Straightening and simplifying a multi view stereo mesh of a city

Yuxuan Kang

Mentor #1: Hugo Ledoux
Mentor #2: Abdoulaye Diakité

TU Delft

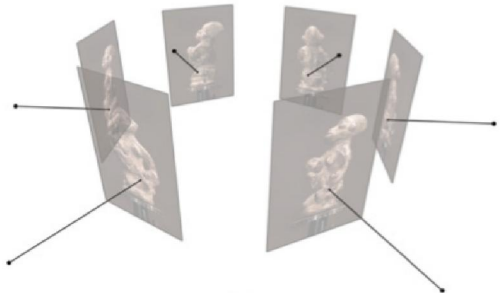
Data: Amsterdam 3D

Amsterdam 3D from Cyclo-Media Technology, Inc

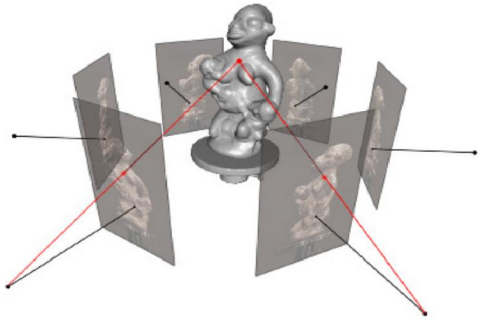
900km roads and 100km 2 area

Generation of MVS mesh

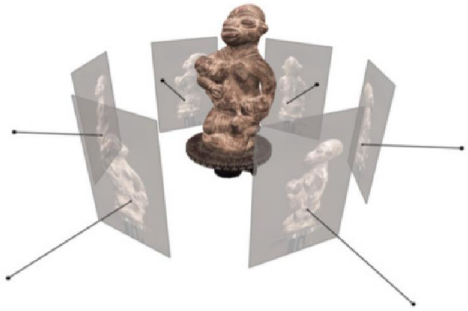
(a) Multi view images



(b) Posed images

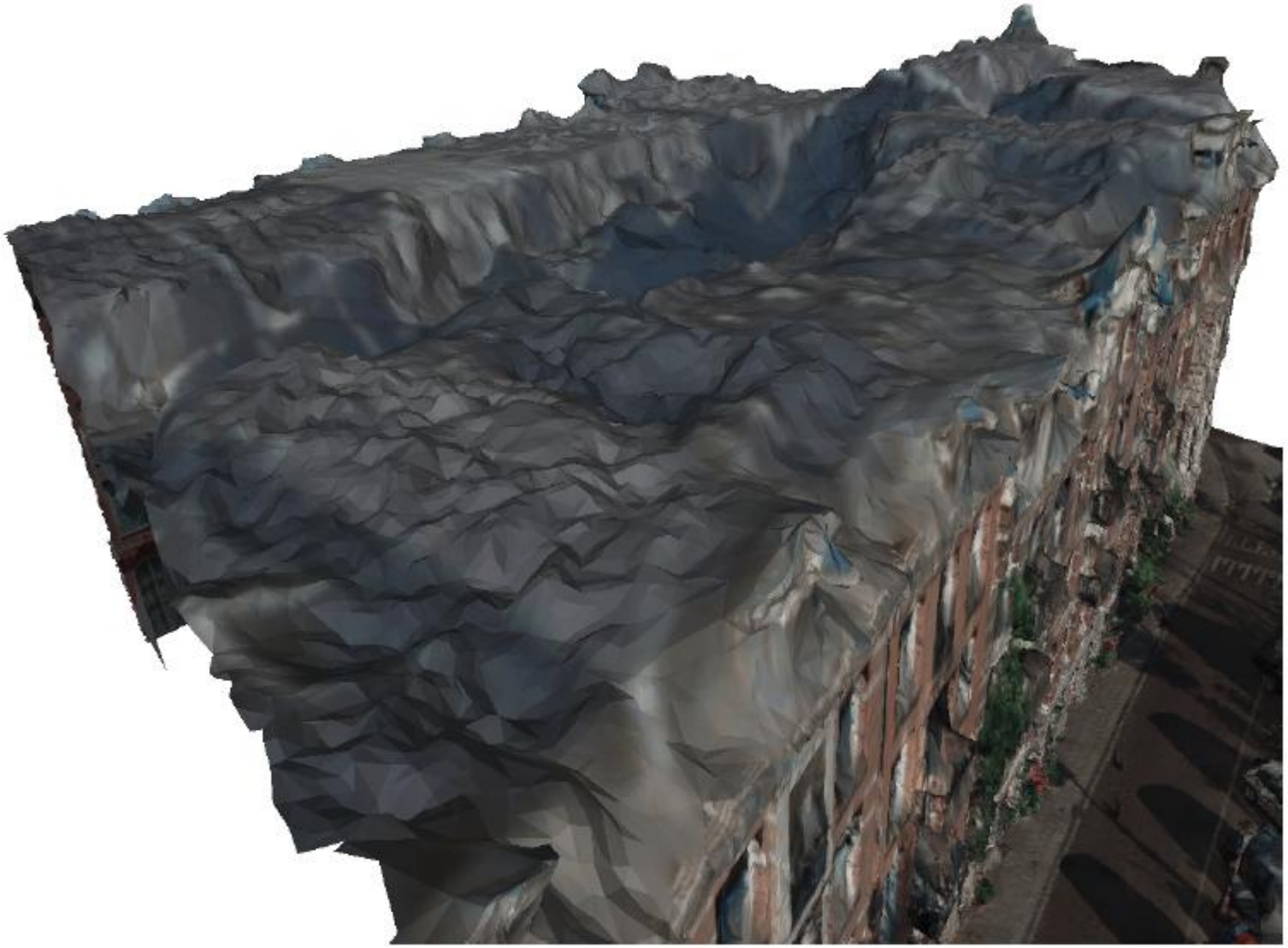


(c) Reconstructed 3D geometry



(d) Textured 3D geometry

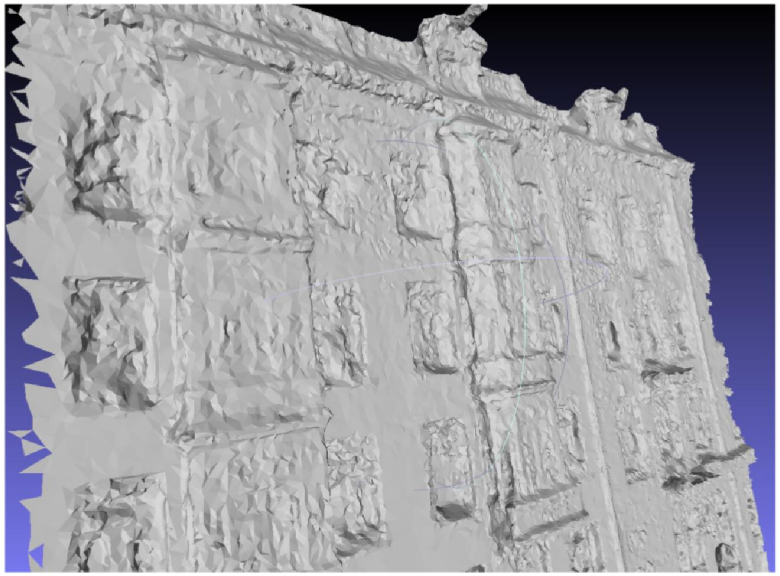
Example of a multi view stereo pipeline from (a) to (d) [Furukawa et al.,2015]



Motivation

- Because of measurement error, many vertices will have deviations from the actual planes they belong to. This will lead to a 3D mesh with low quality, and redundancy.
- By straightening and simplifying the mesh, 3D model can be improved, the city model will be cleaner.

(a) Test data with texture

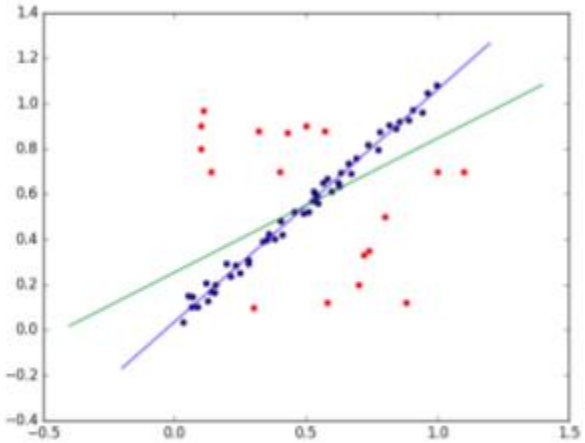


(b) Test data without texture

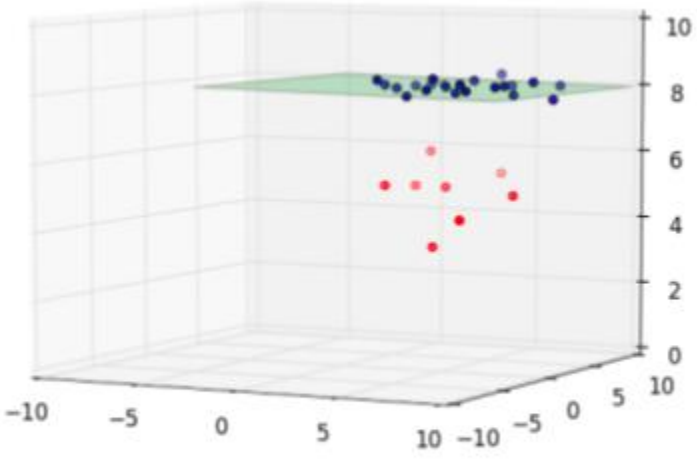
Research questions

- Can RANSAC algorithm based method yield similar or better result than existing approaches for straightening multi view stereo mesh?
- Subquestions:
- 1. What methods are currently used? What are the advantages and disadvantages?
- 2. How can some plane constraints be used for straightening meshes?
- 3. How can geometry/topology/texture information be used?
- 4. Is it feasible to simplify the straightened meshes regarding data storage and attach textures to simplified meshes?

What is RANSAC



(a) Blue line is fitted by RANSAC and green line is fitted by least squares



(b) Plane fitting by RANSAC

- Random sample consensus (RANSAC) [Fischler and Bolles, 1981] is a widely used method for parameter estimation of a mathematical model

Why is RANSAC

- Datasets have low quality
- RANSAC is capable of robustly dealing with data containing more than 50% of outliers [Schnabel et al., 2007]

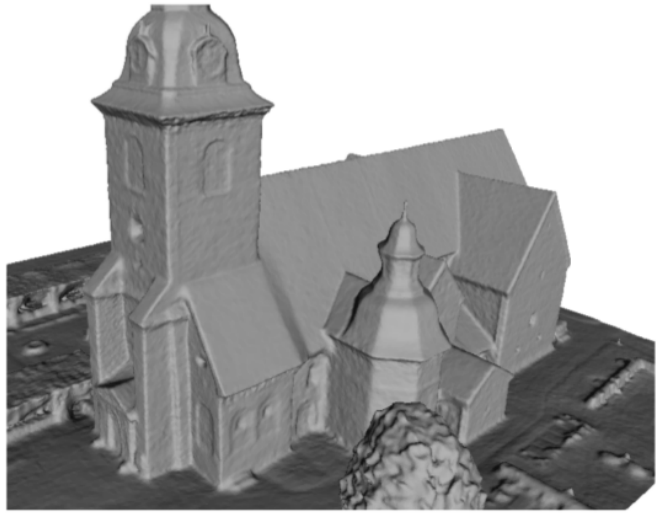
Related work

Most related work:

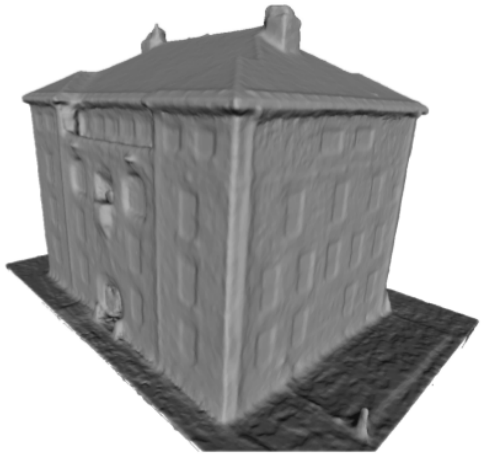
1. Get planar areas according to some features: curvature, normal, texture, elevation, planarity, horizontality etc.
2. Plane fitting on these planar areas.
3. Project points to planes.

Related work

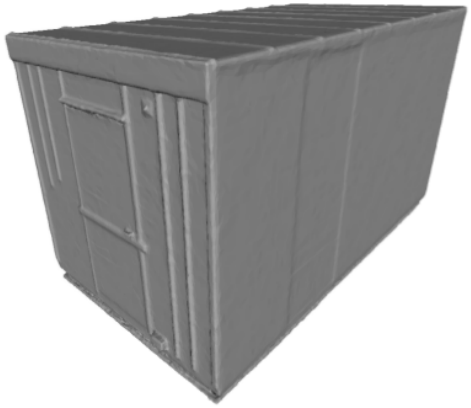
Input data:
- Smoother
- Much less noise
- Good quality



(a) Vreta Church data set



(b) Vasallen data set

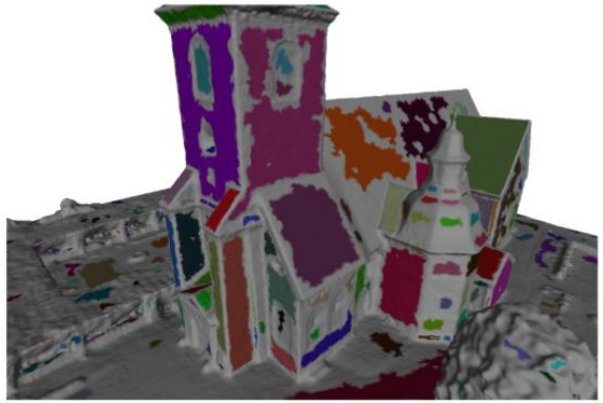


(c) Container data set

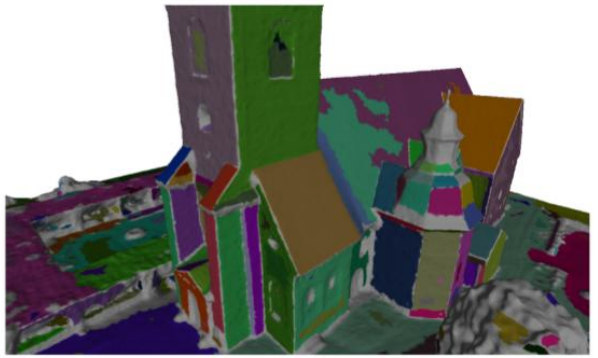
By Mickael Jonsson
(2016, Linkoping, Sweden)

Related work

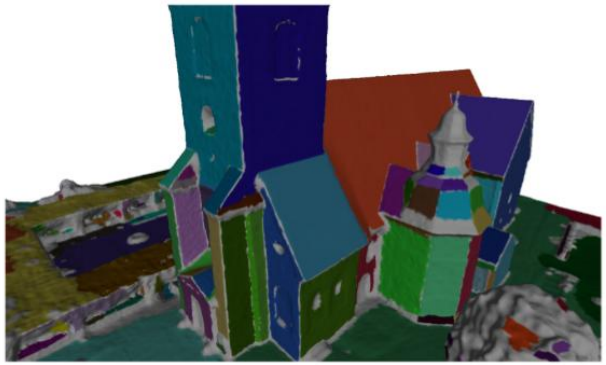
Segment the mesh to find planar areas based on curvature



(a) Initial planes extracted through hierarchical curvature segmentation



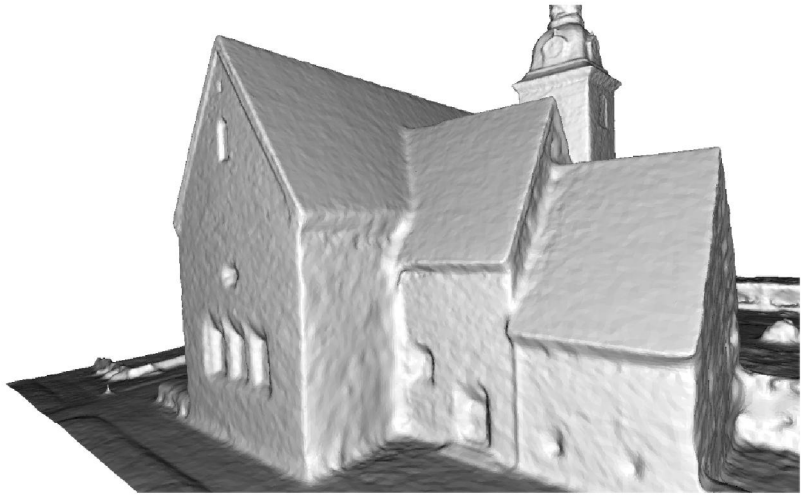
(b) Plane growing



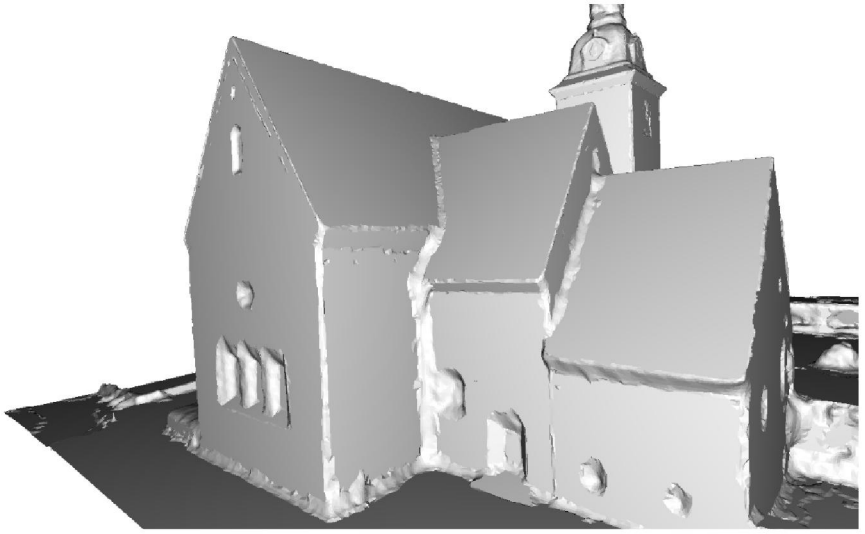
(c) Planar region merged

By Mickael Jonsson
(2016, Linkoping, Sweden)

Related work



(a) Original mesh

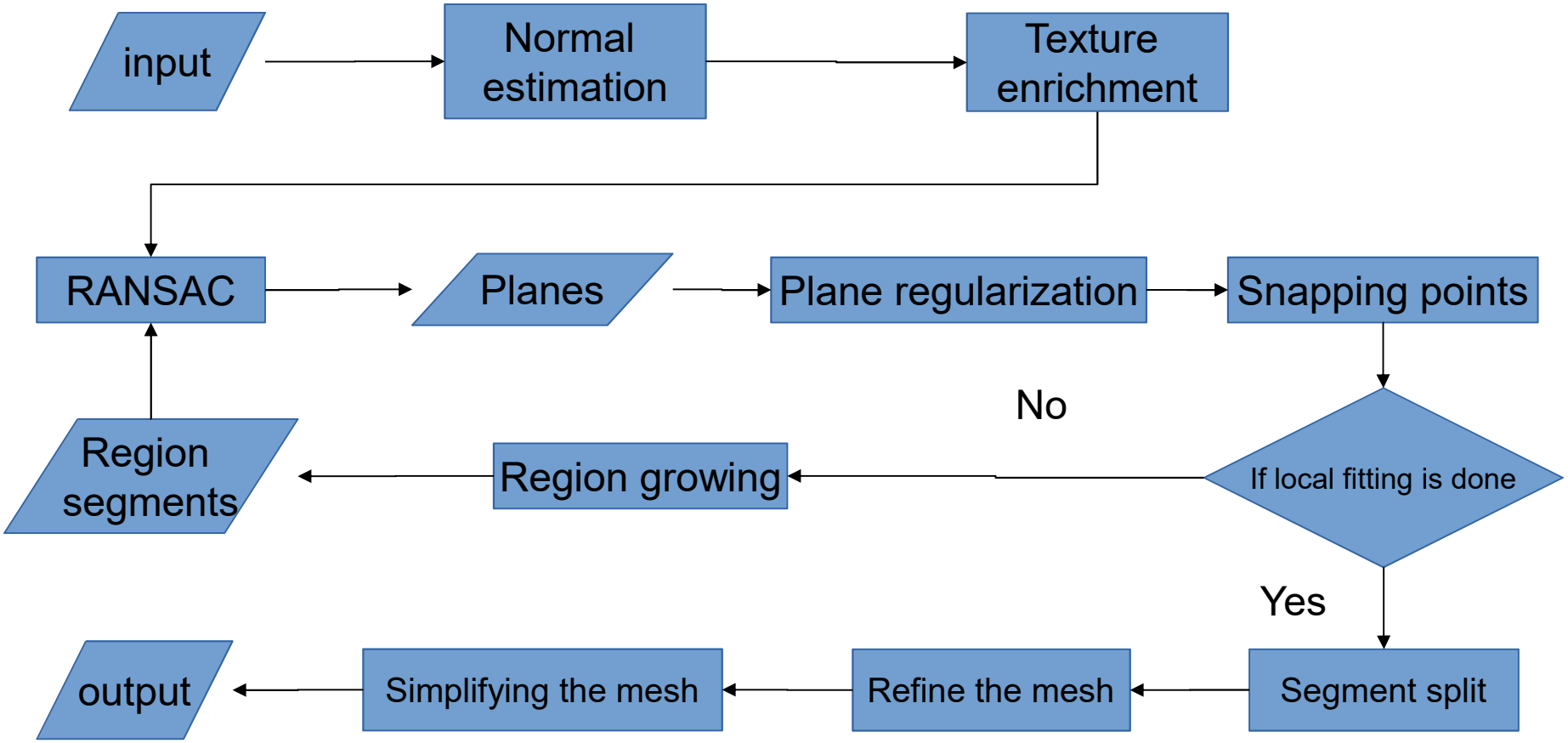


(b) Flattened mesh

Tools and software

- C++
- CGAL (the Computational Geometry Algorithms Library): Polygon Mesh Processing, Point Set Shape Detection, 2D Triangulation
- CImg: Texture information enrichment
- Meshlab for visualization.

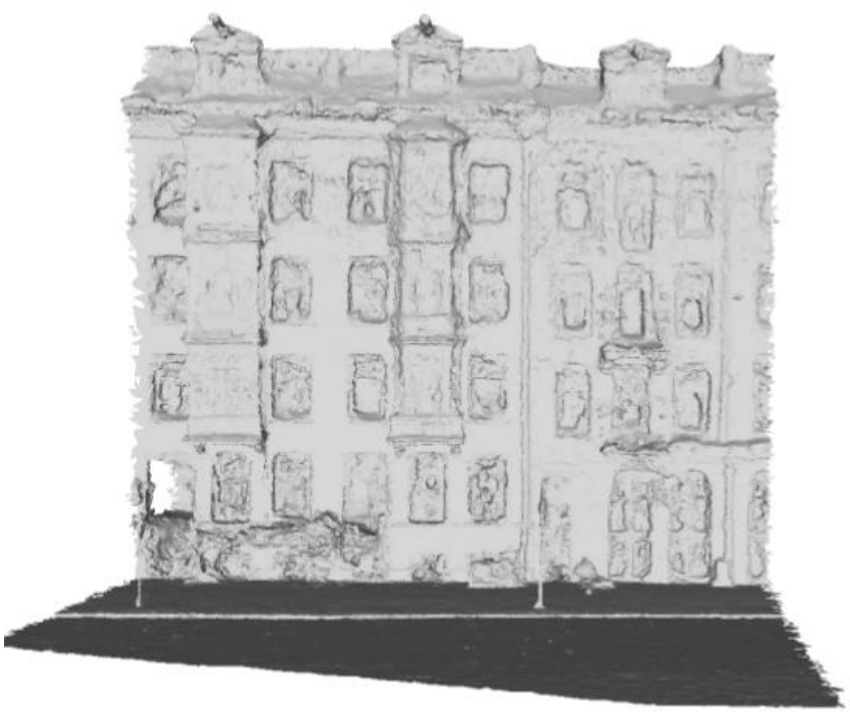
Methodology



Test data 1

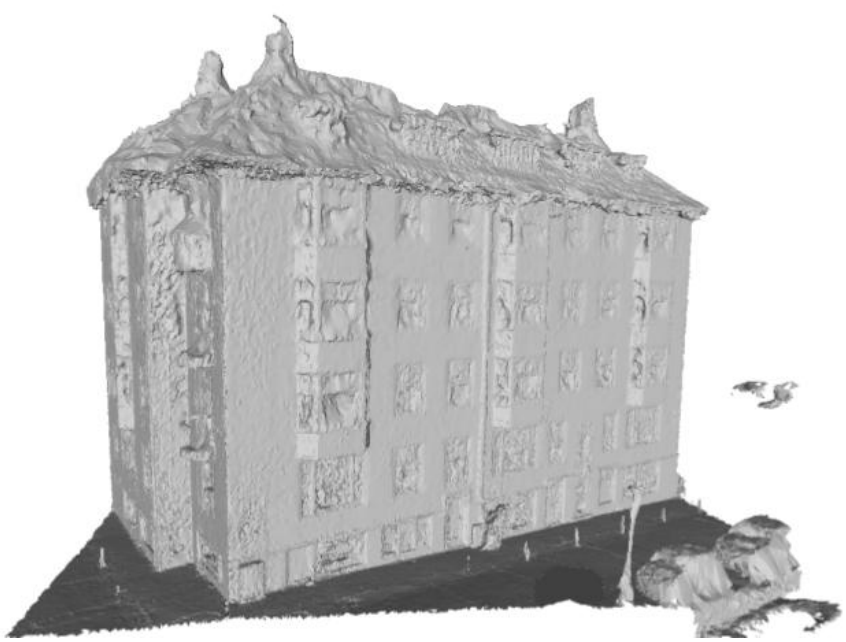
1. facade of two gable roof buildings, one flat roof building
2. part of ground

Test data 2



1. facade of two flat roof buildings
2. vegetation
3. ground

Test data 3

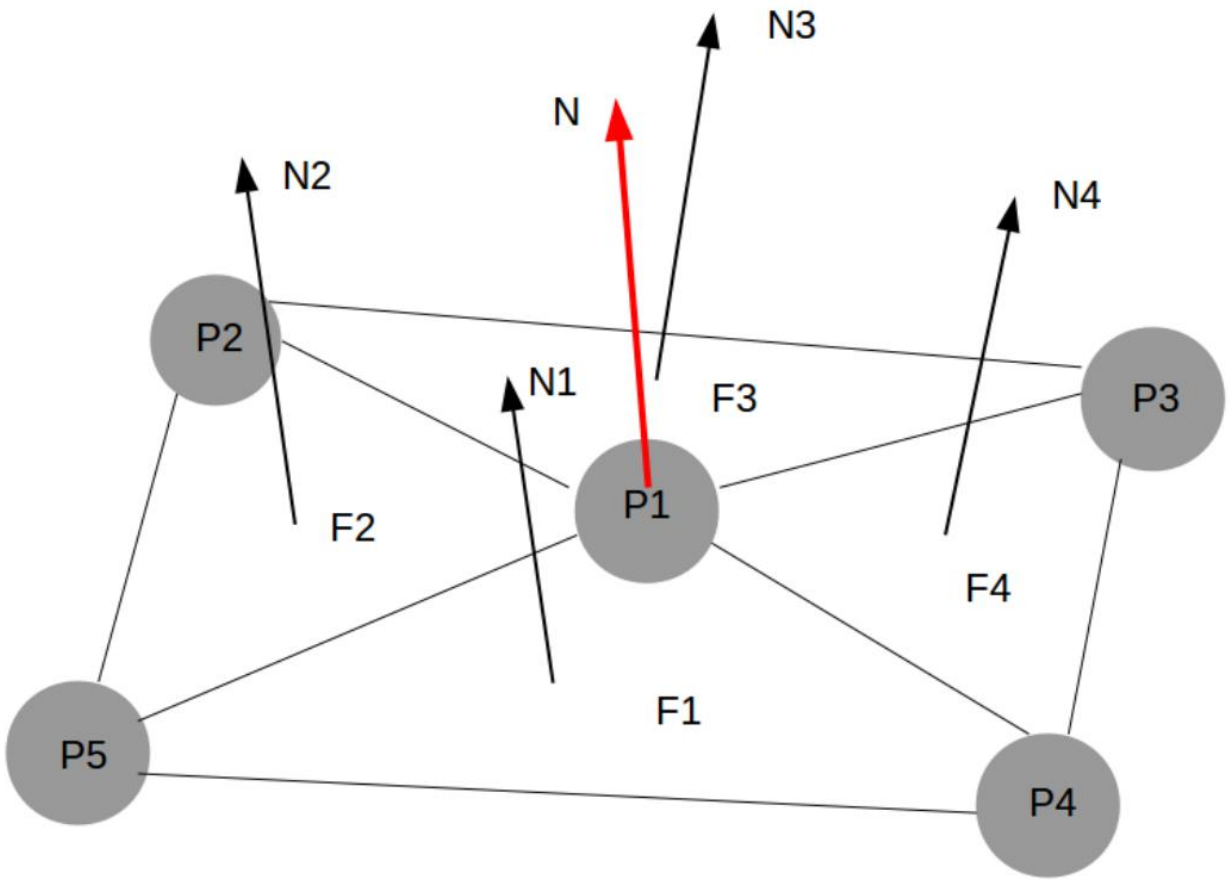


1. One complete building
2. part of ground
3. Cars

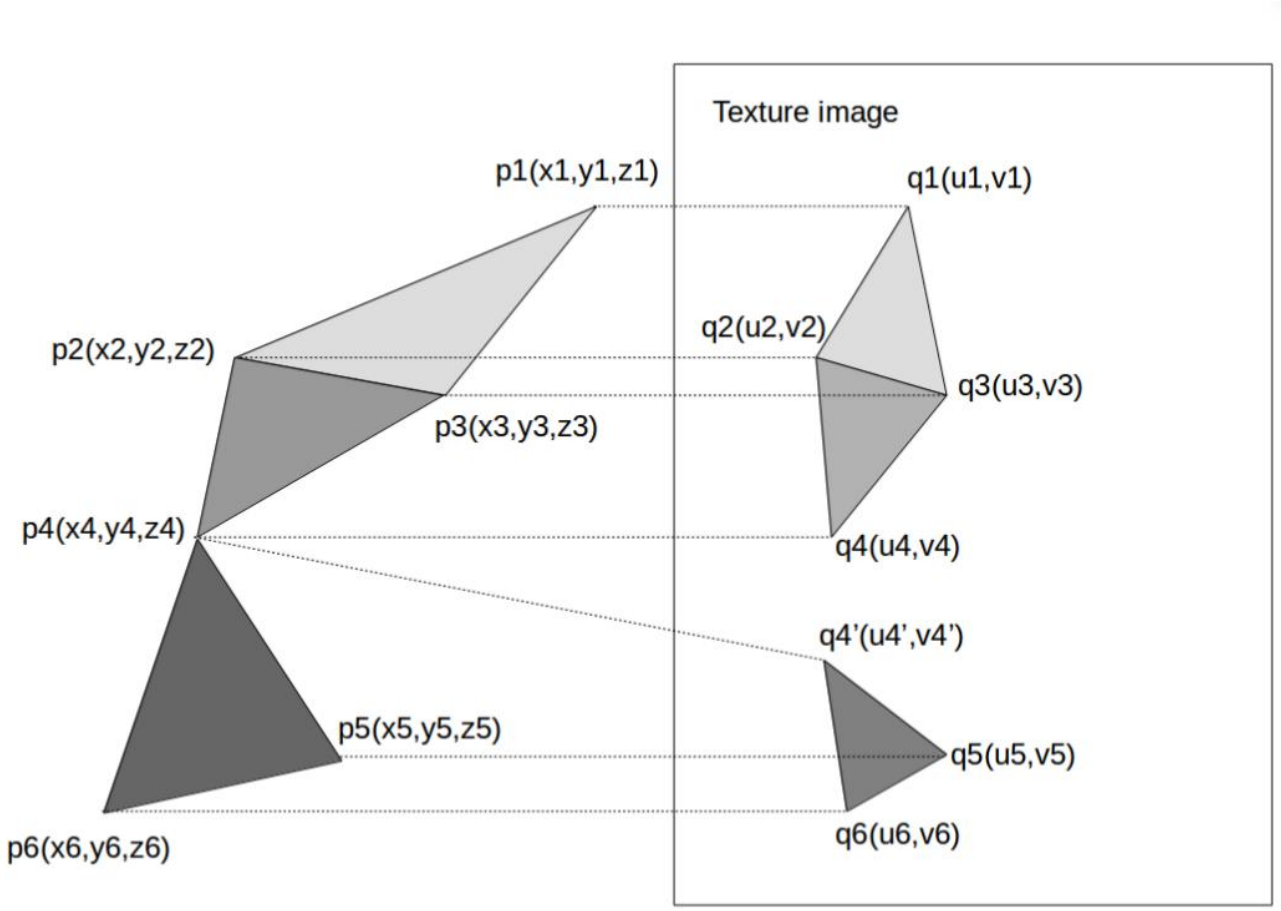
Test data 4

1. multiple buildings on the two sides of the road
2. part of ground
3. Cars, bench etc.

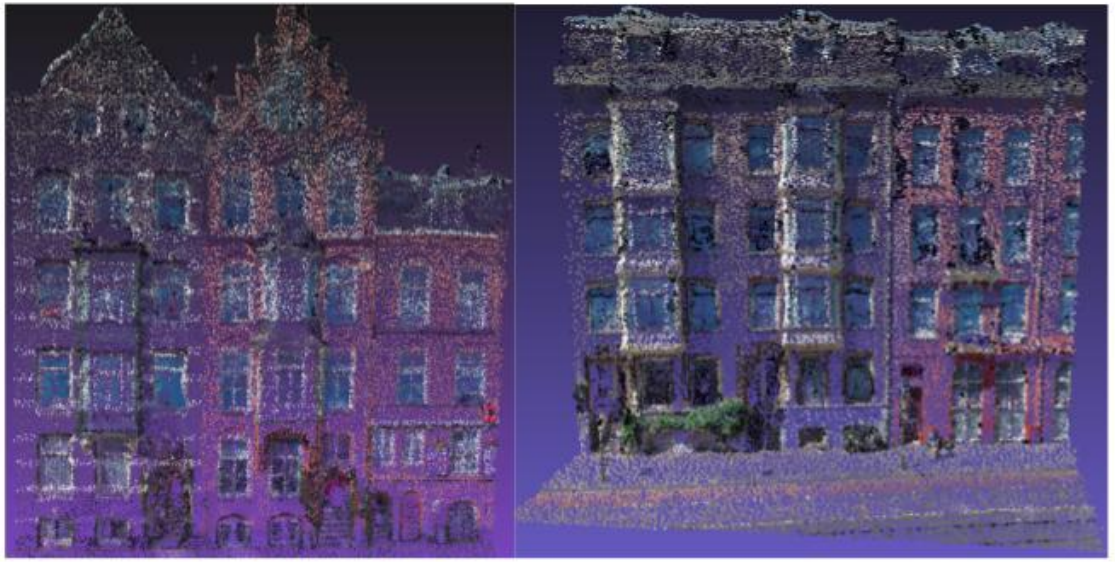
Normal estimation



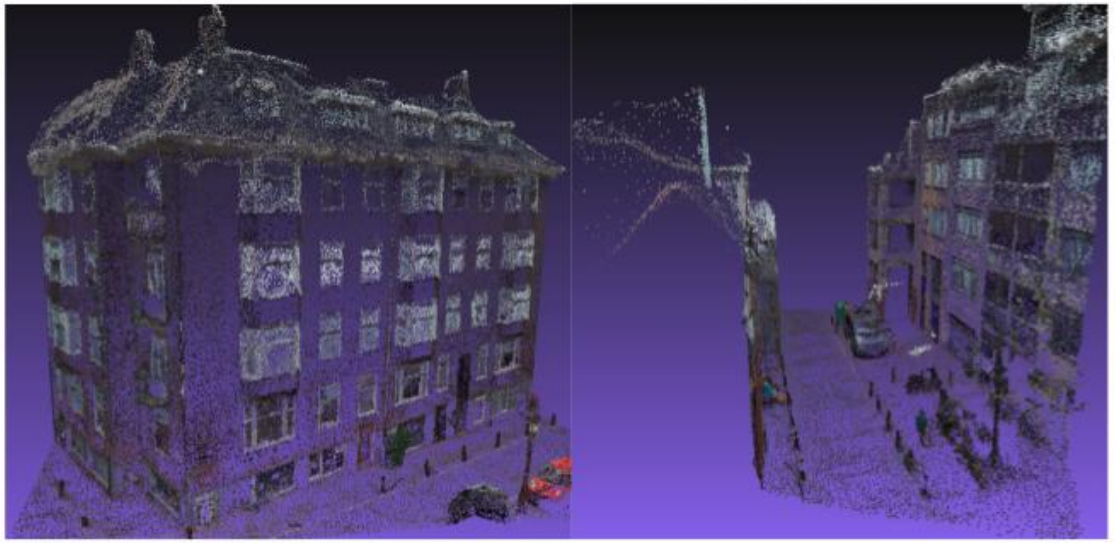
Texture information enrichment



Texture information enrichment

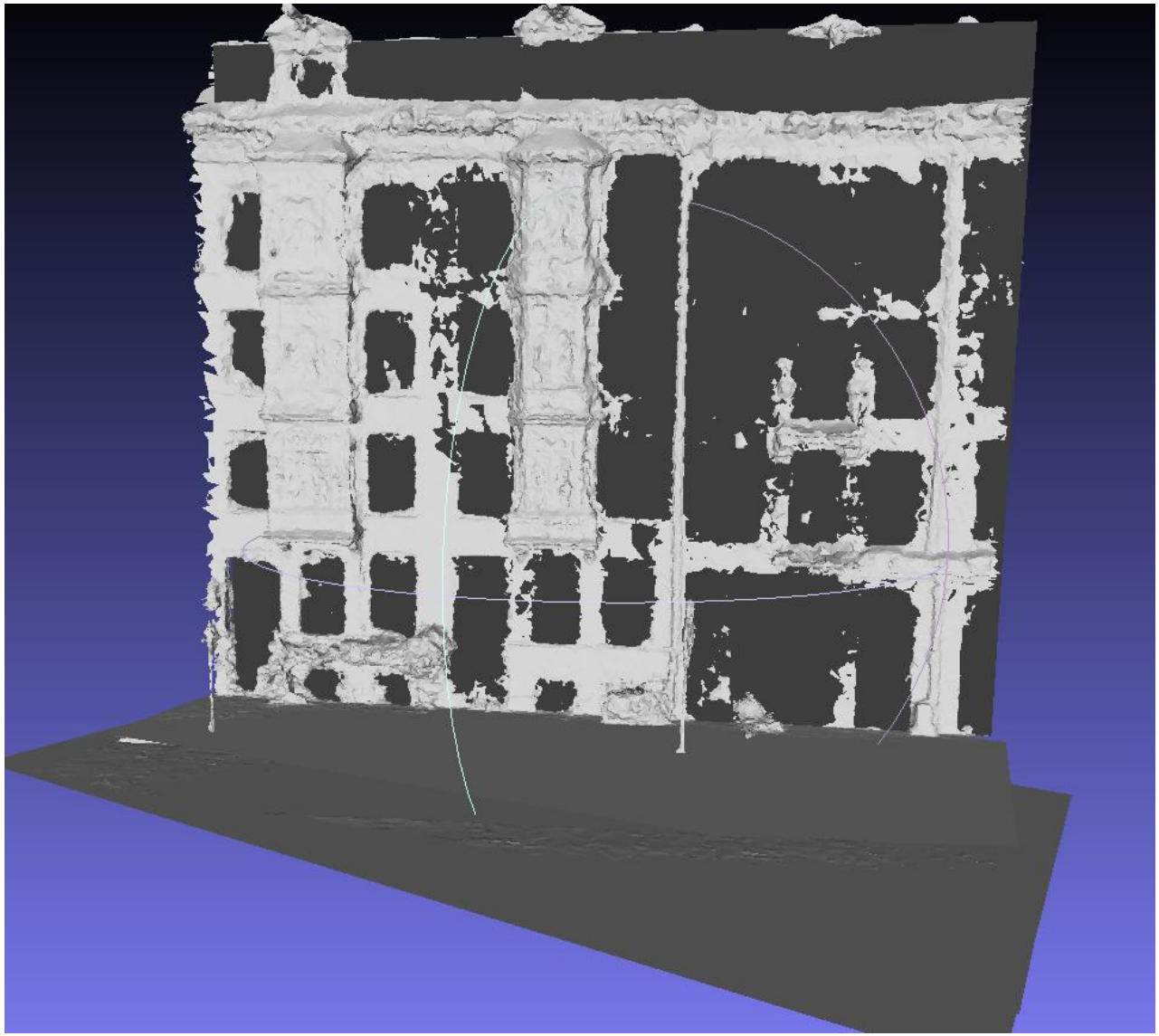


(a) Test data I (b) Test data II



Global fitting

Get main planes:
1. Facade
2. Ground
3. Roof



Plane regularization

- Planes that are near parallel are made parallel: normal vectors of planes that form angles smaller than a user-defined threshold are made equal.

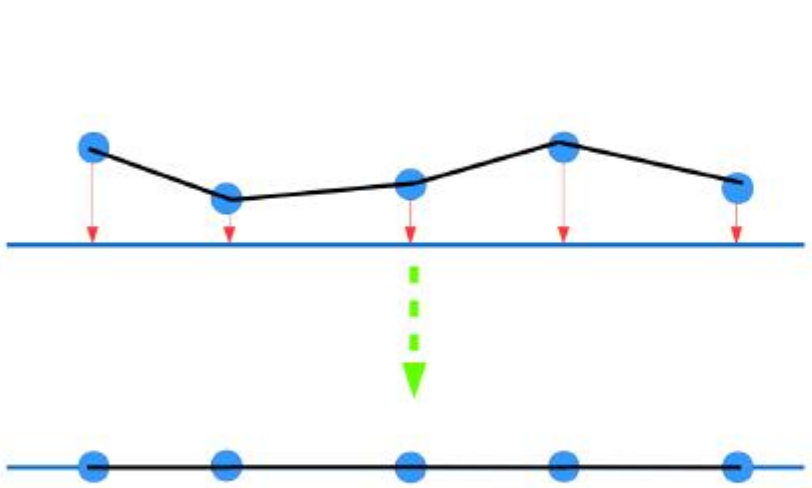
- Planes that are near coplanar are made coplanar.

- Planes that are near orthogonal are made exactly orthogonal.

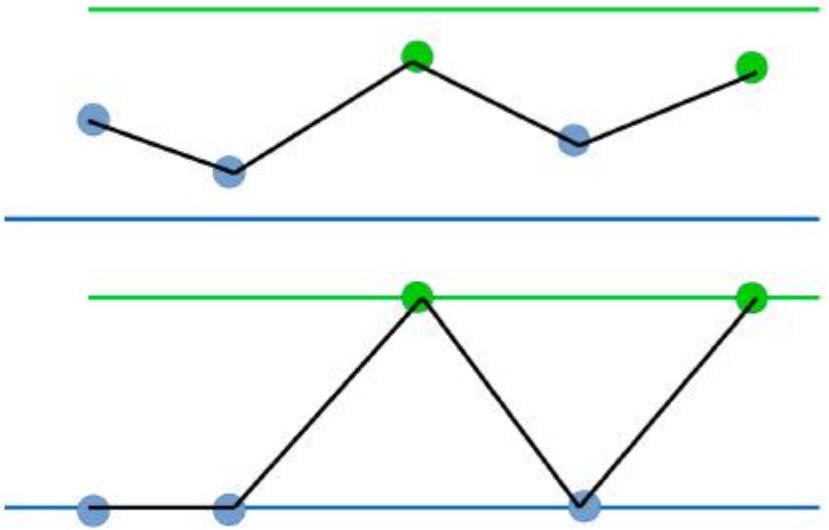
- Planes that are near symmetrical with respect to a user-defined axis are made symmetrical.

From Verdie et al. [2015]

Near coplanar problem



(a) Point snapping

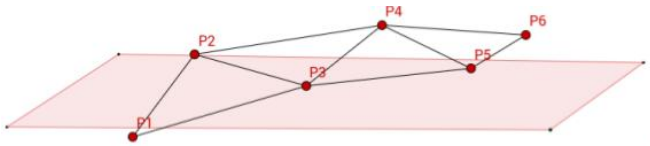


(b) Near coplanar planes problem

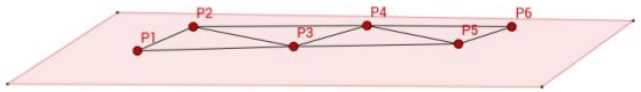
Snapping

1. Snapping vertices to planes

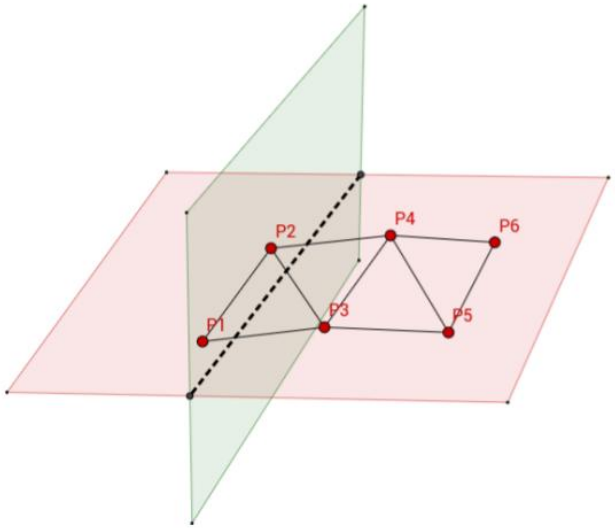
2. Snapping vertices to intersection lines



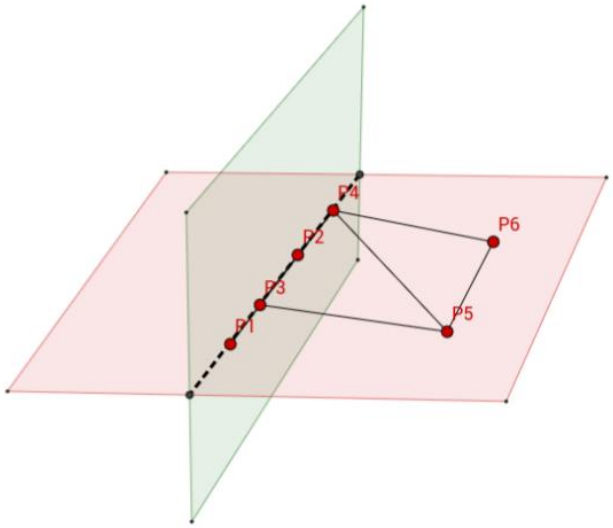
(a) Original mesh and fitted planes



(b) Snapping vertices to the corresponding plane

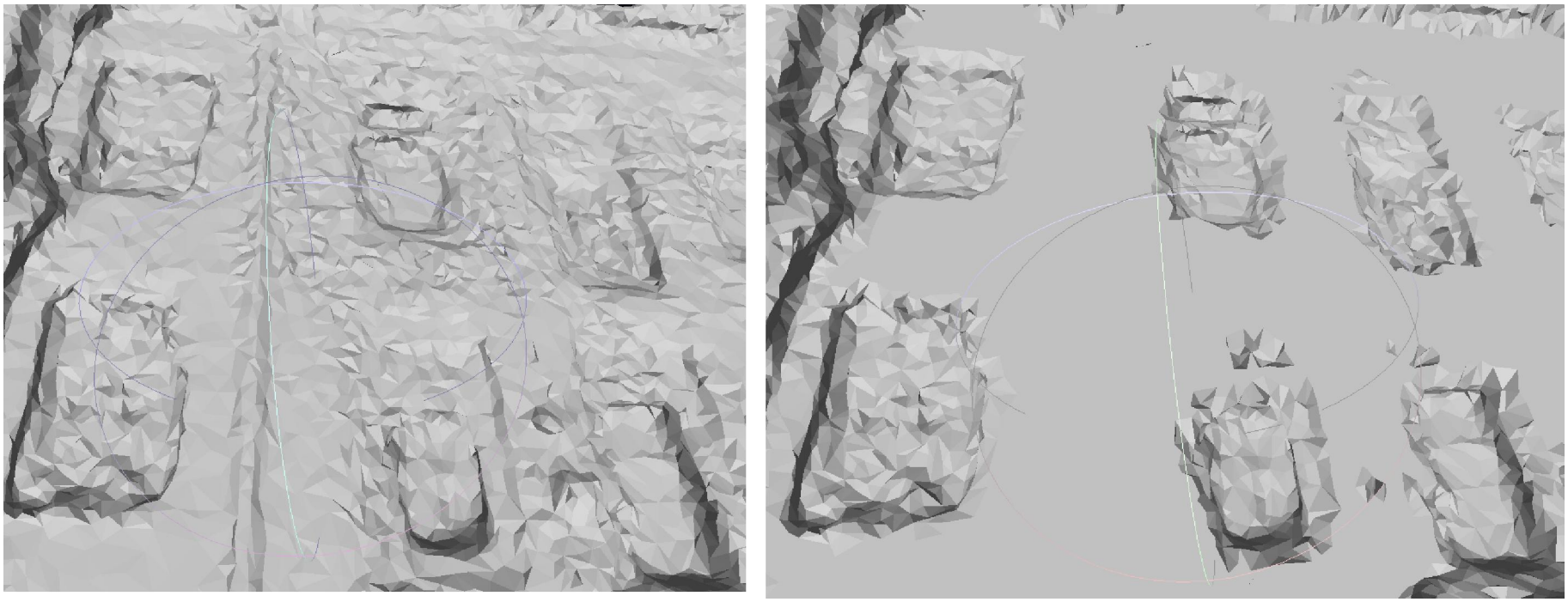


(c) Original mesh and two fitted planes

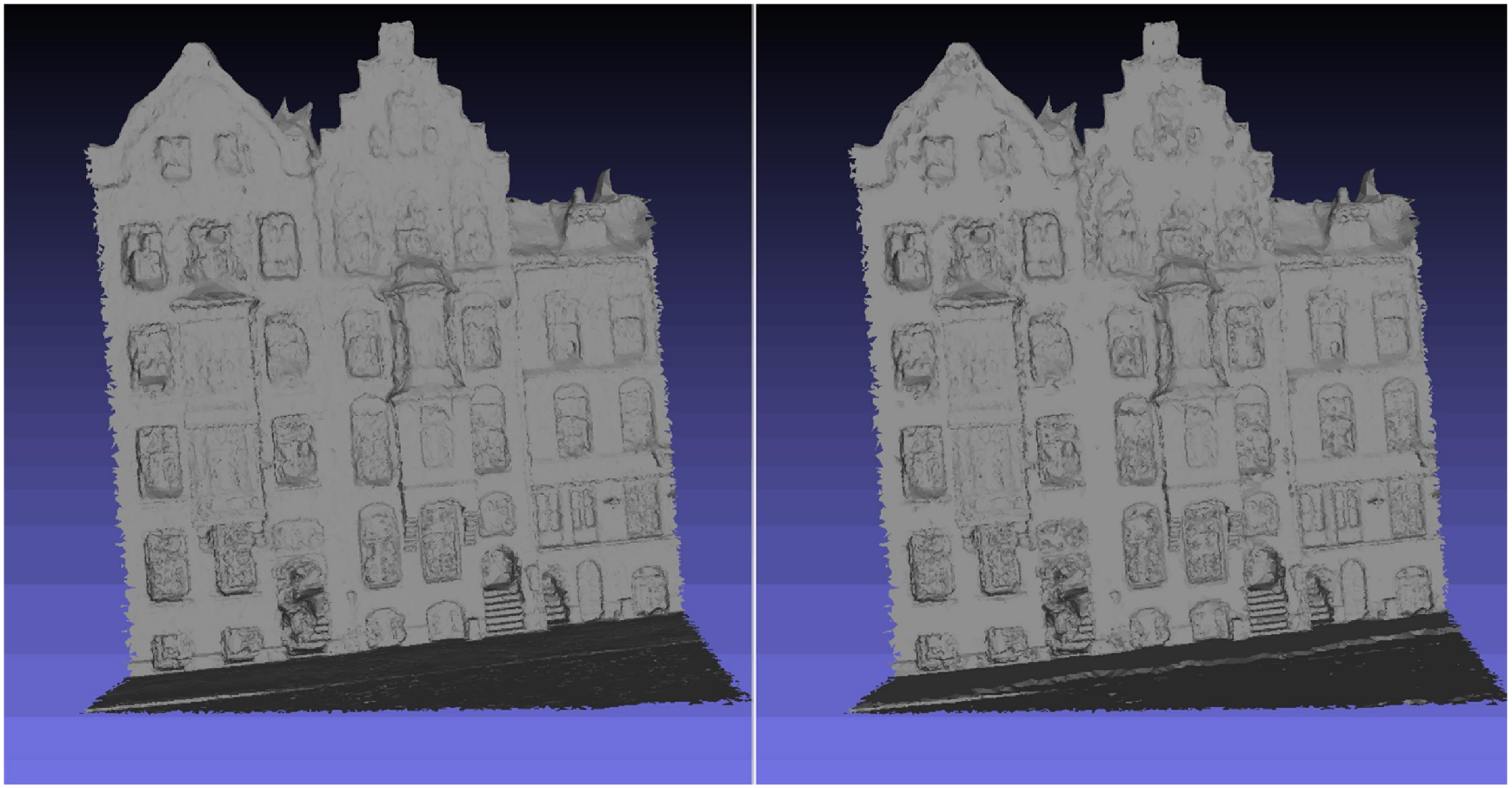


(d) Snapping vertices to the intersection line

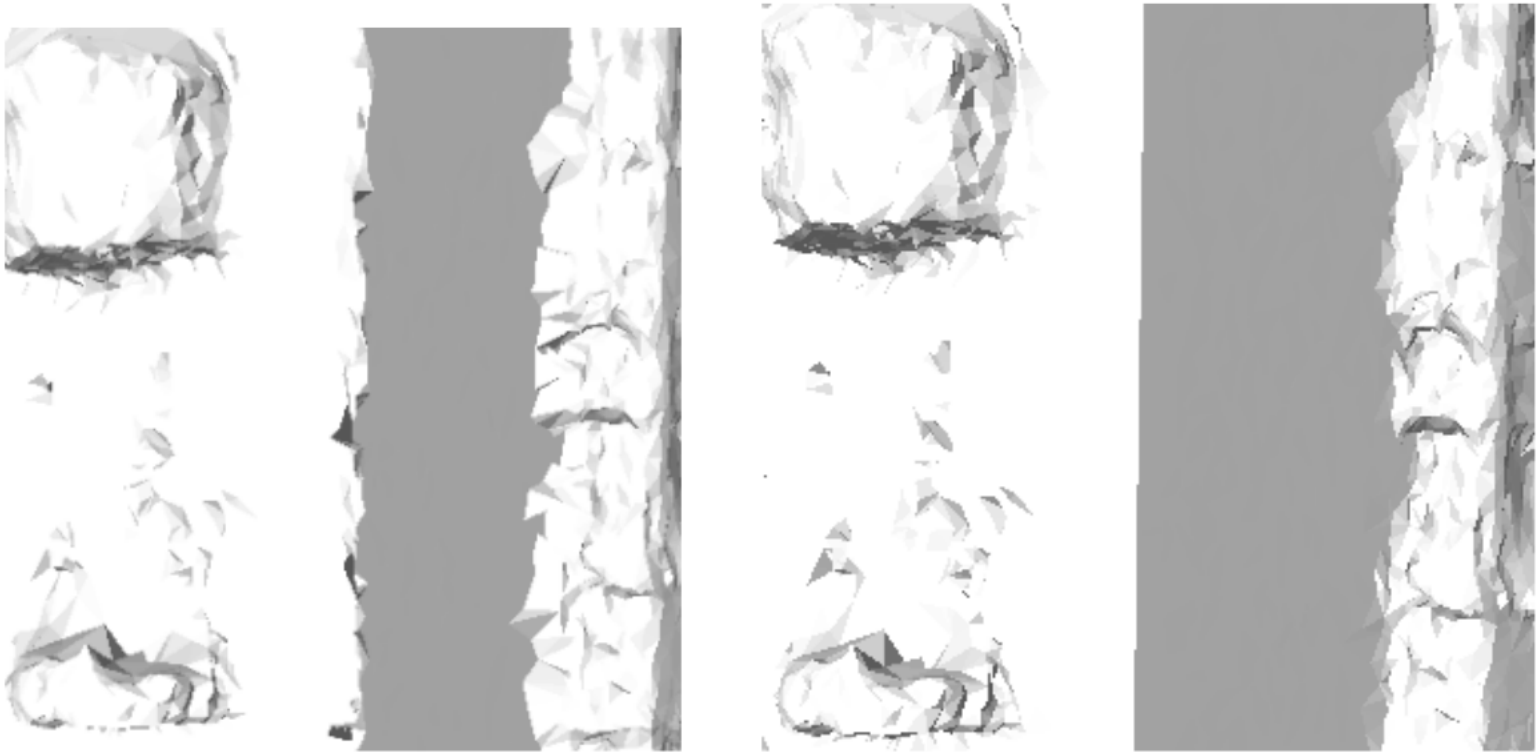
Snapping vertices to planes



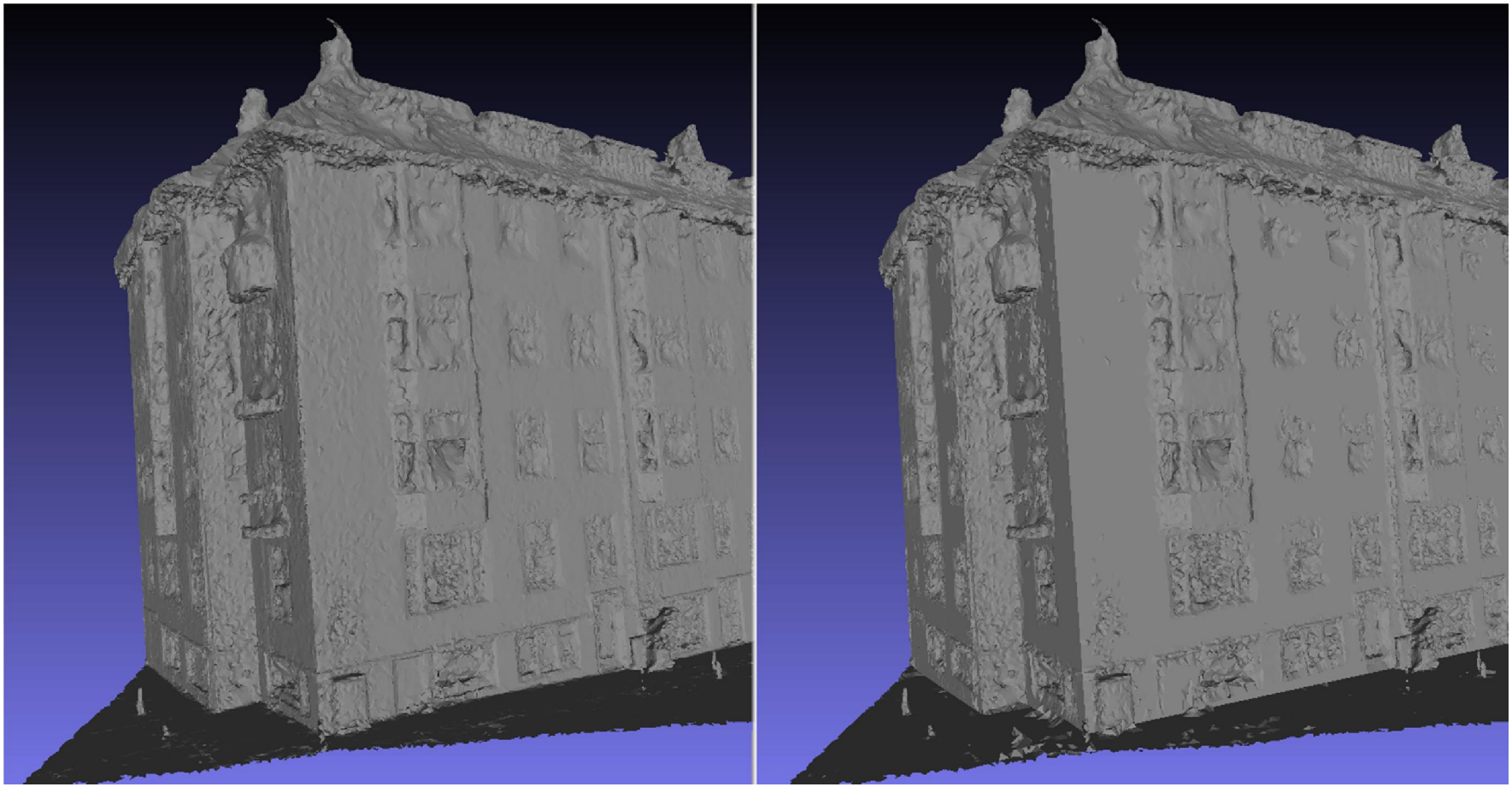
Snapping vertices to planes



Snapping vertices to intersection lines

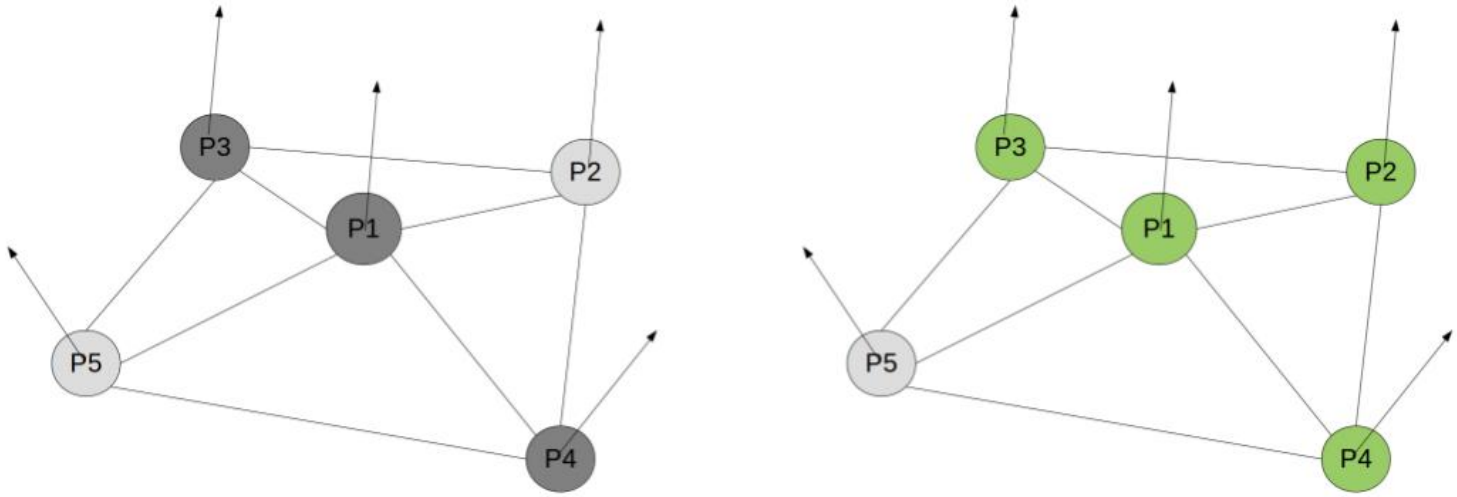


Snapping vertices to intersection lines



Mesh segmentation

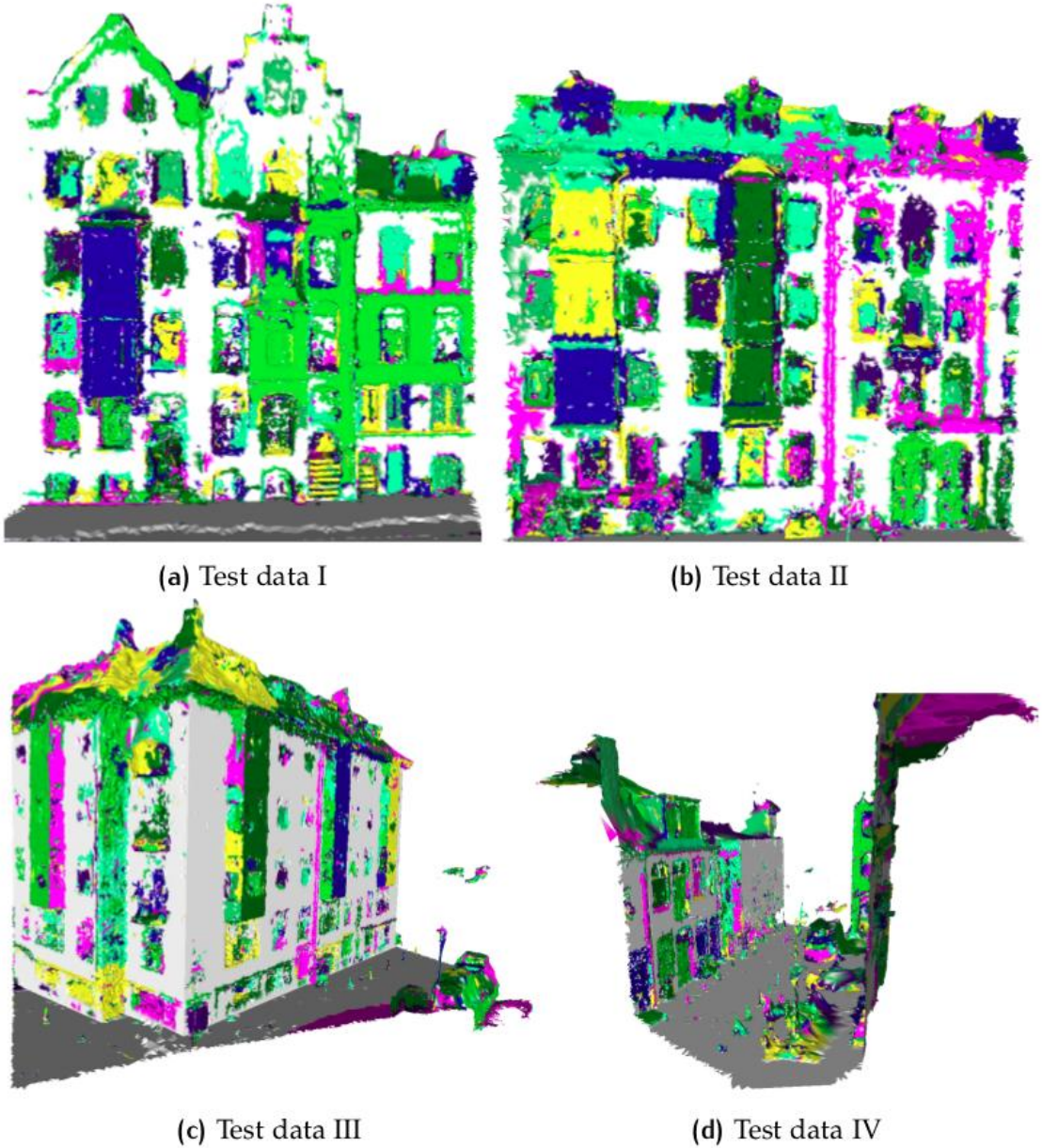
Method- Region growing



Rules:

1. If the adjacent vertex has similar normal, it will be included in the same segment. (Dot product)
2. If the adjacent vertex does not have similar normal, but has similar color, it will be included in the same segment. (Manhattan distance)

Segmentation Result



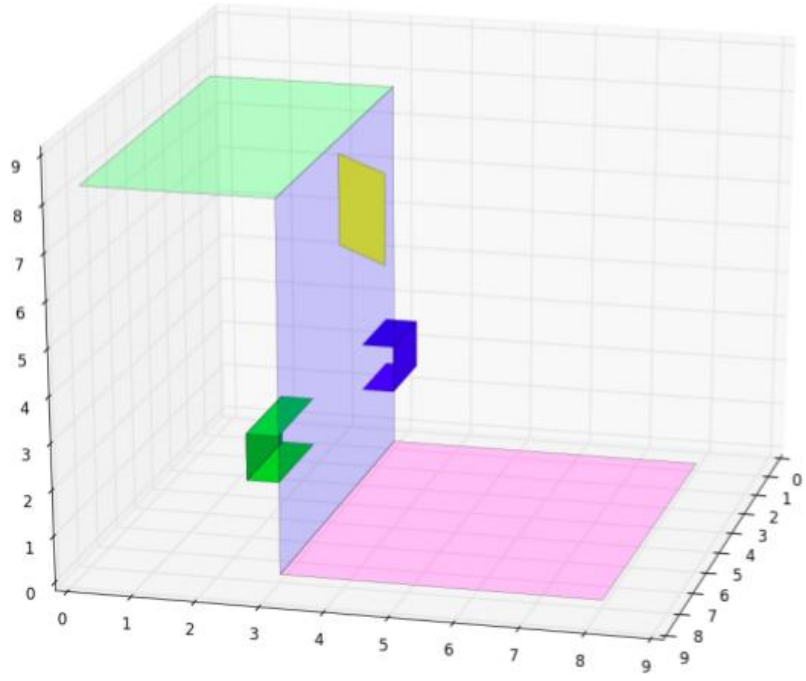
(a) Test data I

(b) Test data II

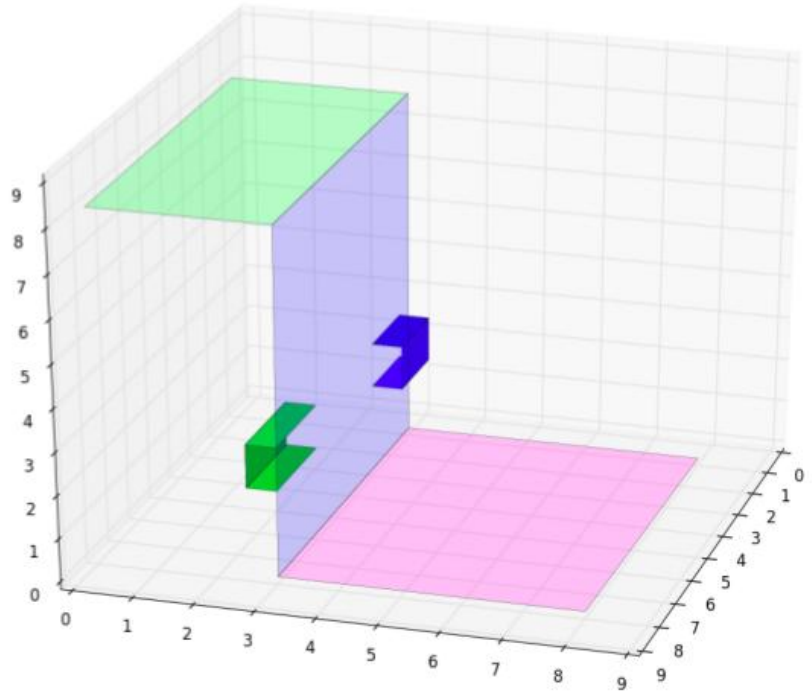
(c) Test data III

(d) Test data IV

Plane constraint
In local fitting

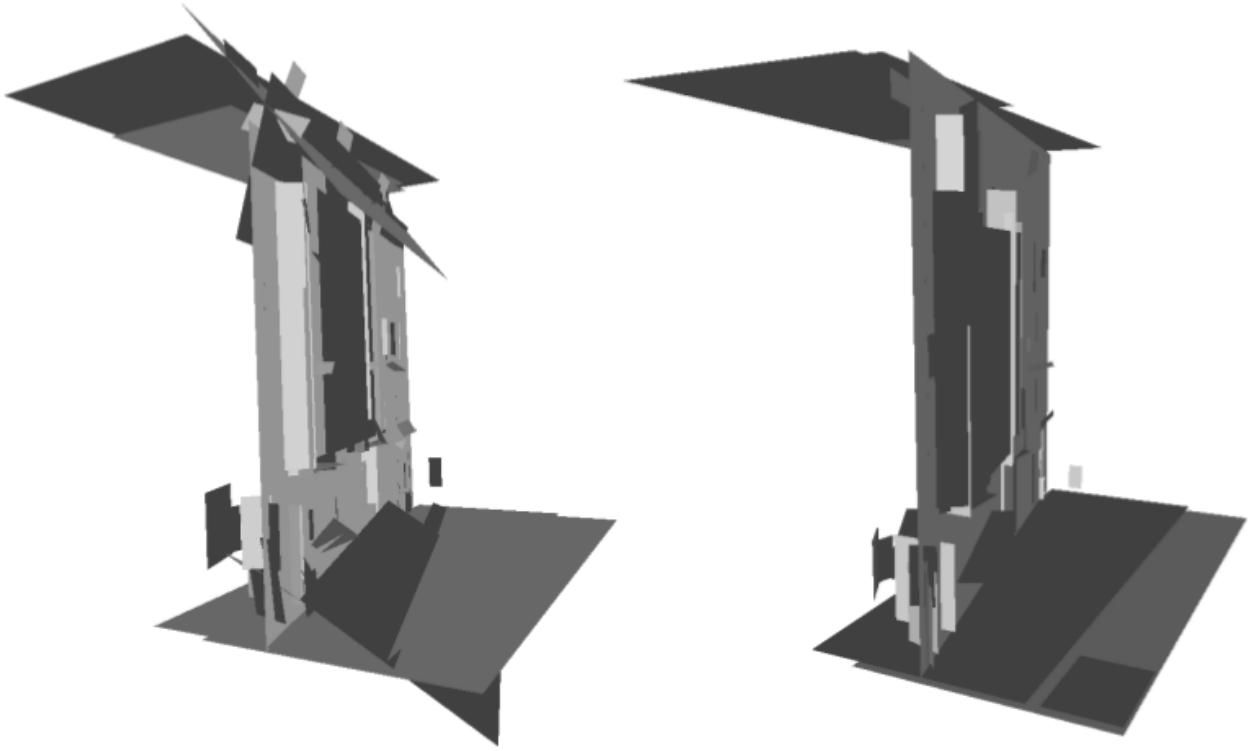


(a) Planes fitted in global fitting and local fitting



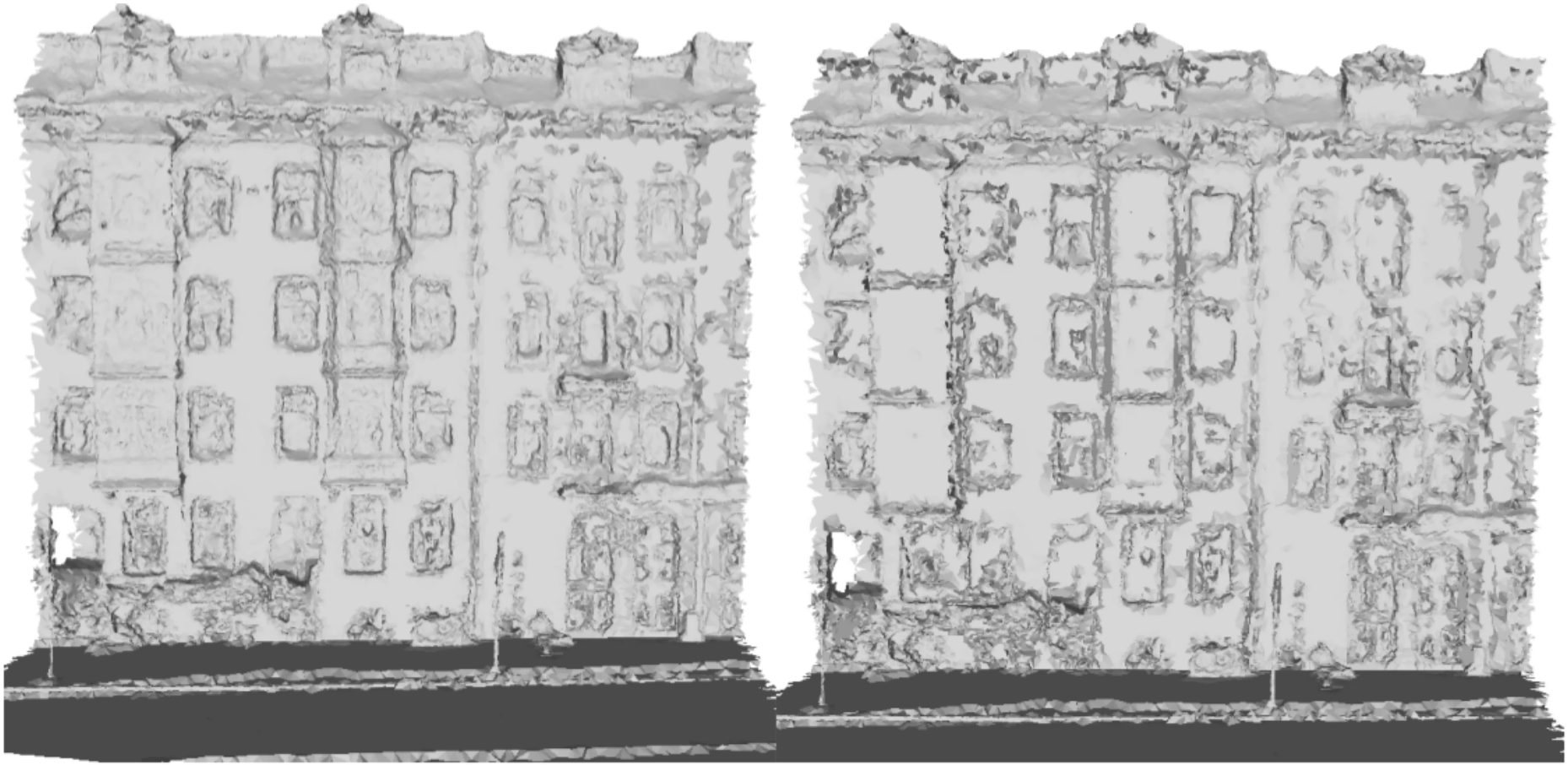
(b) Remove unregularized planes

Plane constraint
In local fitting



(a) Test data II unconstrained planes

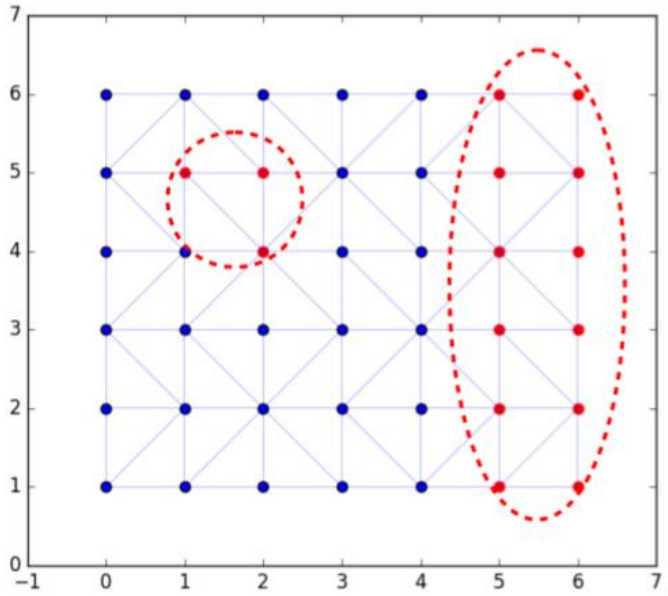
(b) Test data II constrained planes



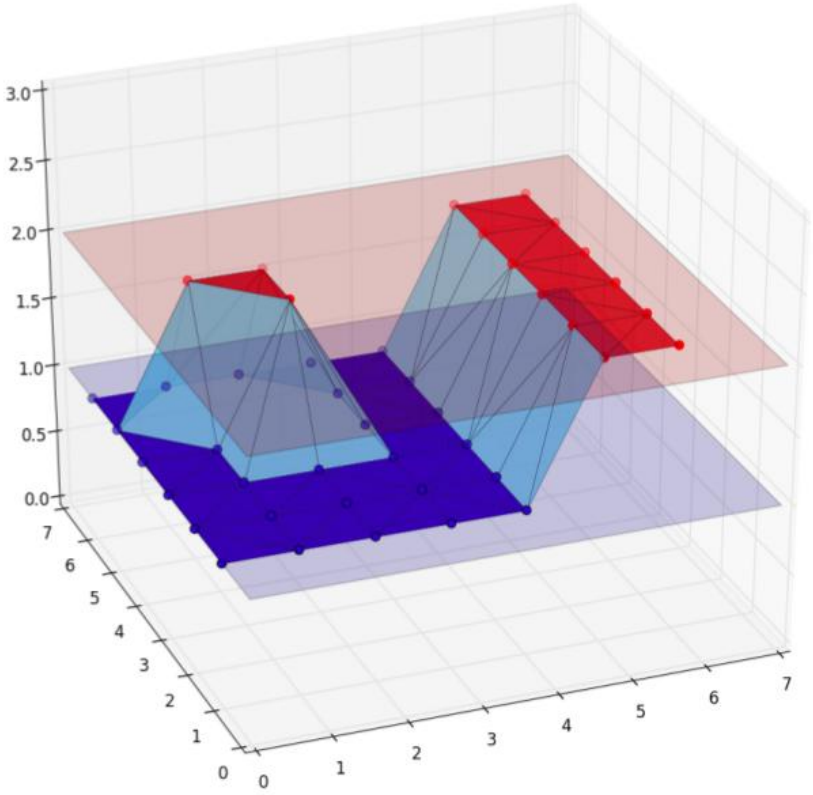
(a) Global fitting

(b) Local fitting

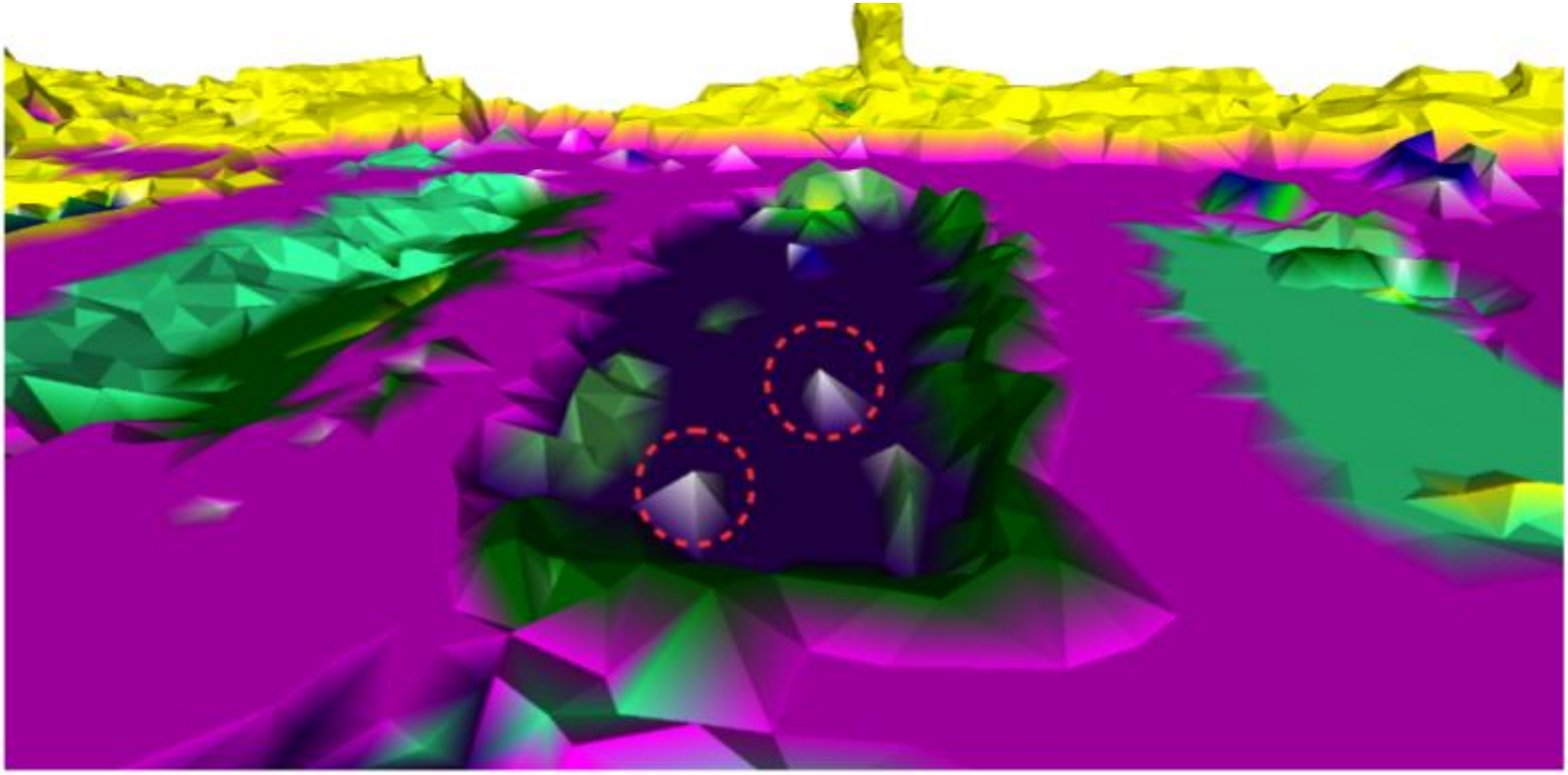
Segment split

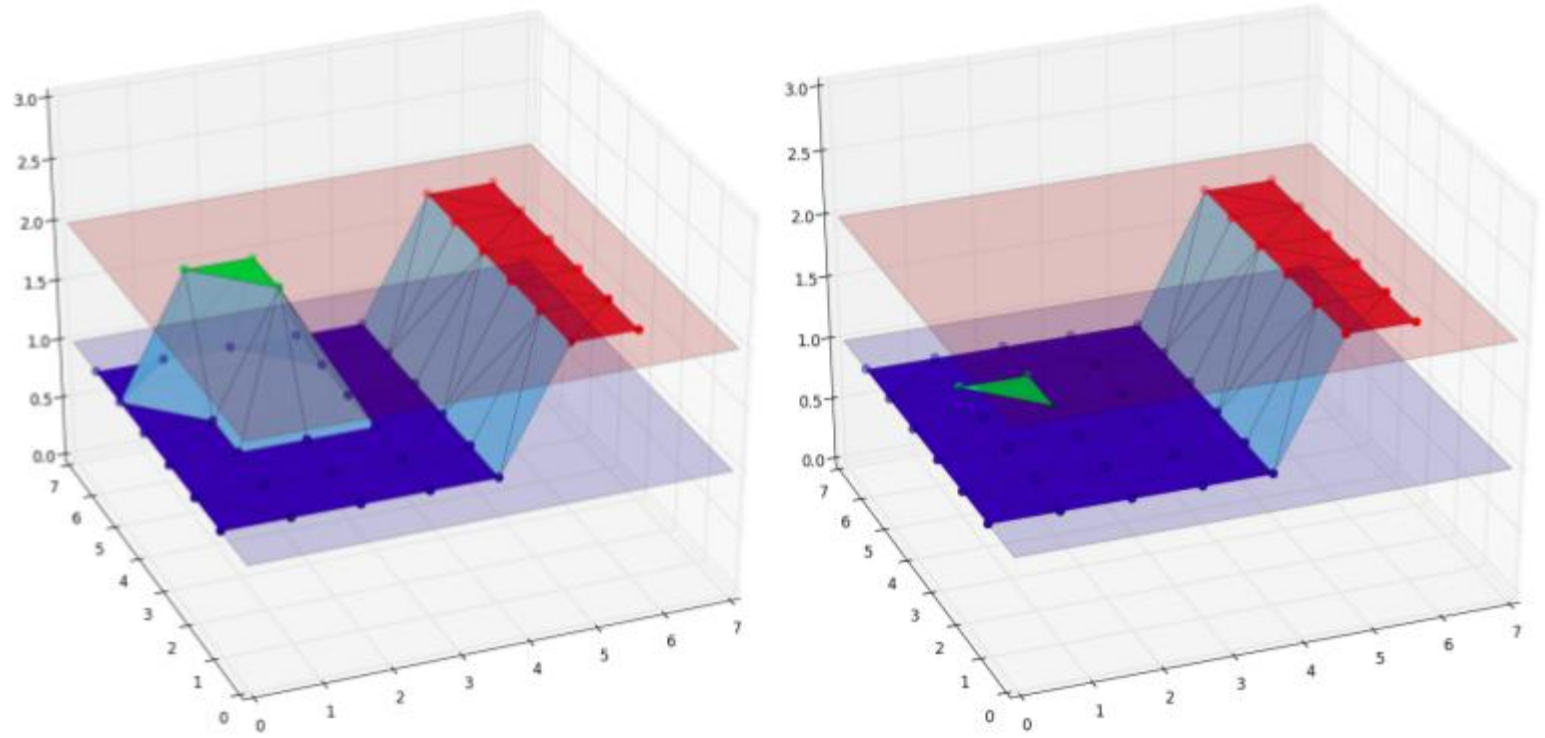


(a) Points belong to one segment are separated in 2D space

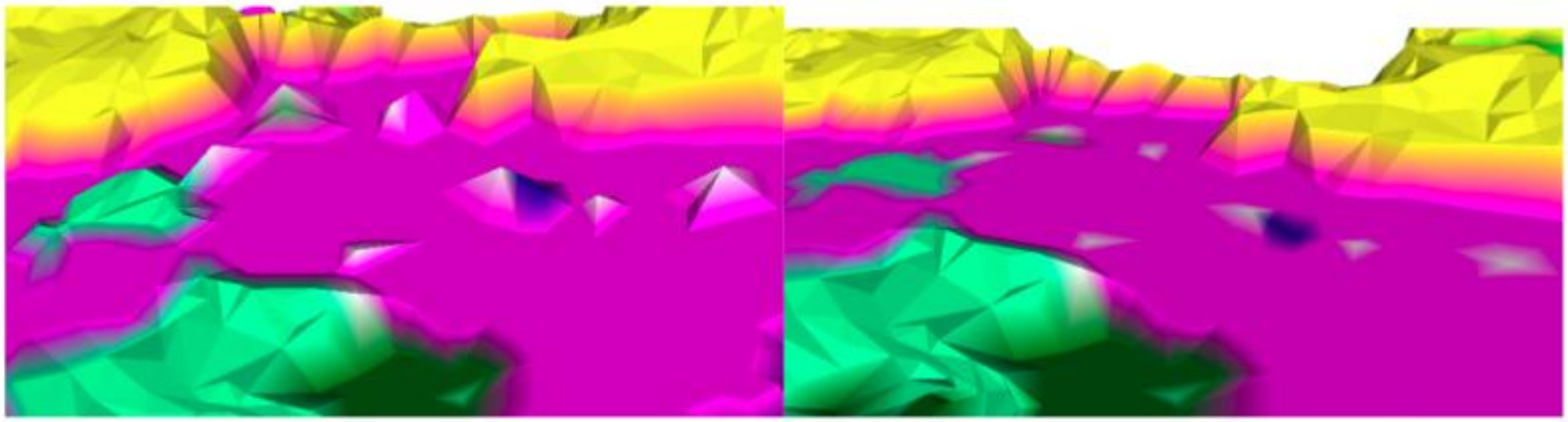


(b) Spikes problem in 3D space





(a) Split red segment into two segments **(b)** Refine the mesh by removing the spike (green and red)



(a) Spikes on facade

(b) After removing spikes

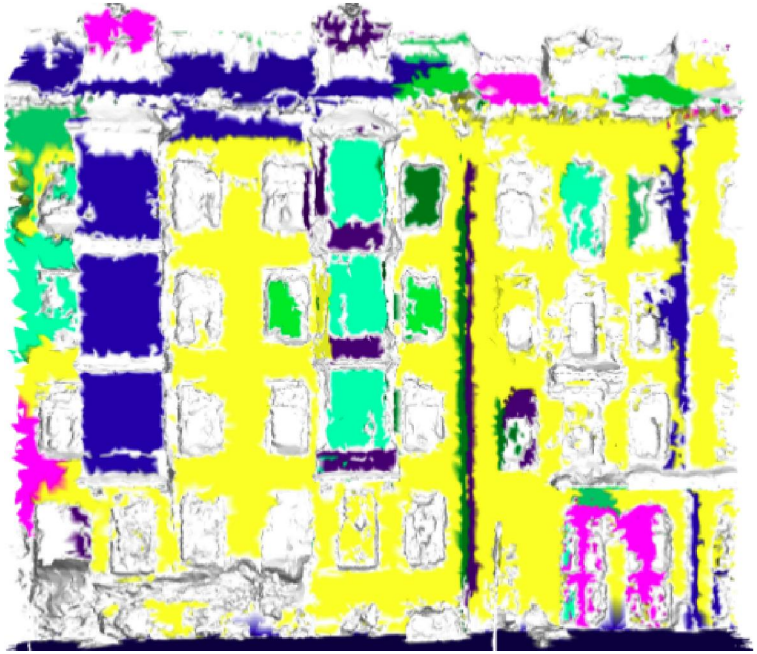
Test data 1

(a) Original mesh (b) Result (c)Result with color

Test data 2

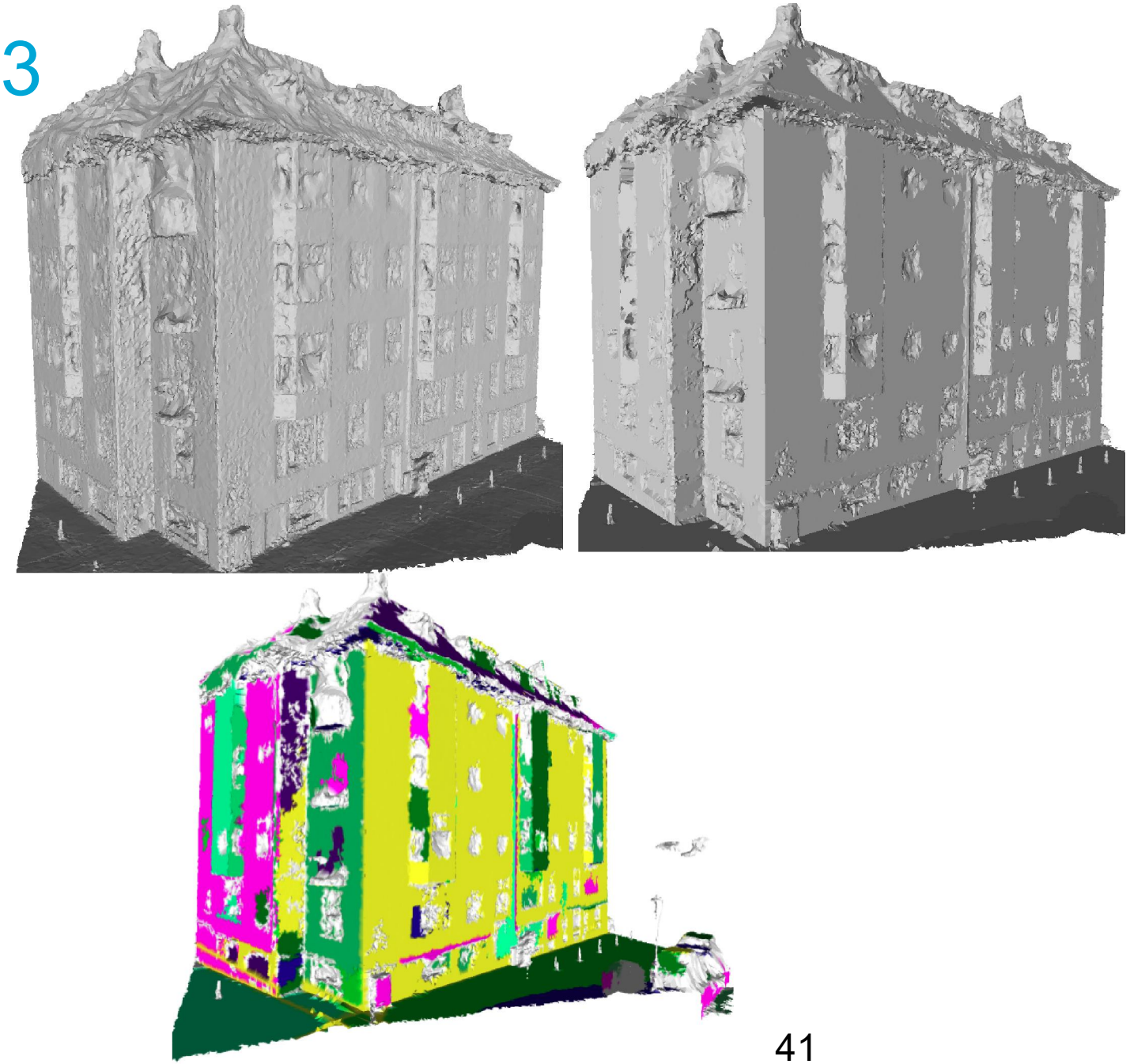
(a) Original mesh

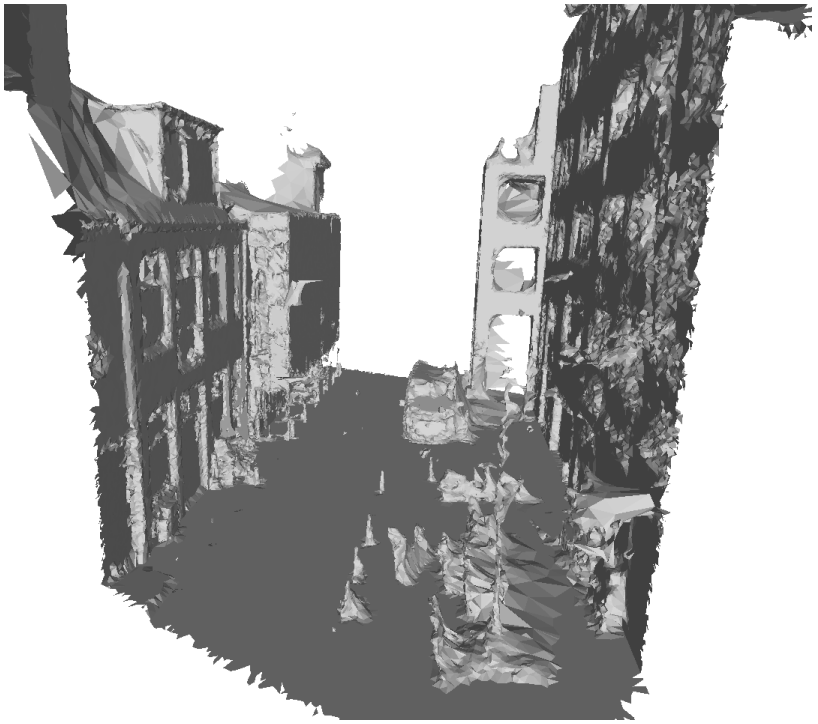
(b) Result

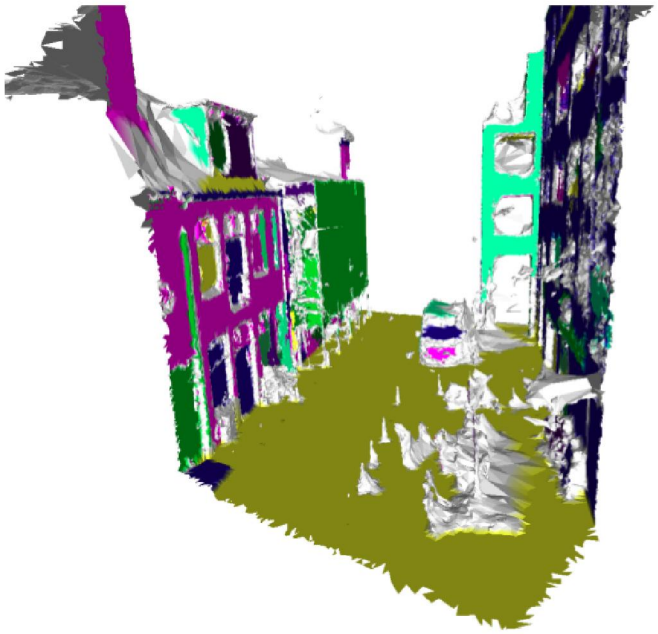


(c)Result with color

Test data 3

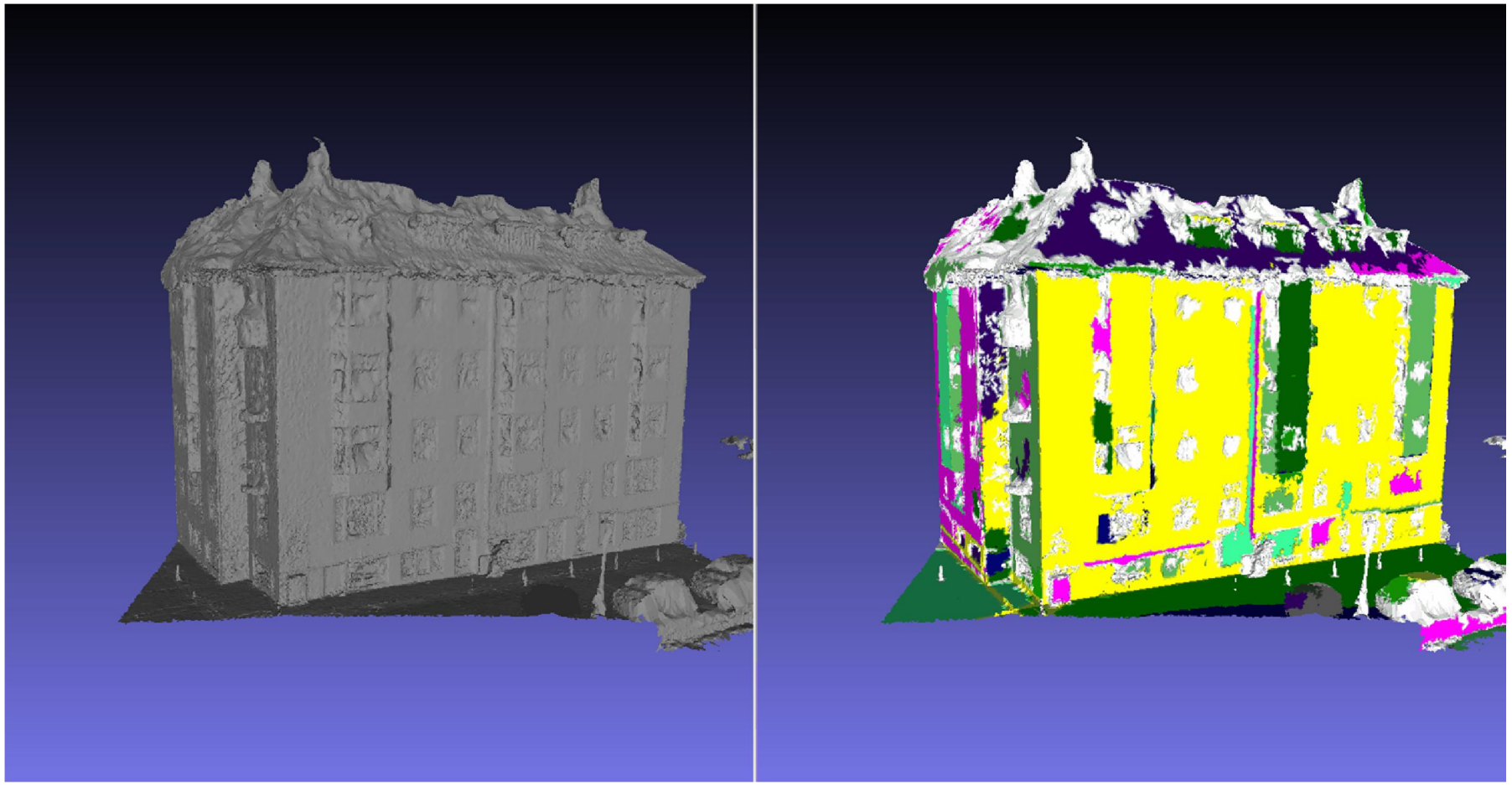




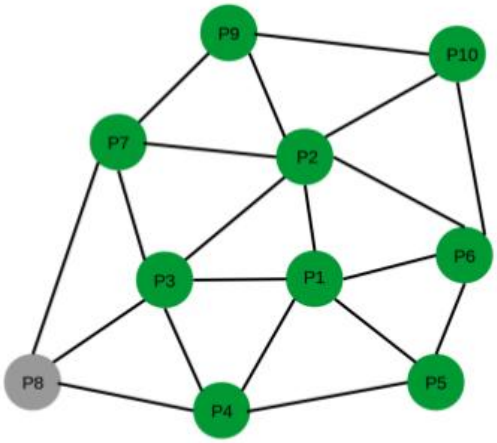


Test data 4

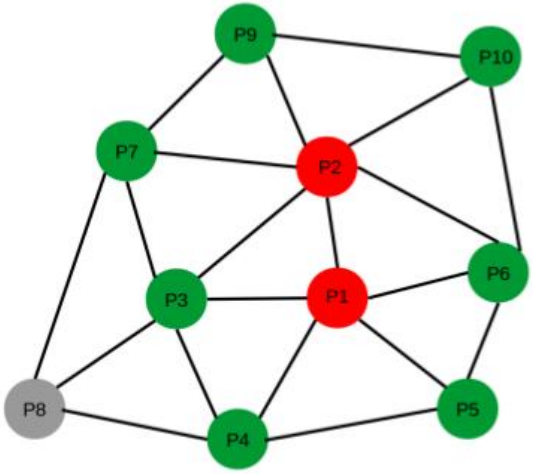
Demo



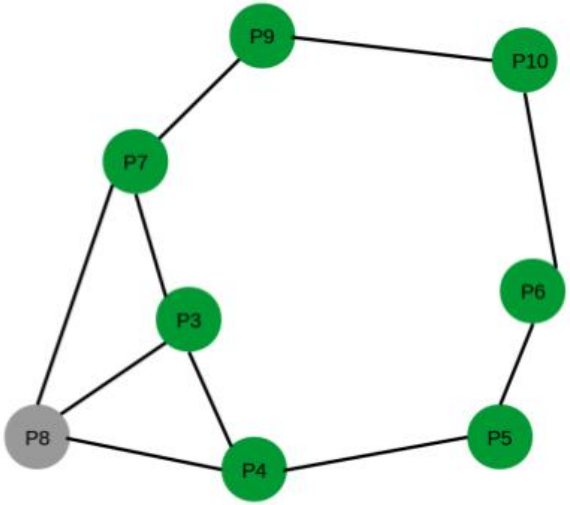
Mesh simplification



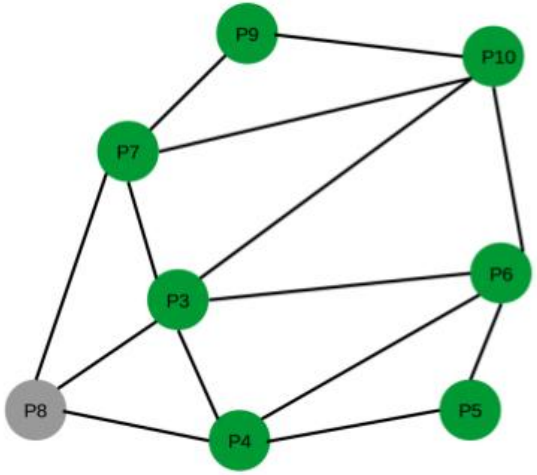
(a) Original mesh



(b) P_1 and P_2 are defined as unneccessary points according to the rules

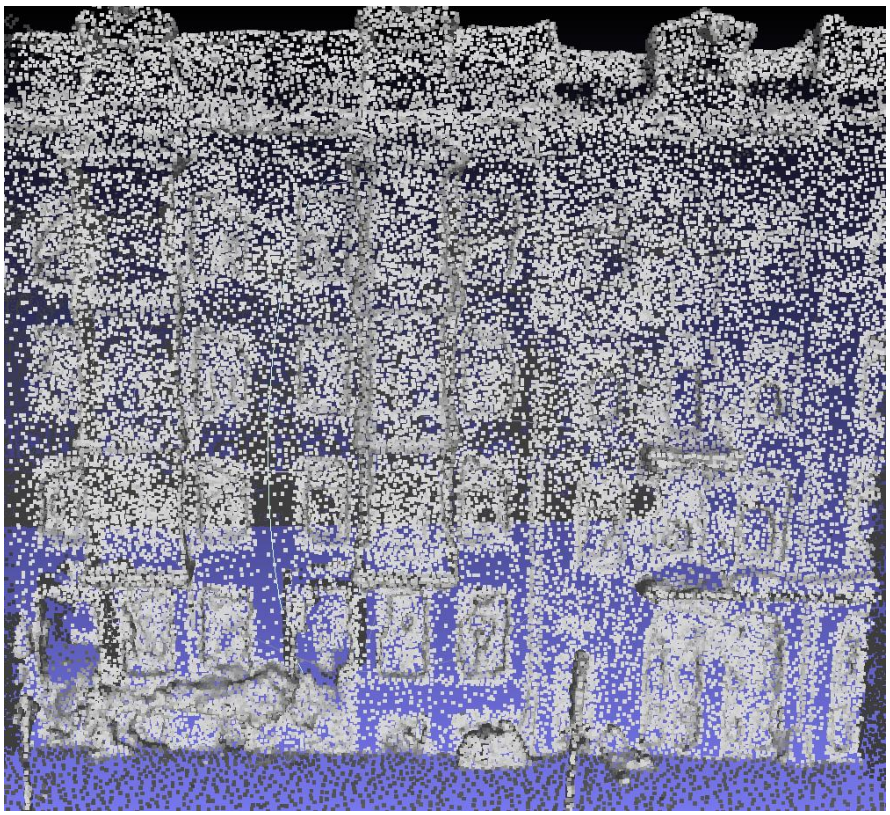


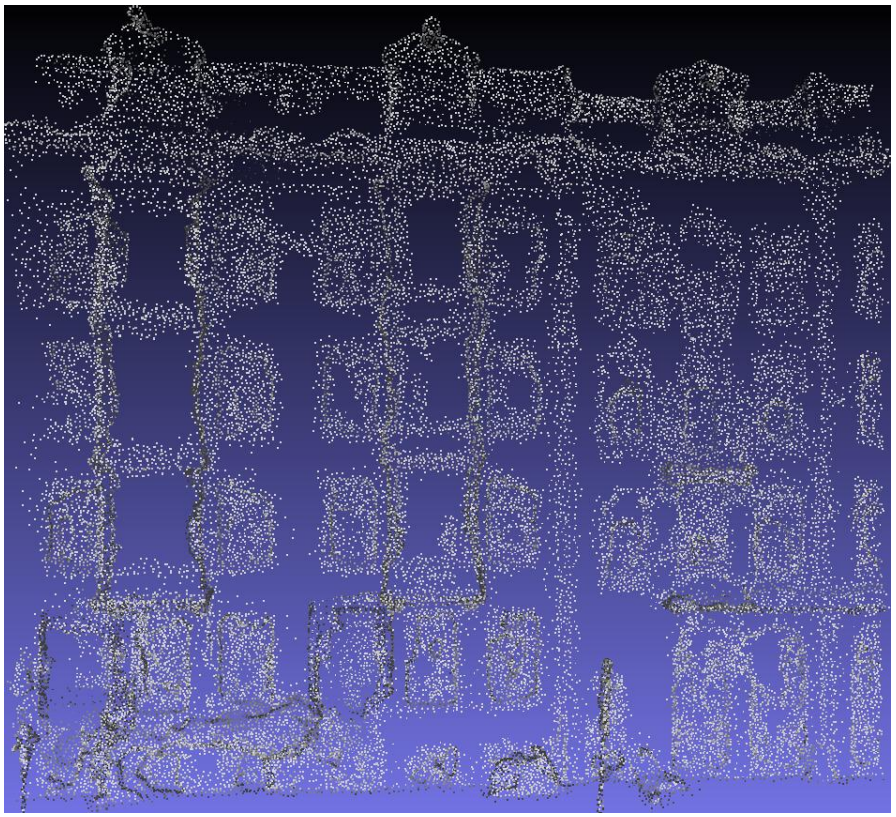
(c) Remove unneccessary points



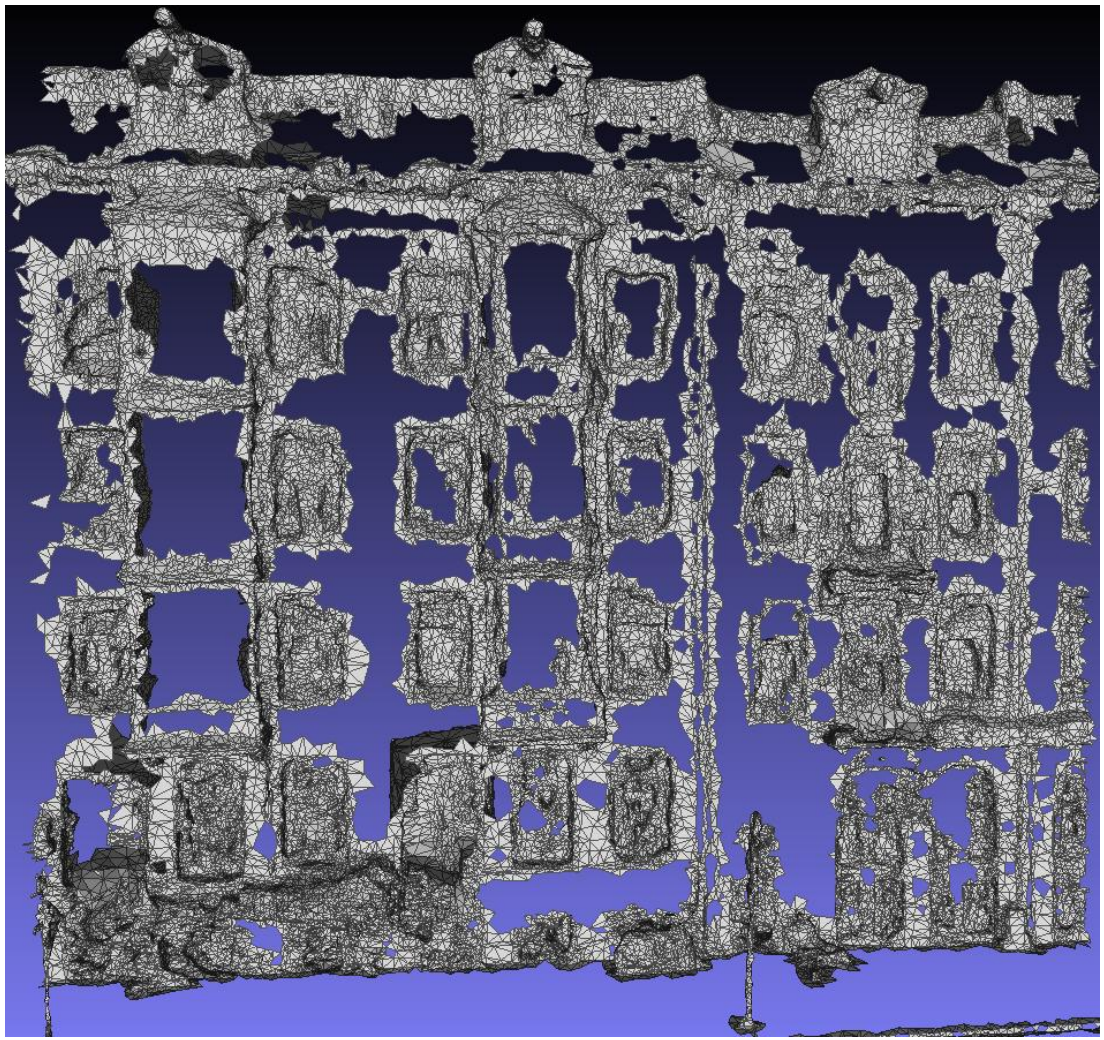
(d) Retriangulate the mesh

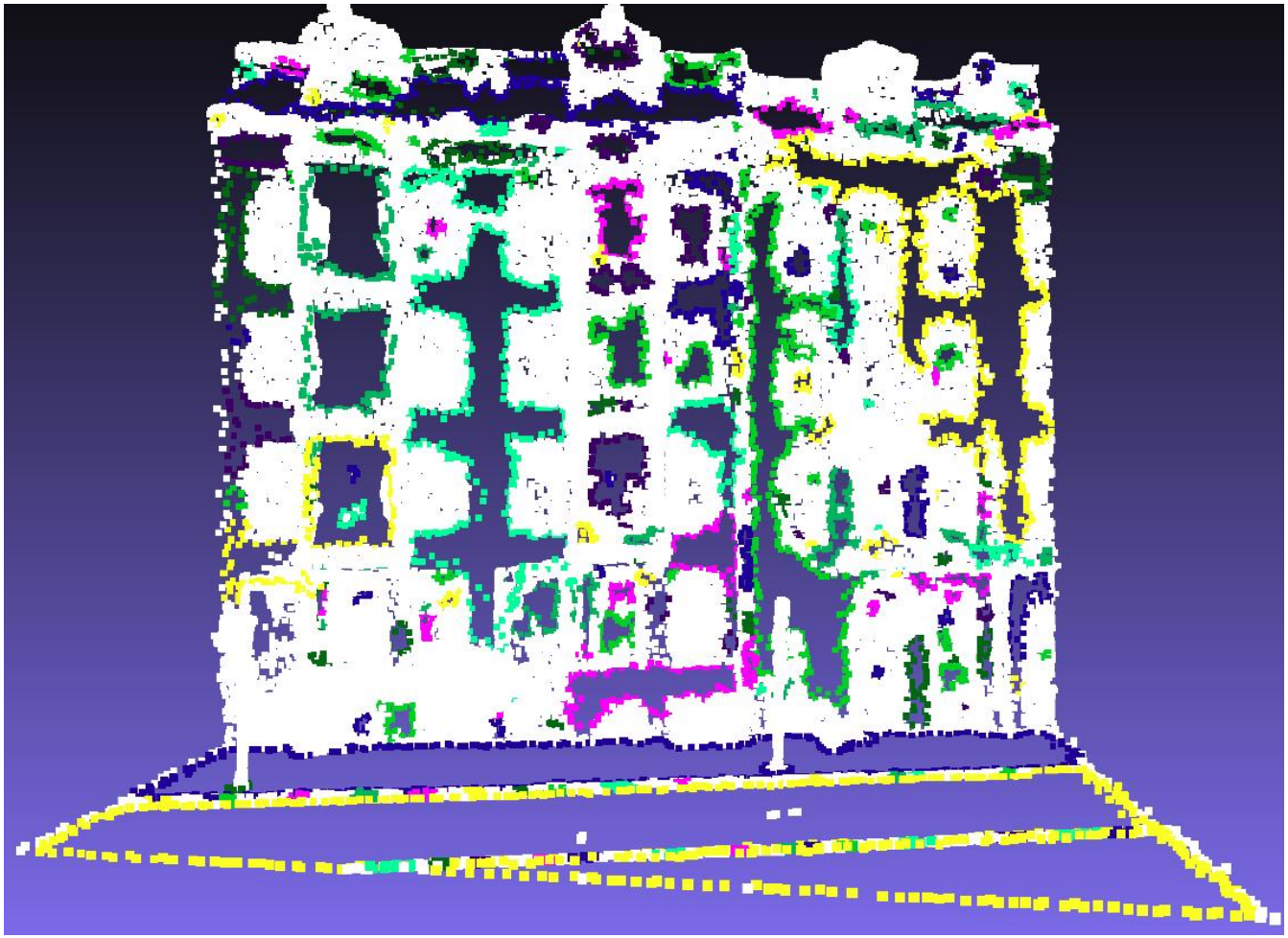
Remove unneccessary vertices

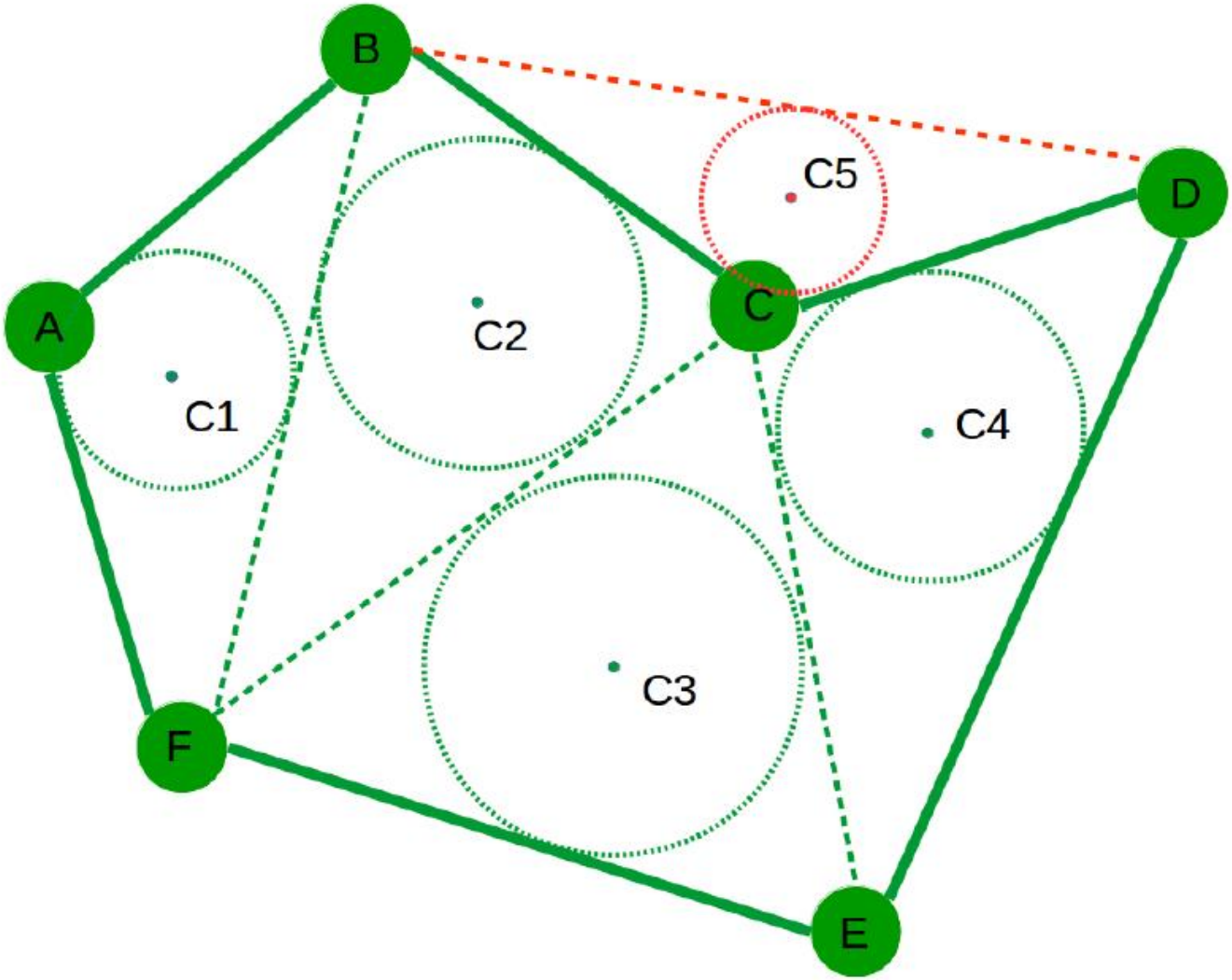


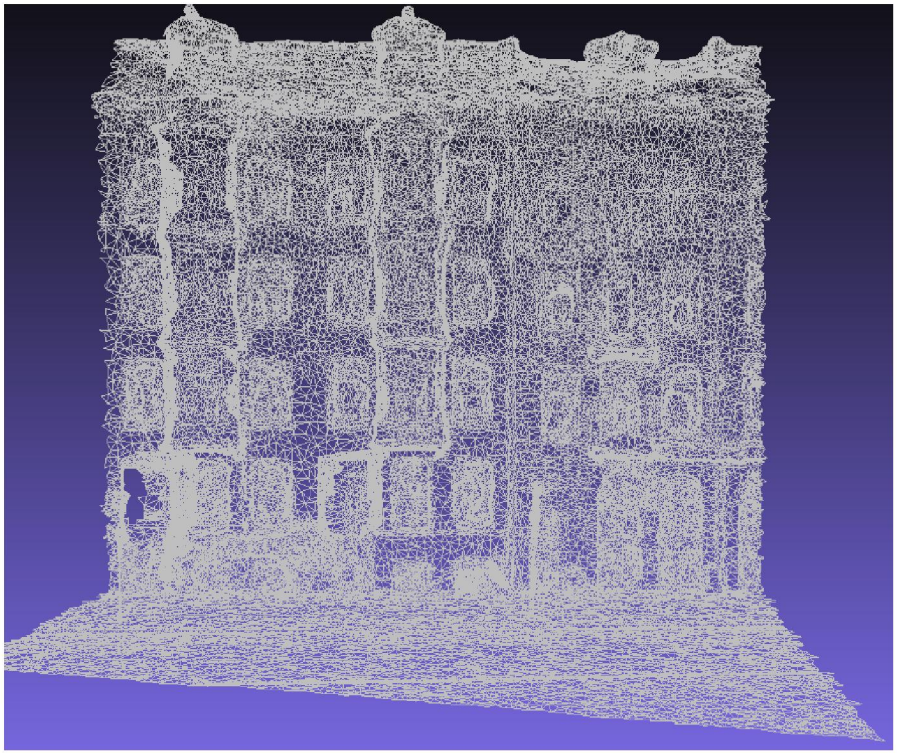


Kept faces



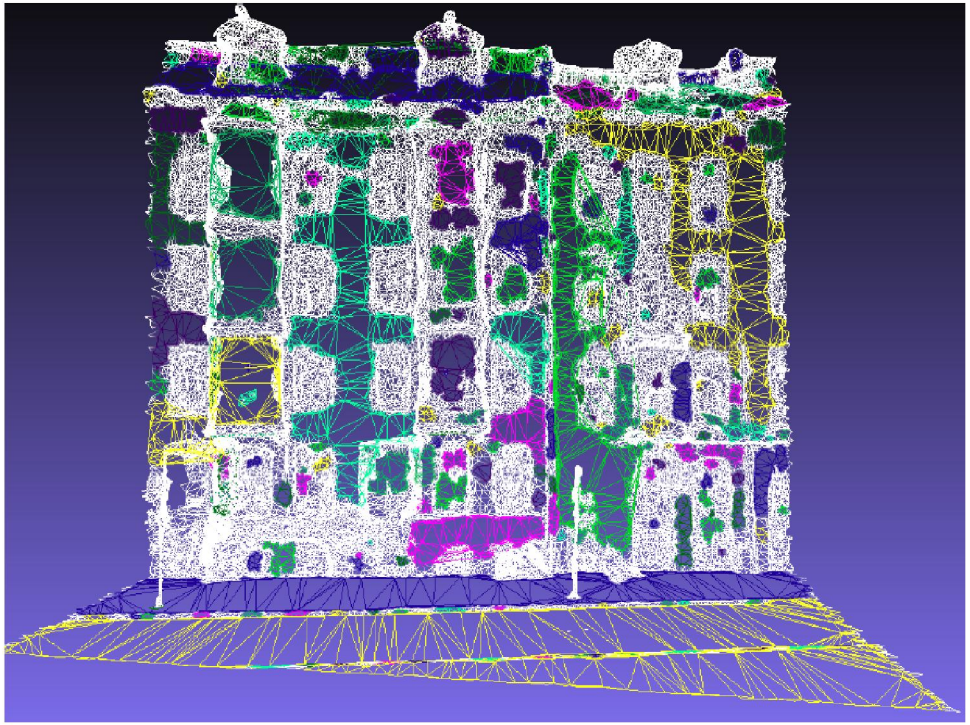



B
C5
D
A
C2
C
C1
C4
C3
F
E



(a) Original mesh

Original mesh: 54962 vertices and 109082 faces

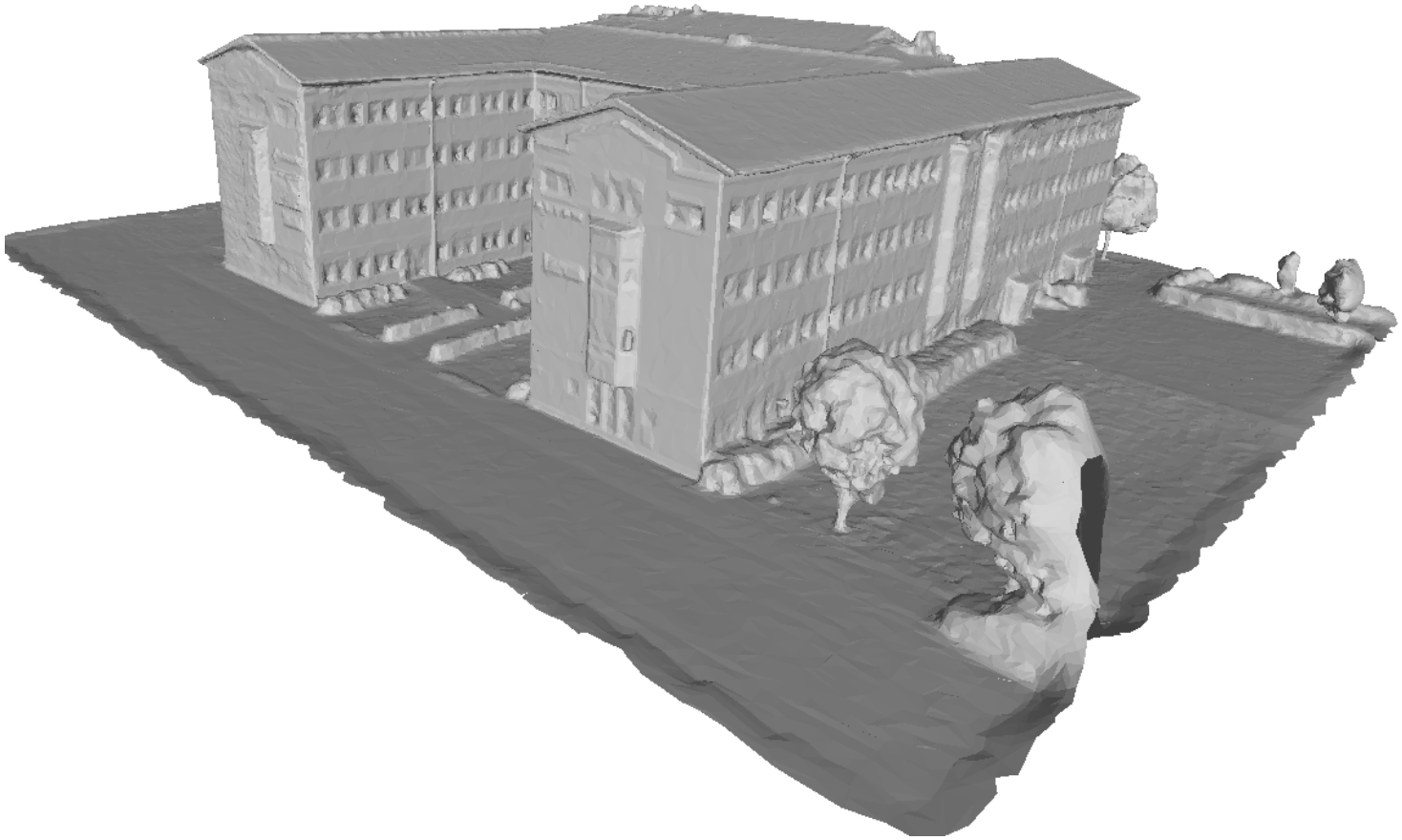


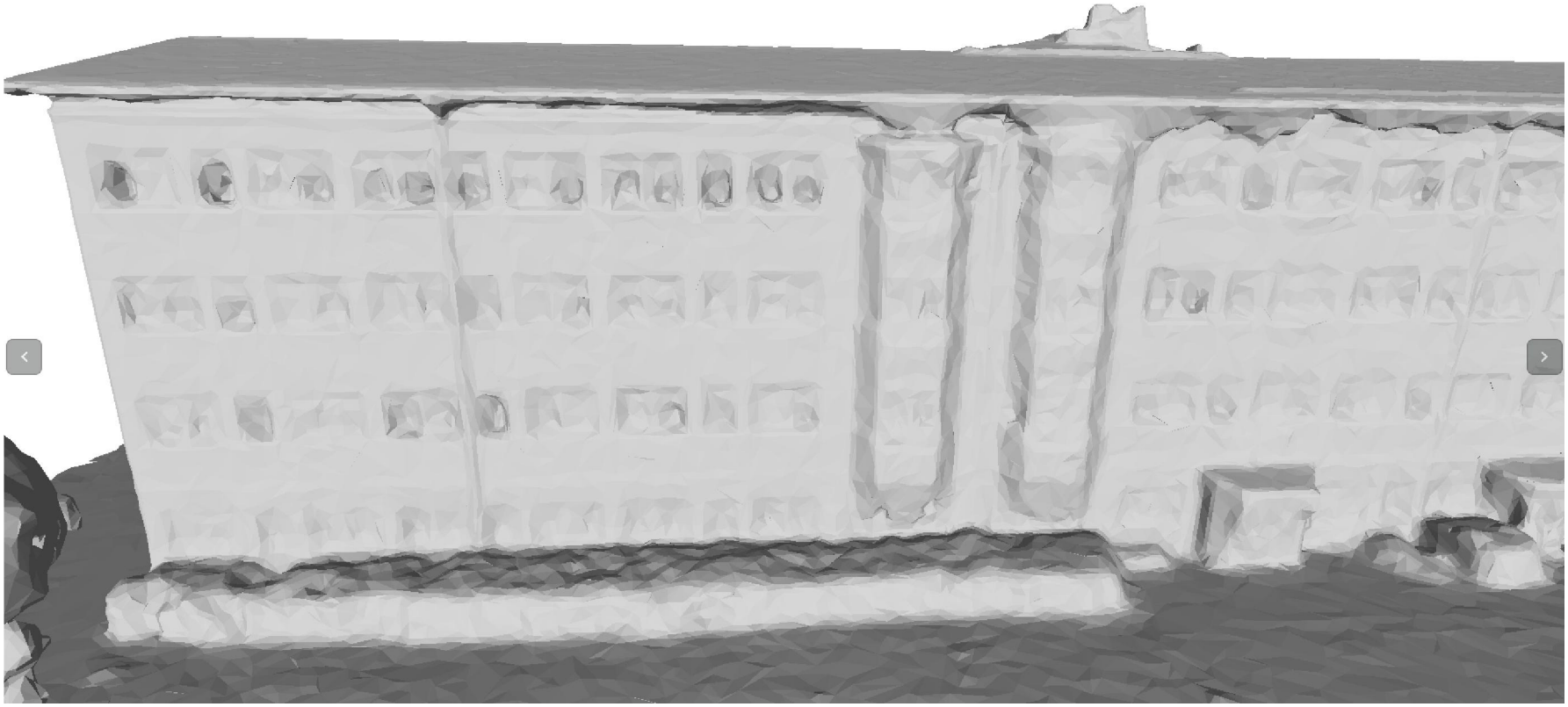
(b) Simplified mesh

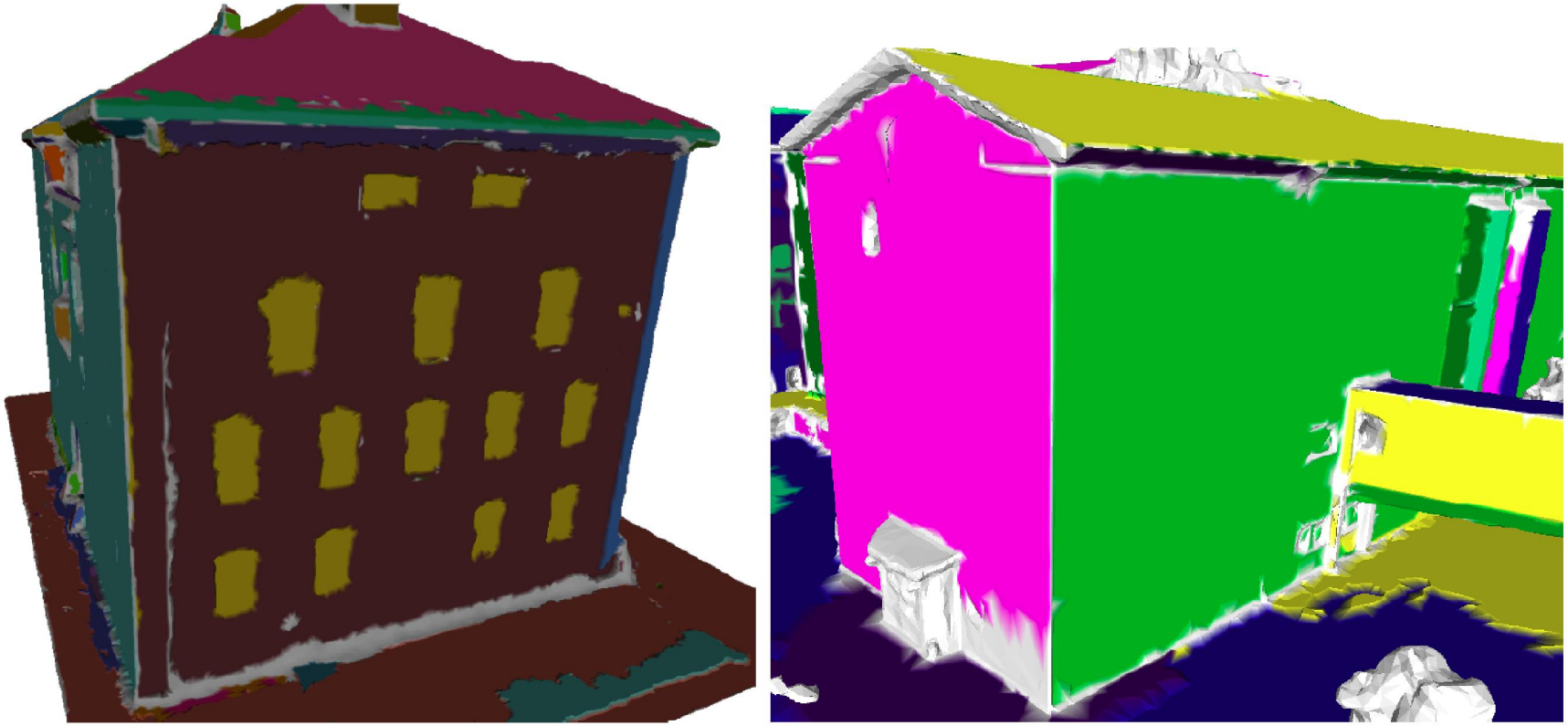
Simplified mesh: 44624 vertices and 88452 faces

TU Delft

Comparison







(a) Result from Jonsson [2016]

(b) Result from this thesis

(a) Result from Jonsson [2016]

(b) Result from this thesis

Conclusion

- Can RANSAC algorithm based method yield similar or better result than existing approaches for straightening multi view stereo mesh?

Yes, according to the comparison, it produces similar result.
More importantly, it can apply to data with low quality

- Subquestions:

- 1. What methods are currently used? What are the advantages and disadvantages?
- Segmentation based. Stable and good results but only when segmentation can achieve good result.

- 2. How can some plane constraints be used for straightening meshes?
- Plane constraints can be used to improve the quality of detected planes. However, plane constraints should be used carefully otherwise they will have negative influences on the detected planes.

- 3. How can geometry/topology/texture information be used?
- Plane fitting is based on the geometry of the points. Topology information is useful in mesh segmentation. Moreover, in mesh simplification, the topology information is used as well. Texture information can be used in region growing.

- 4. Is it feasible to simplify the straightened meshes regarding data storage and attach textures to simplified meshes?

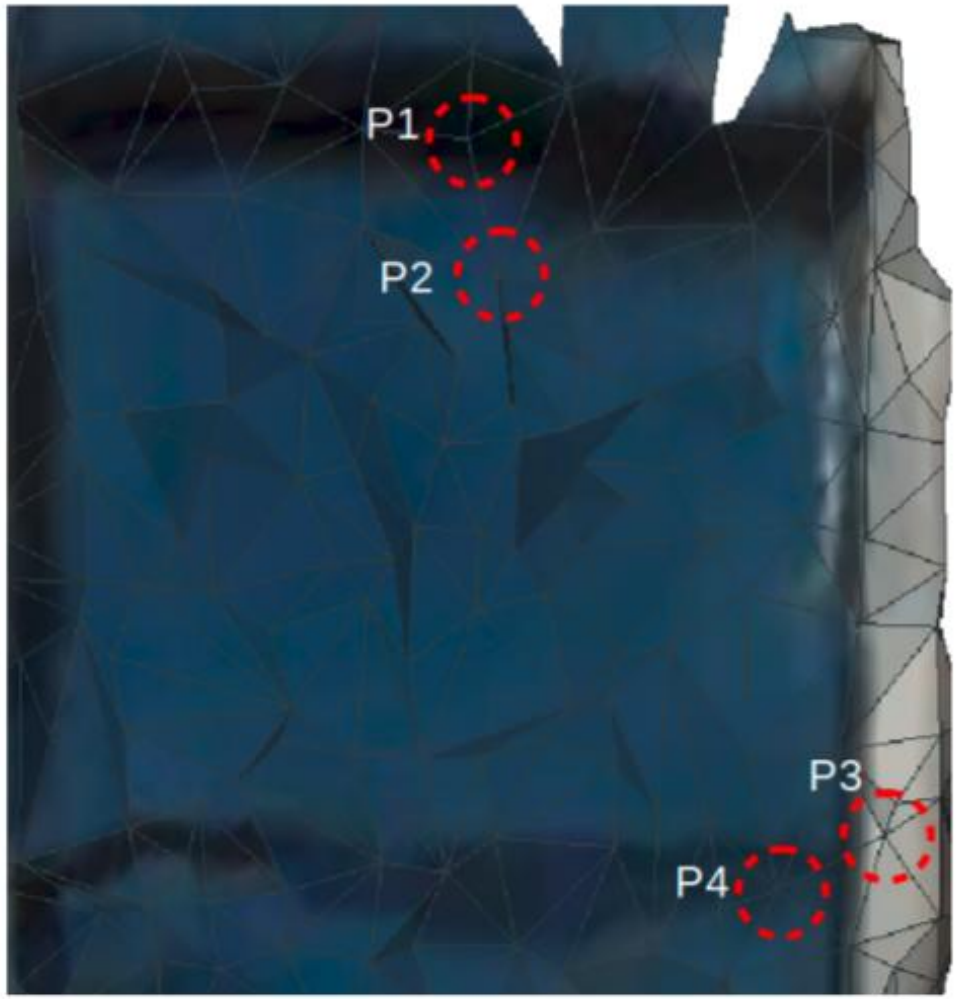
- Since many vertices are snapped to the same planes, it is not necessary to keep all the triangles, so many triangles are removed after simplification. The indices of the vertices and the number of triangles are changed. The texture information is highly related to triangles and indices, thus it is tricky to attach textures back to simplified meshes.

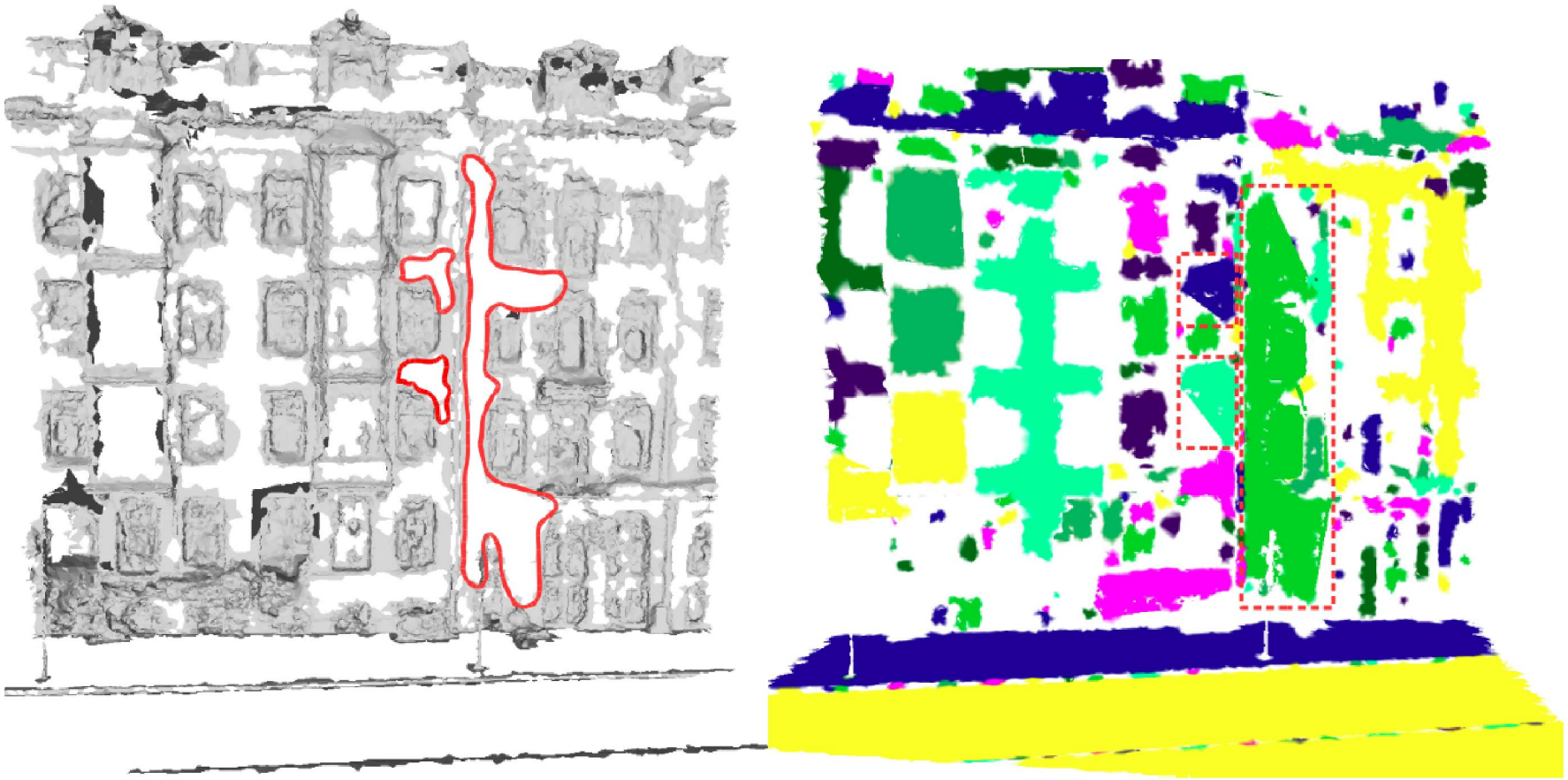
Recommendation

- RANSAC does not consider spatial relations but only relies on the parameters of detecting a plane. i.e points on two separate buildings might be inliers of the same plane.
- Detect ground first, then segment different buildings, input individual buildings to RANSAC.

- Using finite polygon instead of infinite plane. i.e not all
- building planes intersect but with infinite plane representation, every two planes intersect.

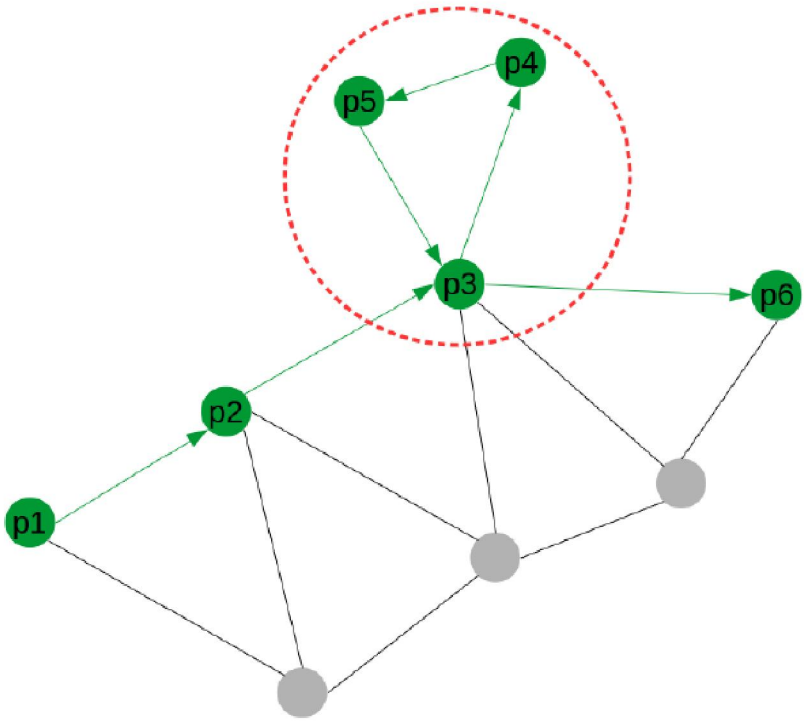
- Better use texture information
- Smooth image to get rid of some details
- Use color of the triangles



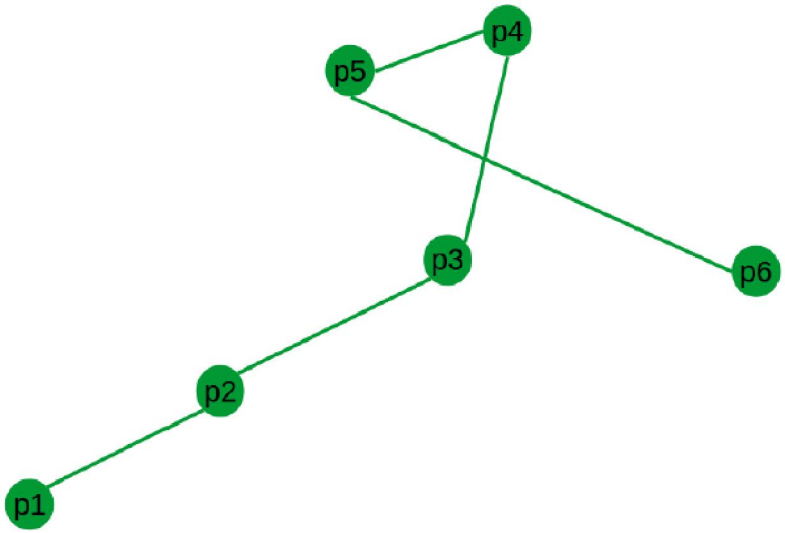


(a) Original concave shapes of the polygon

(b) Convex shapes after triangulation



(a) Edge-based region growing on non-manifold mesh



(b) Self-intersection problem

Thank you!