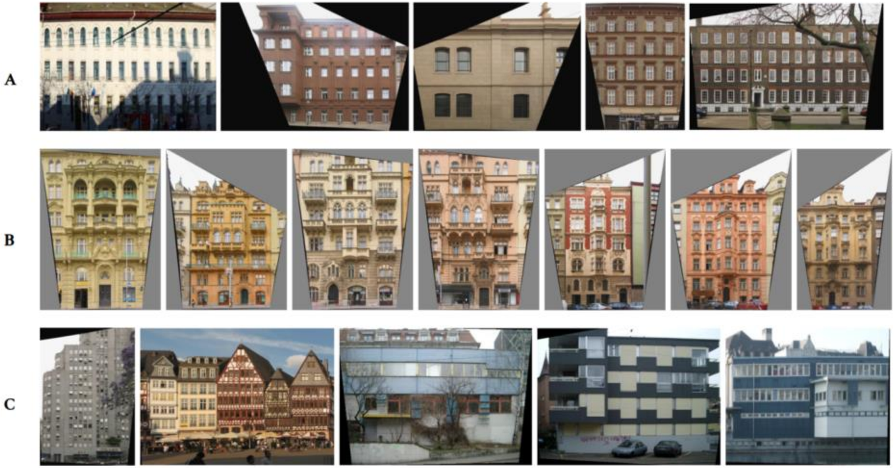
Class	P	R	F_1
window	0.823	0.816	0.819
balcony	0.662	0.691	0.676
door	0.516	0.487	0.501

Performance of facade elements detection using all the test data.

The performance of window detection is much better than the other two classes.

Results & evaluation: Object detection evaluation

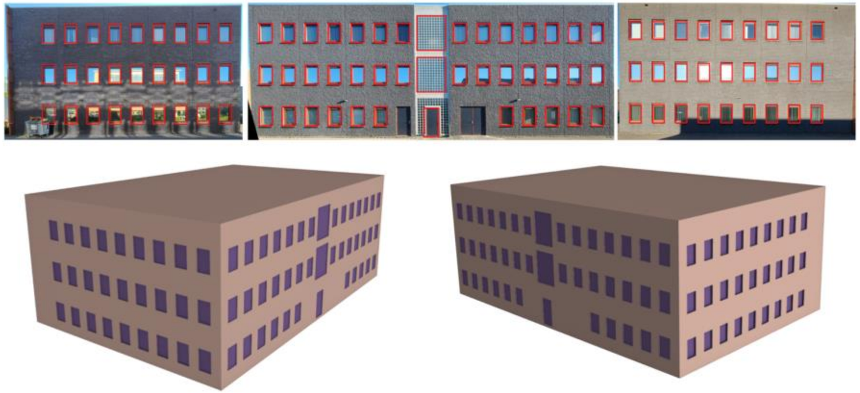
We select three groups of images according to the facade styles to test the window detection performance in different situations:

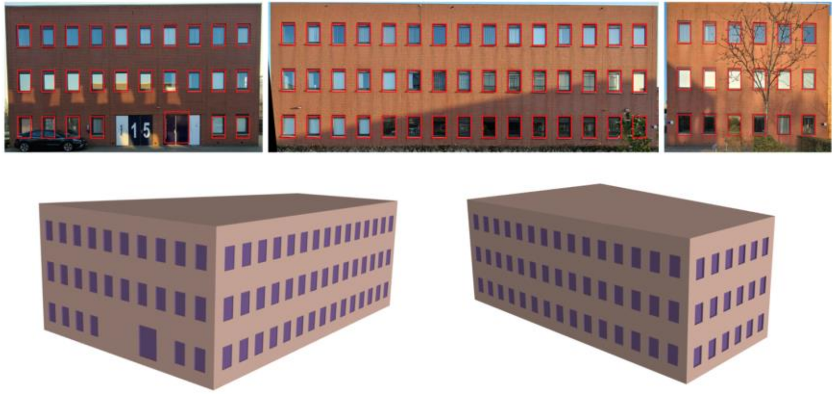


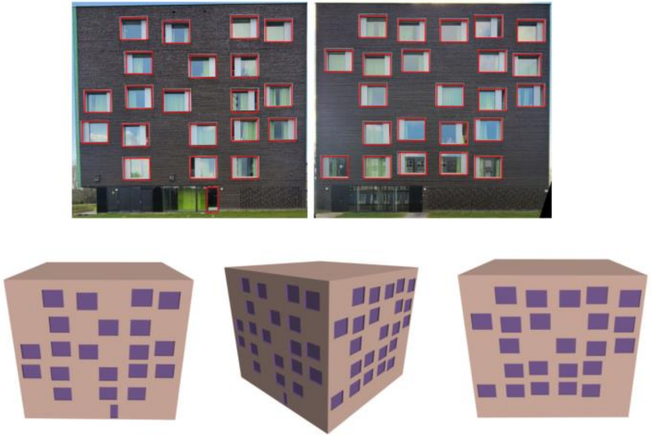
Group	P	R	F_1
A	0.915	0.872	0.893
B	0.878	0.744	0.805
C	0.869	0.457	0.599

Performance of window detection using facade images from different groups.

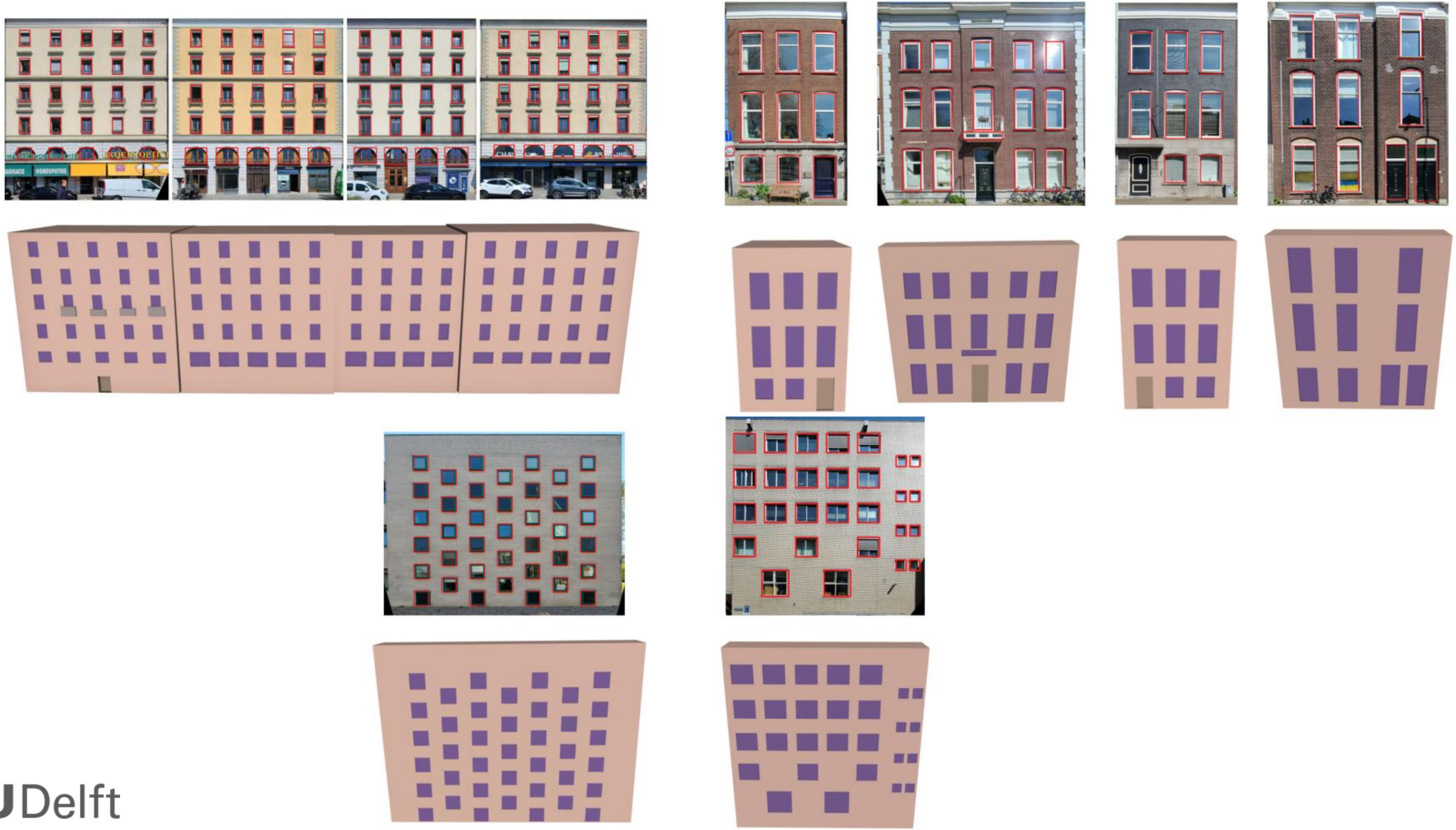
Results & evaluation: Reconstruction results



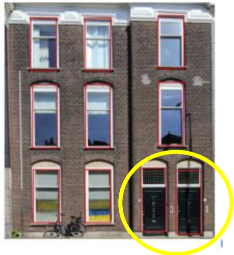


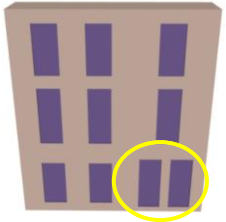


Results & evaluation: Reconstruction results



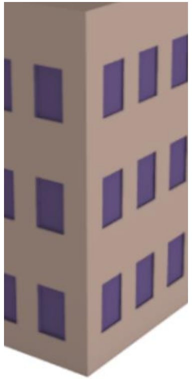
Results & evaluation: Reconstruction results



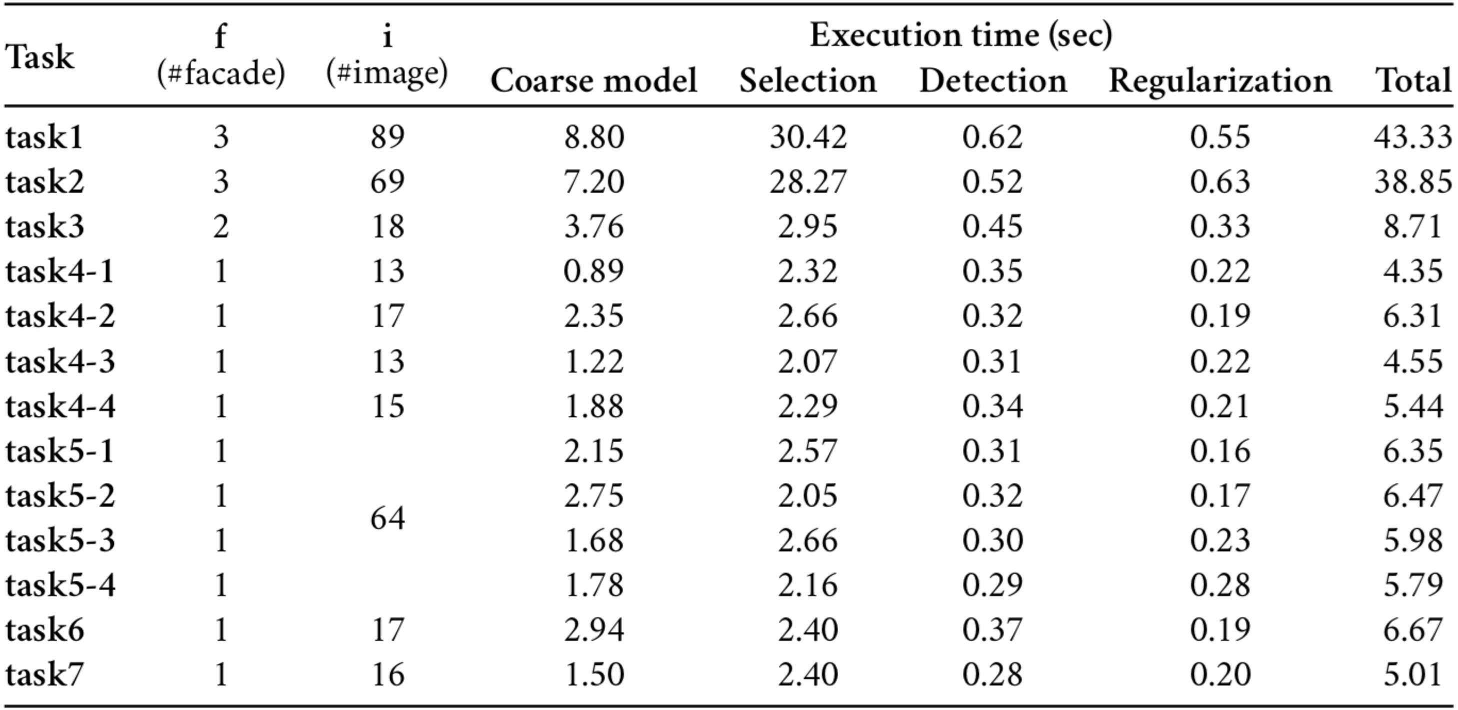


Evaluation of the reconstruction results:

- Good performance for windows
- Bad performance for doors and balconies
- Not aligned on different walls



Results & evaluation: Efficiency



Task	f	i	Execution time (sec)				
	(#facade)	(#image)	Coarse model	Selection	Detection	Regularization	Total
task1	3	89	8.80	30.42	0.62	0.55	43.33
task2	3	69	7.20	28.27	0.52	0.63	38.85
task3	2	18	3.76	2.95	0.45	0.33	8.71
task4-1	1	13	0.89	2.32	0.35	0.22	4.35
task4-2	1	17	2.35	2.66	0.32	0.19	6.31
task4-3	1	13	1.22	2.07	0.31	0.22	4.55
task4-4	1	15	1.88	2.29	0.34	0.21	5.44
task5-1	1	64	2.15	2.57	0.31	0.16	6.35
task5-2	1		2.75	2.05	0.32	0.17	6.47
task5-3	1		1.68	2.66	0.30	0.23	5.98
task5-4	1		1.78	2.16	0.29	0.28	5.79
task6	1	17	2.94	2.40	0.37	0.19	6.67
task7	1	16	1.50	2.40	0.28	0.20	5.01

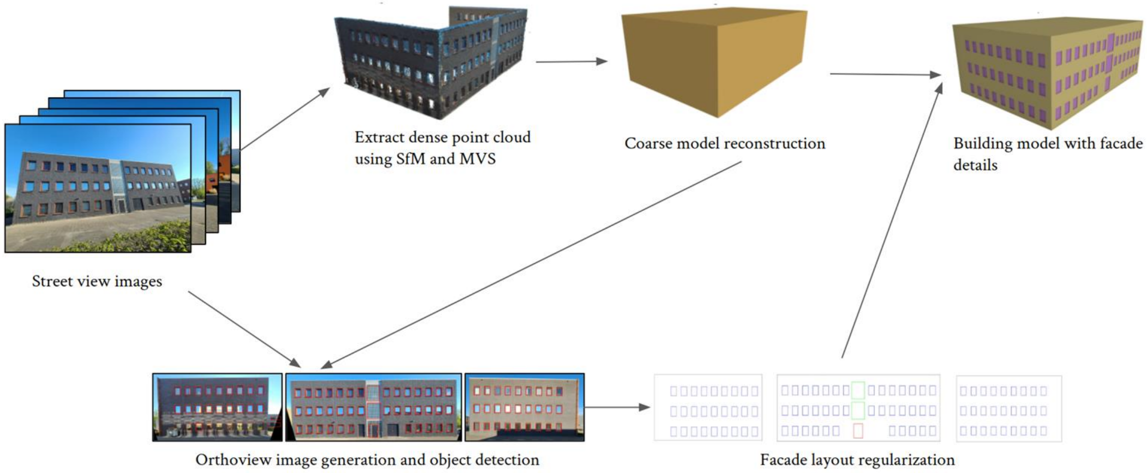
Summary of the runtime of every step in the pipeline.

Contents

- Introduction
- Related work
- Methodology
- Results and evaluation
- Conclusions

Conclusions: Research question

How can we bring more facade details into the image-based urban reconstruction pipeline?


Extract dense point cloud using SfM and MVS
Coarse model reconstruction
Building model with facade details
Street view images
Orthoview image generation and object detection
Facade layout regularization

- Efficient
- Regular
- Automatic

Conclusions: Limitations

- If no image contains all the façade elements, this method will fail.
- Largely depend on the quality of the object detection.
- Doors and balconies are poorly detected.

Conclusions: Future work

- Investigate the façade styles and their layout principles.
- Enhance the Faster R-CNN for façade element detection.
- Apply this method to large-scale city modeling.

… …

Large-scale coarse models

Google Street View images

UAV images

Thanks!