UNIVERSITY OF TWENTE.

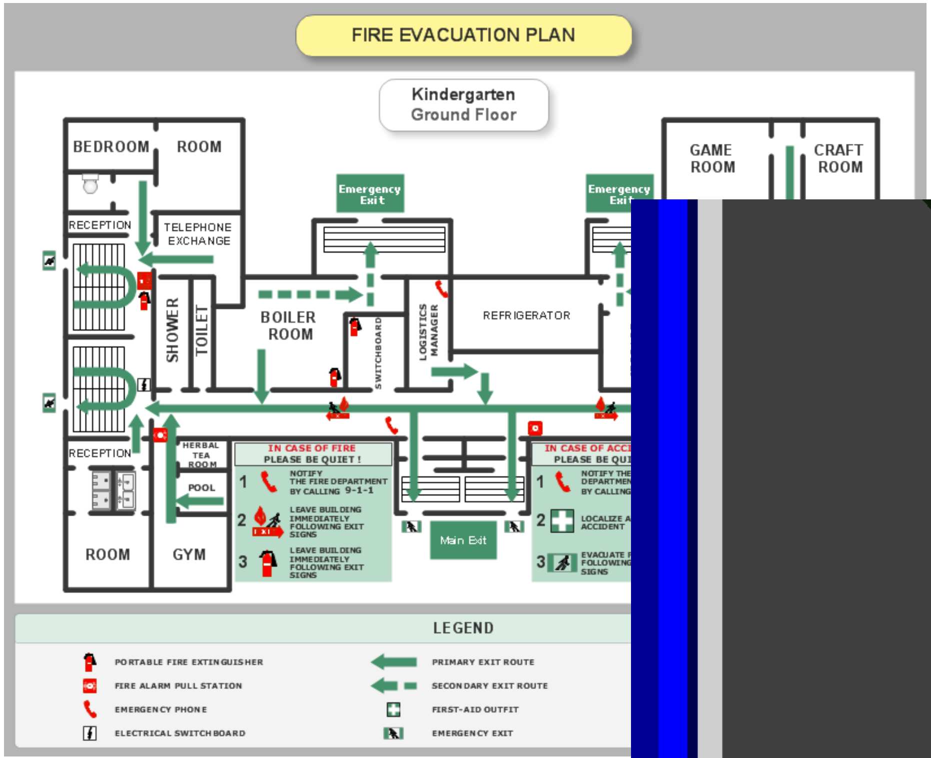
3D modeling of interior spaces: Learning the language of indoor architecture

Kourosh Khoshelham
Lucia Díaz-Vilariño

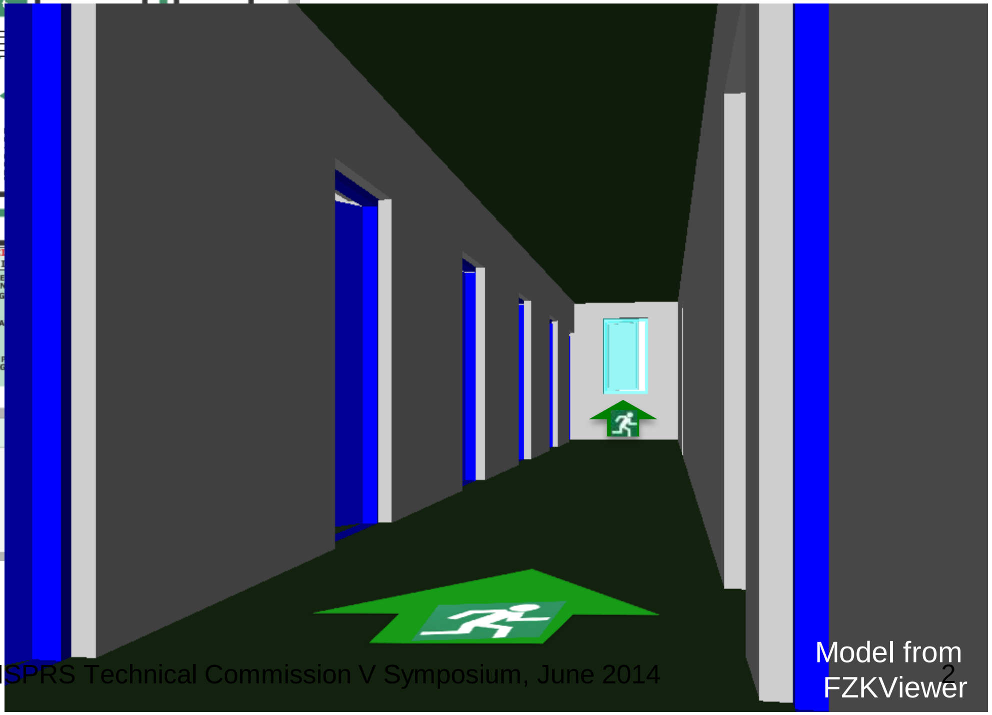
ITC
FACULTY OF GEO-INFORMATION SCIENCE AND EARTH OBSERVATION

NEED FOR 3D INDOOR MODELS

- Crisis management in large public buildings
- Automated route generation and navigation



From: http://www.conceptdraw.com/



Model from FZKViewer

3D INDOOR MODELING: THE CHALLENGE

- Manual modeling:

  → Can take months depending on the complexity of the building and the required level of details.

- Automated modeling:

  → Should be able to handle a large variety of indoor architectures.

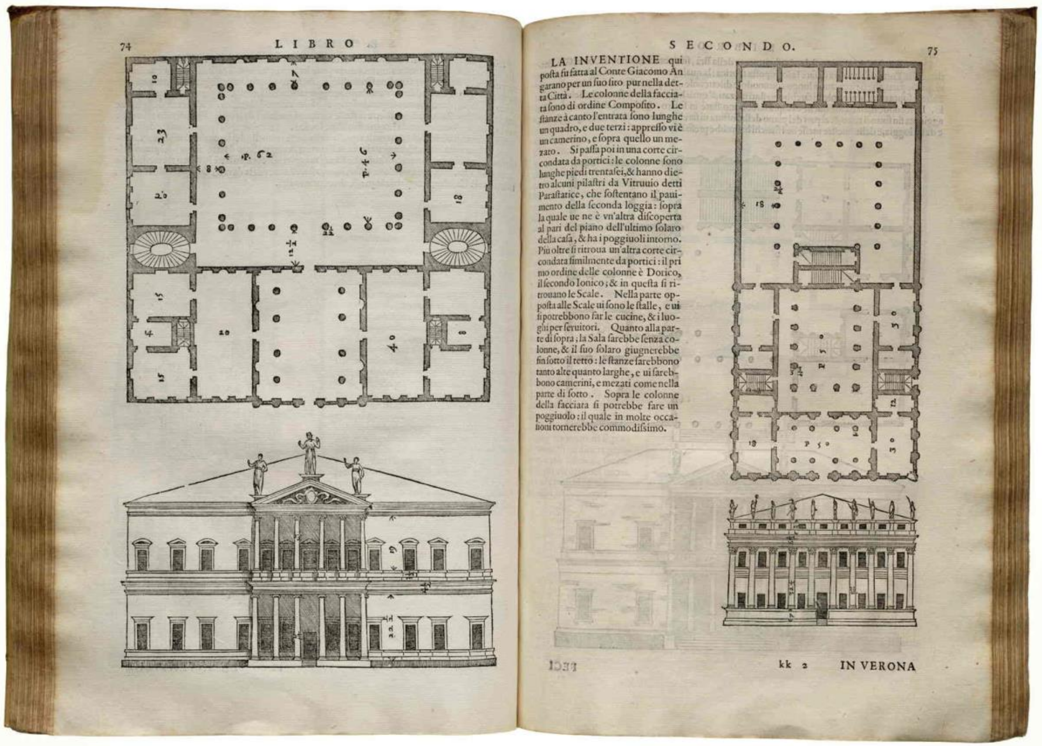
The Cubicus building in the UT campus.

INDOOR ARCHITECTURAL DESIGN

- Characterized by:
  - → Repetition
  - → Regularity
  - → Creativity

- Example: Palladian indoor designs

Andrea Palladio (1508 –1580)

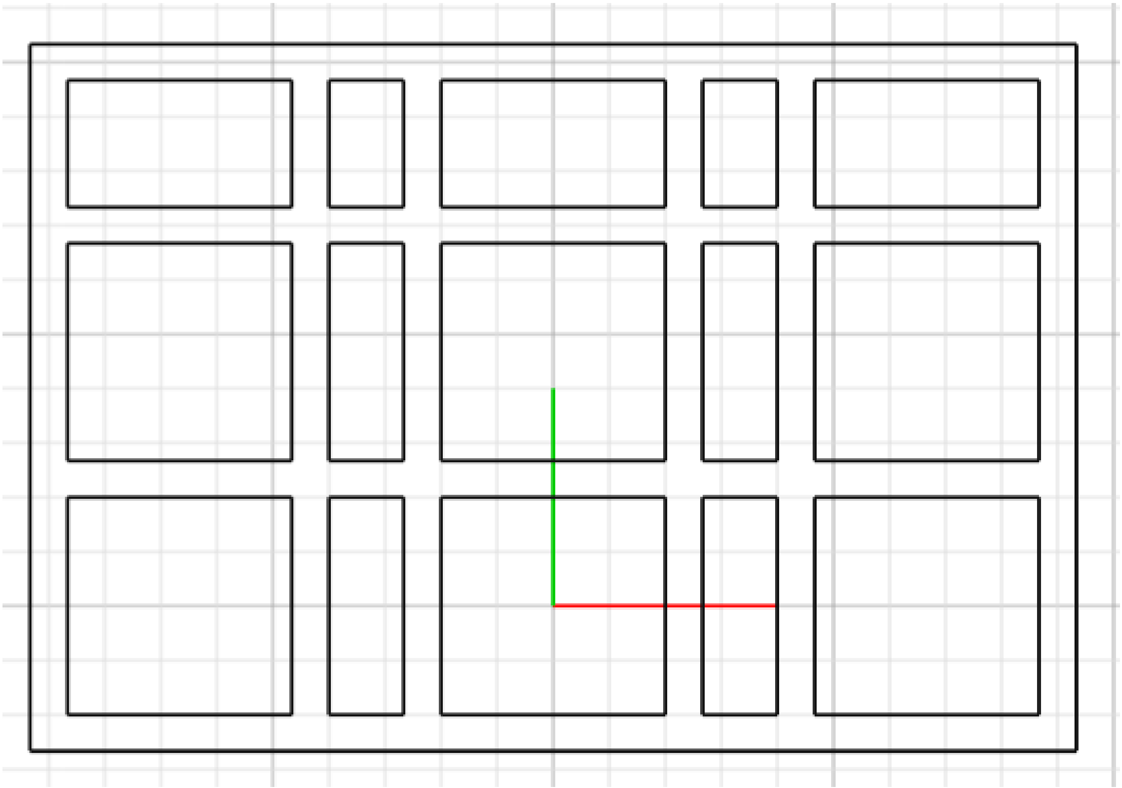


PALLADIAN GRAMMAR (SIMPLIFIED)

- Rule 1: Make a grid of rectangular spaces

Repetition

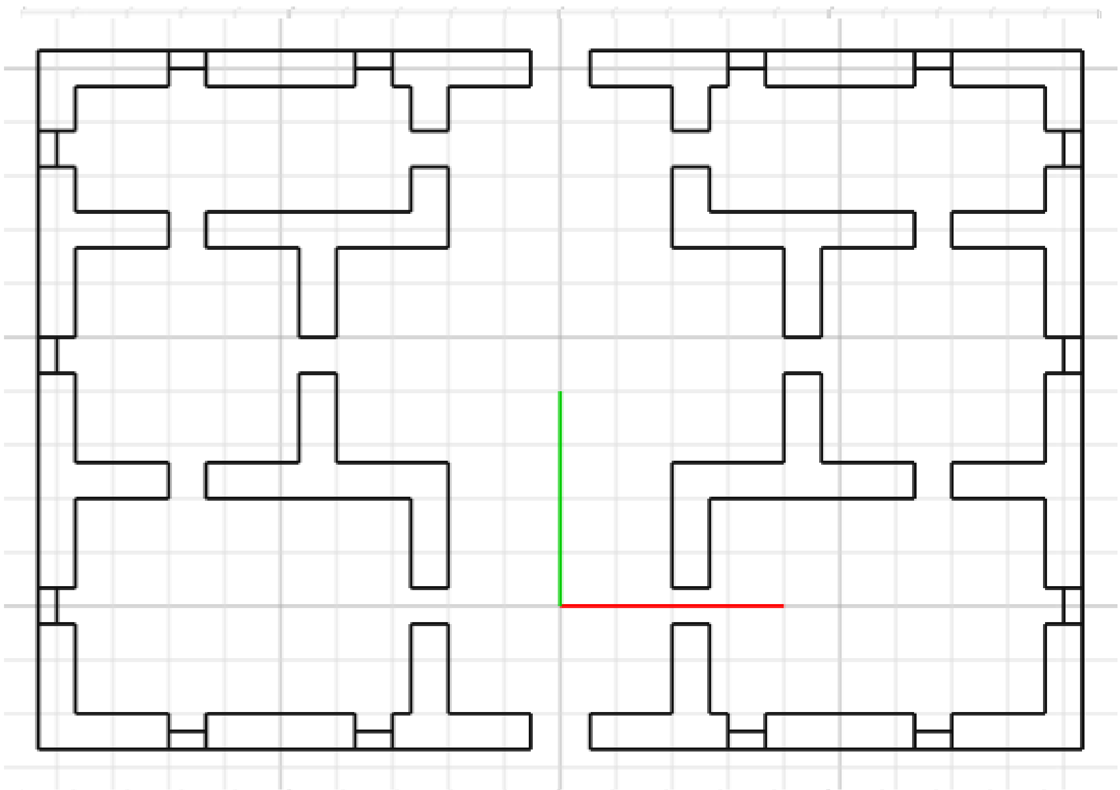
Regularity



http://grape.swap-zt.com/

PALLADIAN GRAMMAR (SIMPLIFIED)

- Rule 2: Collapse <u>some</u> of the walls → Creativity
- Rule 3: Insert aligned doors and windows



http://grape.swap-zt.com/

A SHAPE GRAMMAR FOR INDOOR MODELING

- Starting symbol (S): a unit cube

- Rule 1: place a cuboid
  If there are points on its ceiling

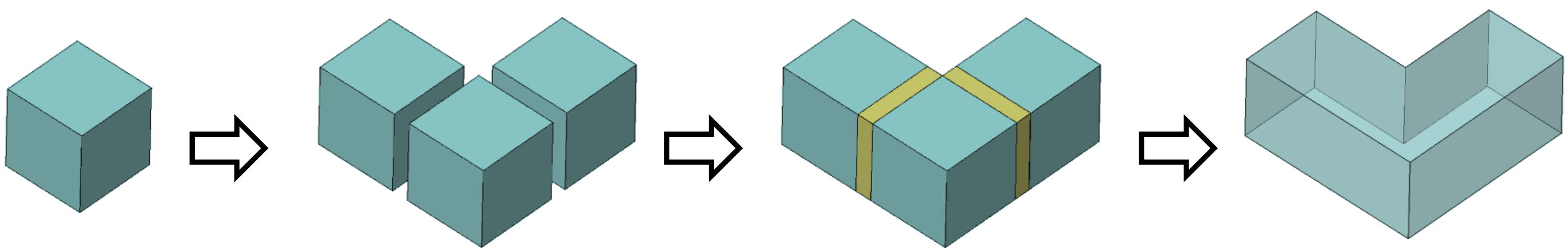
  $R^{1}_{\text{place\_cuboid}}: S \rightarrow H \cdot S$

- Rule 2: connect two cuboids
  If they are not separated by a wall

  $R^{2}_{\text{connect cuboids}}: \{N_1, N_2\} \rightarrow N_3$

- Rule 3: merge two cuboids
  If they have a common face

  $R^{3}_{\text{merge\_cuboids}}: \{N_1, N_2\} \rightarrow N_3\ (T)$



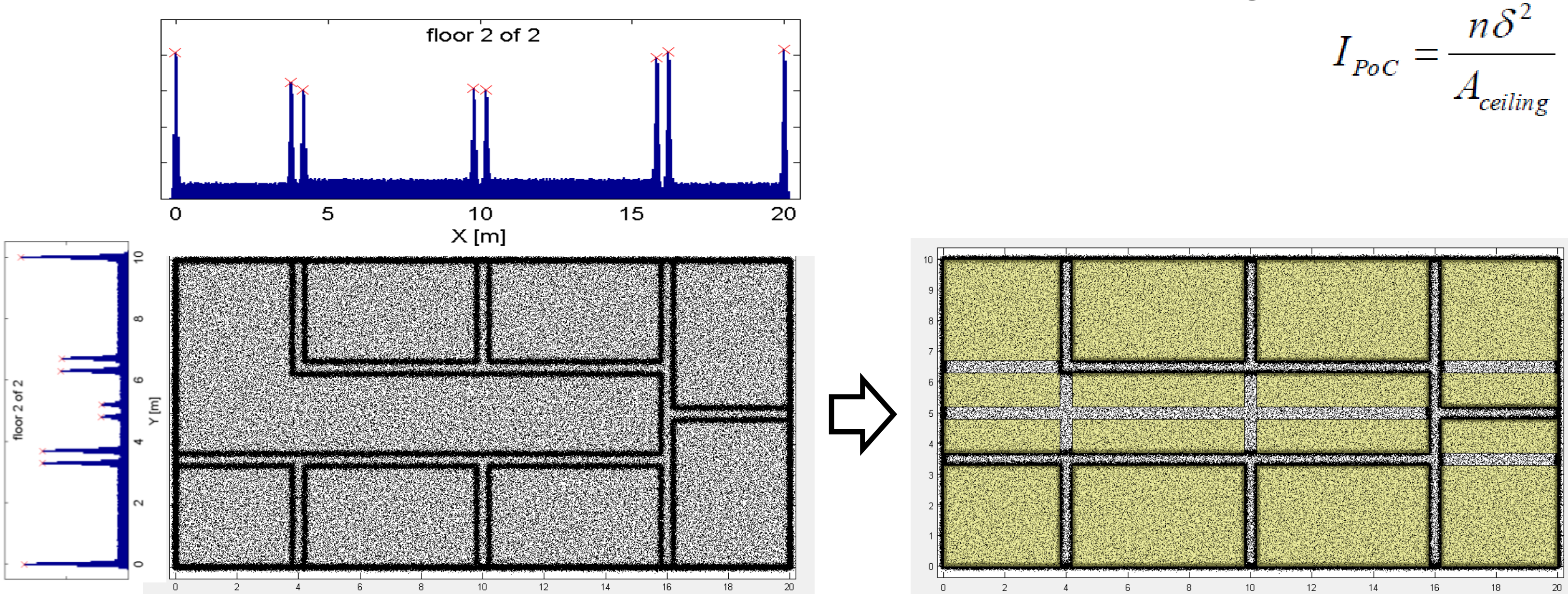
$S - R^1 - R^1 - R^1 - R^2 - R^2 - R^3 - R^3 - R^3 - R^3$

LEARNING GRAMMAR RULES FROM A POINT CLOUD

- Rotate point cloud such that walls are aligned with x- and y- axes;
- Location and size of cuboids from histogram of x, y, z coordinates;
- Constraint: each cuboid should have points on its top face.

Points-on-ceiling index:

$$I_{PoC} = \frac{n\delta^2}{A_{ceiling}}$$

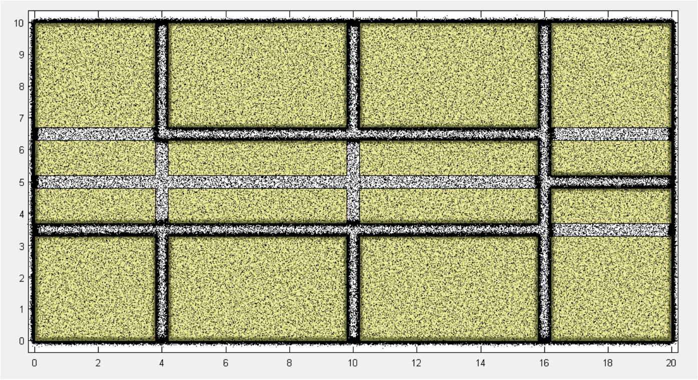


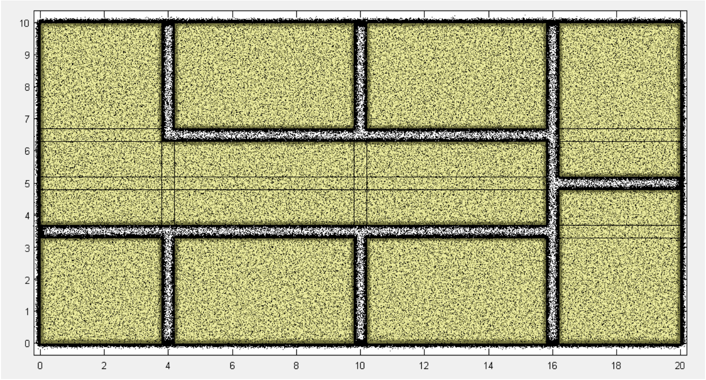
LEARNING GRAMMAR RULES FROM A POINT CLOUD

- Connecting based on:
  1. Adjacency in the initial grid;
  2. If the connecting cuboid is not on an interior wall.

Points-on-wall index:

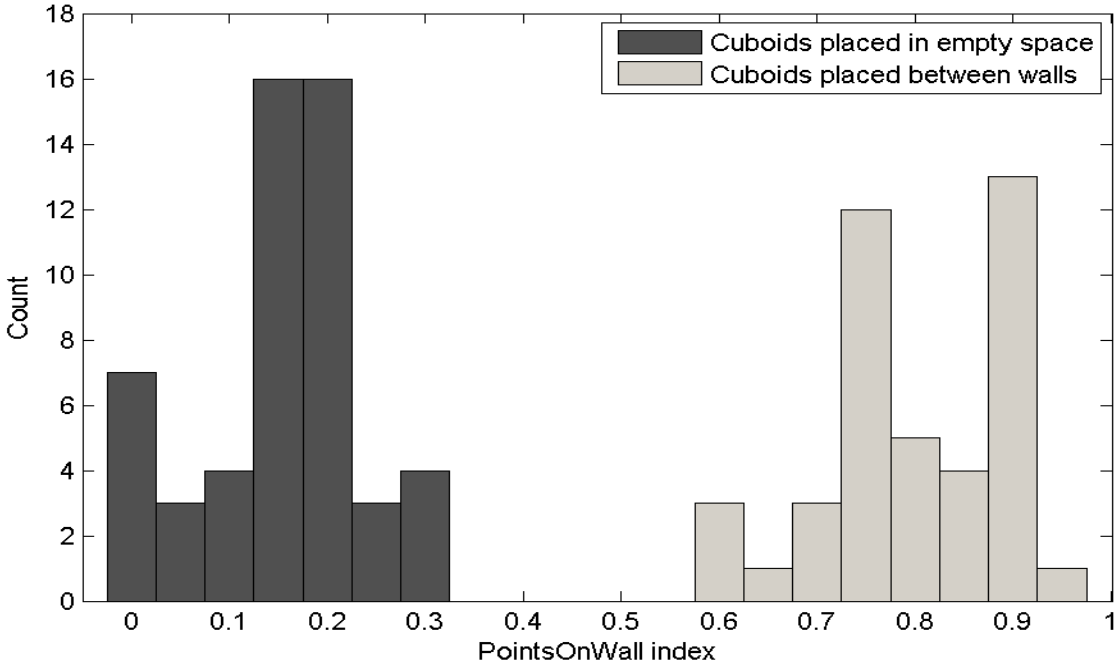
$$I_{PoW} = \frac{n\delta^2}{A_{wall}}$$





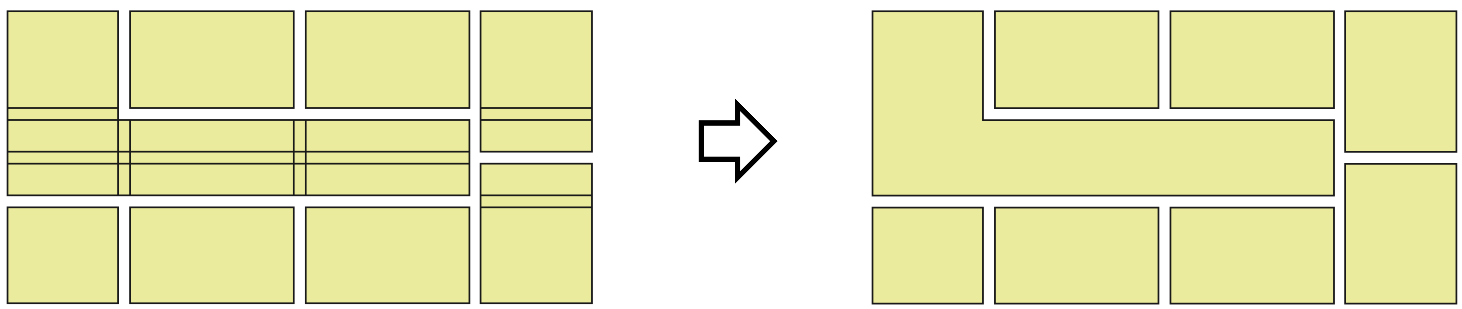
SEPARABILITY OF WALLS AND EMPTY SPACES

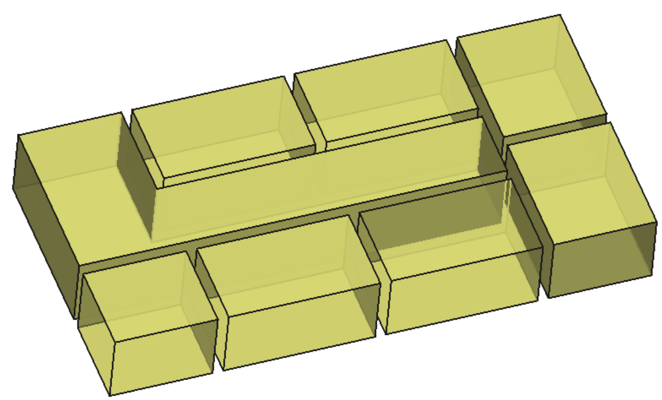
- Using Points-on-Wall index:



LEARNING GRAMMAR RULES FROM A POINT CLOUD

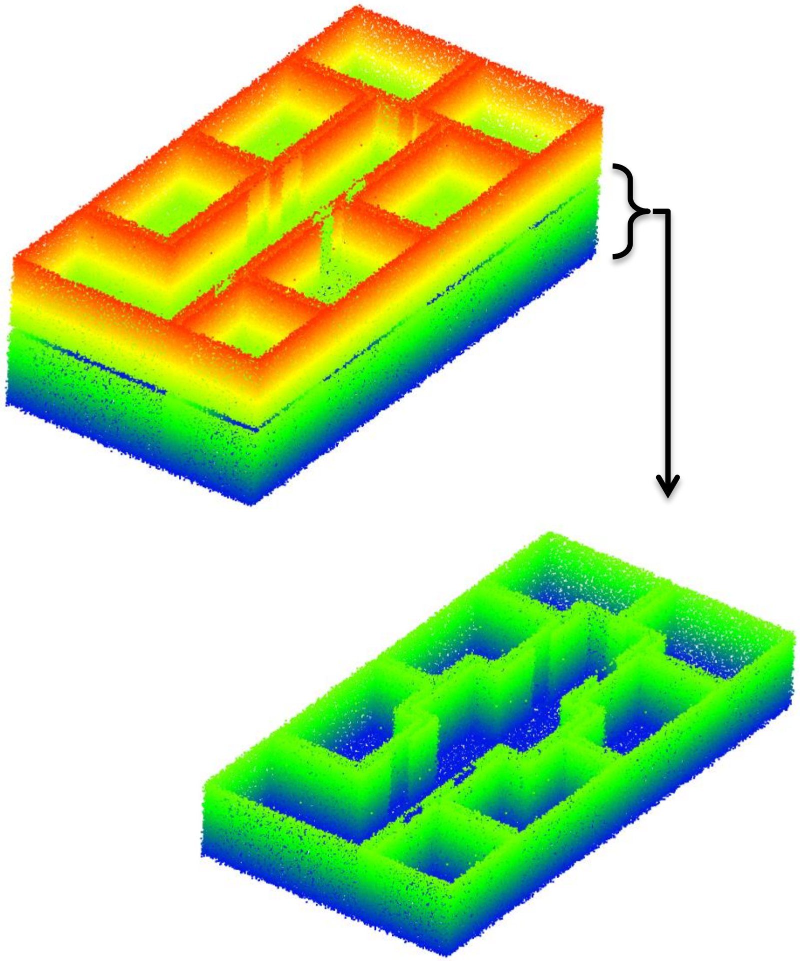
- Merging based on:
  1. If two cuboids have a common face;
  2. If both are non-terminal.



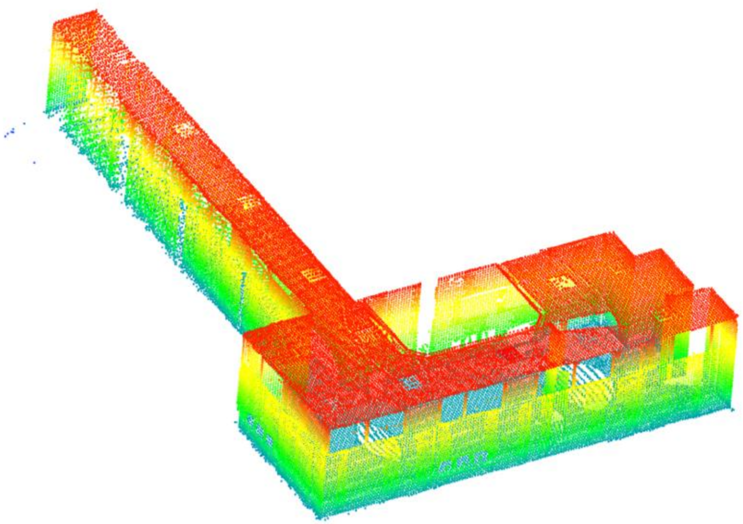


EXPERIMENTS

- Simulated point cloud
  → two-storey; multiple spaces

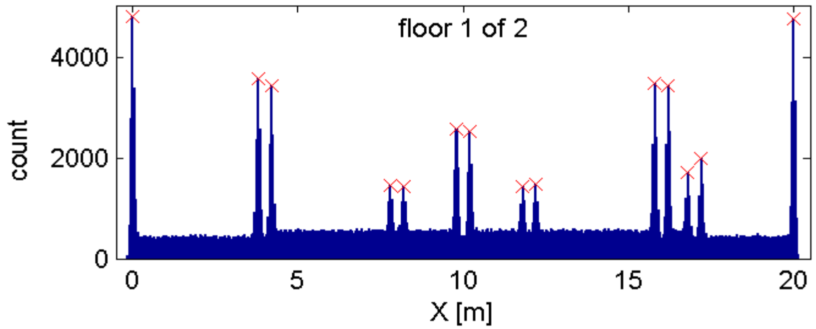


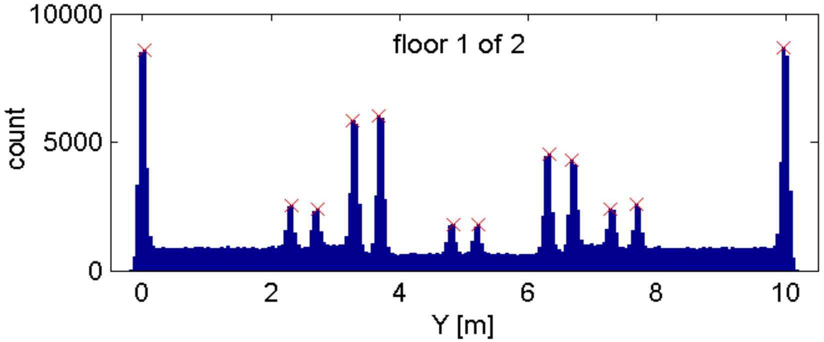
- Real point cloud

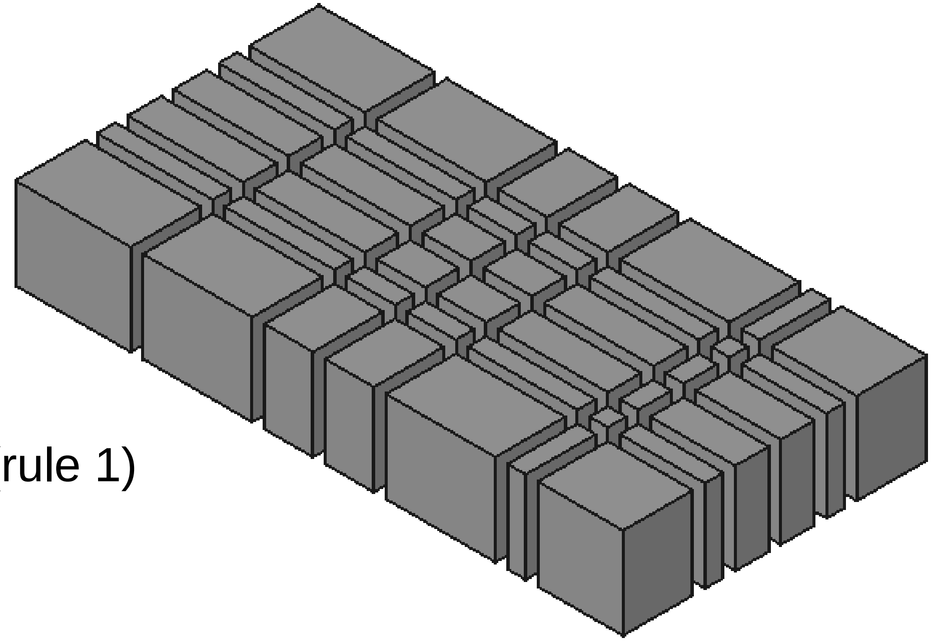


RESULTS

- Simulated 1st floor:



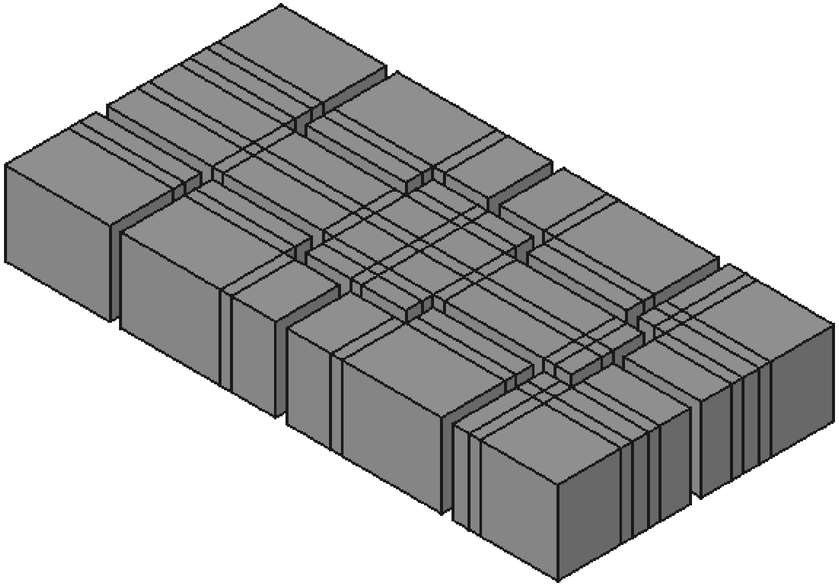




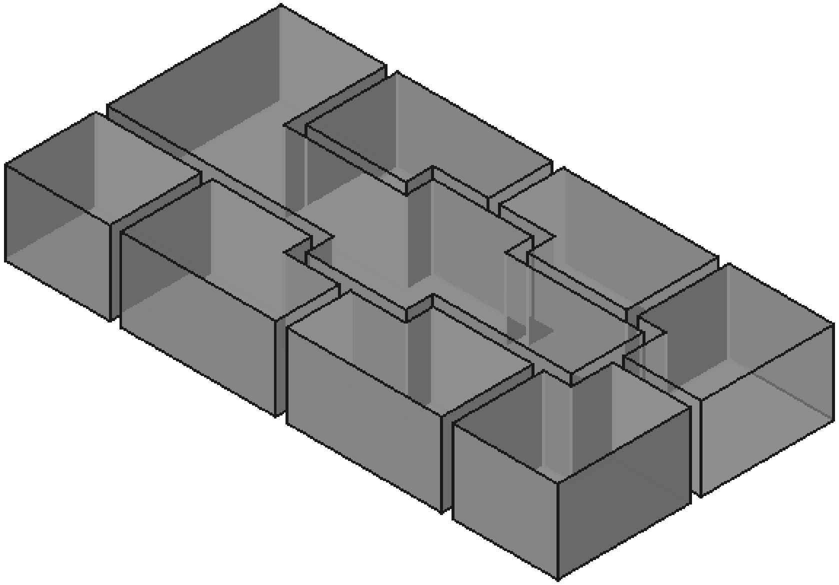
Initial cuboids (rule 1)

RESULTS

- Simulated 1st floor:



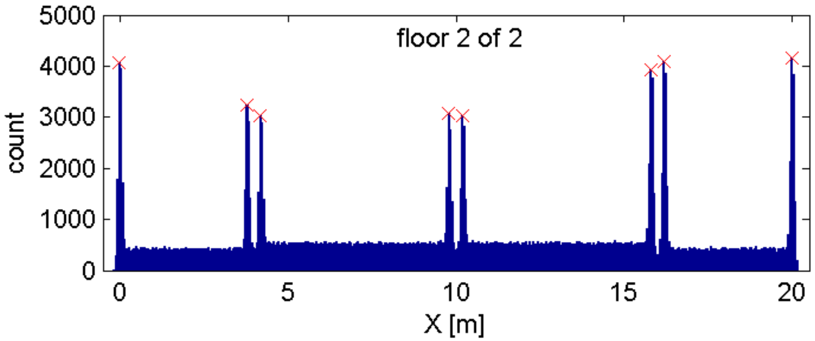
Connected cuboids (rule 2)

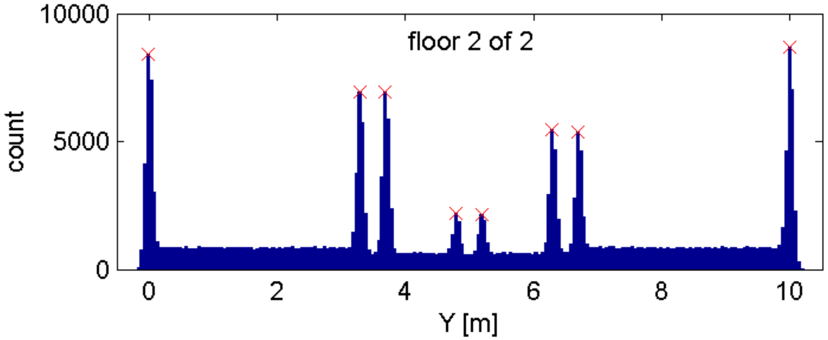


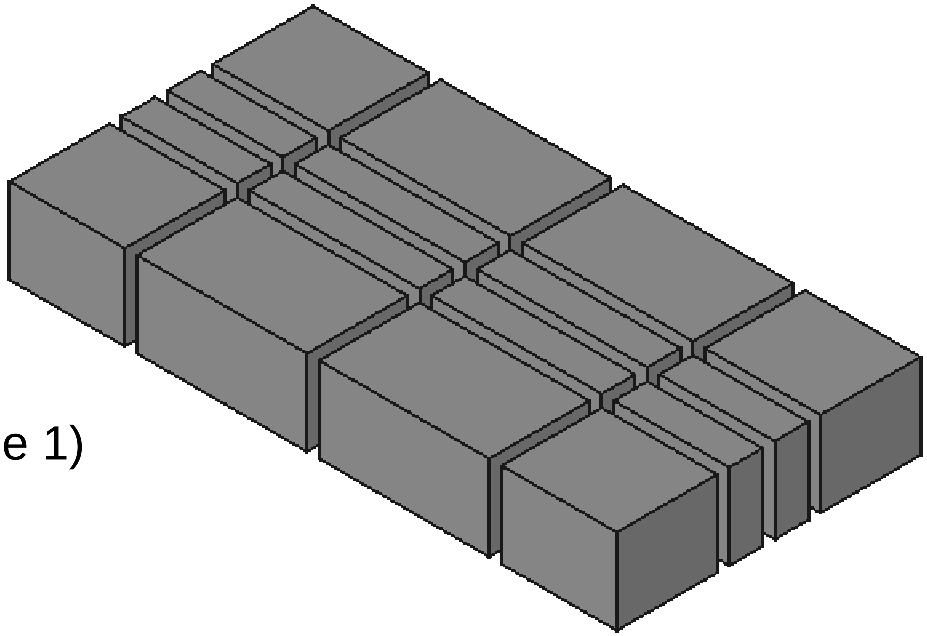
Merged cuboids (rule 3)

RESULTS

- Simulated 2nd floor:



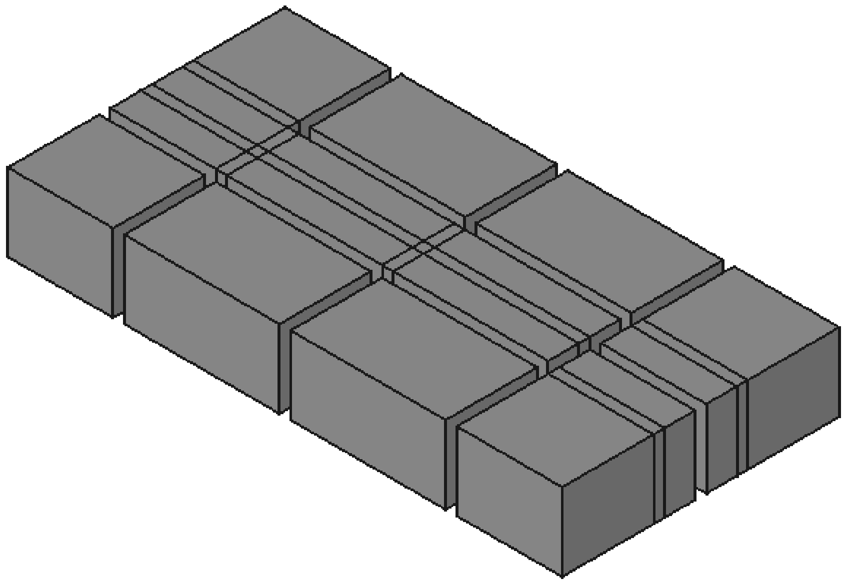




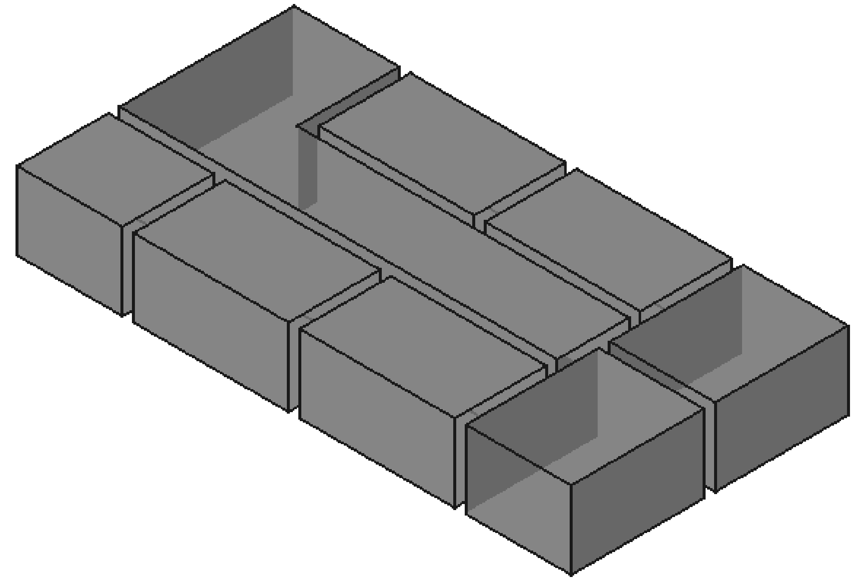
Initial cuboids (rule 1)

RESULTS

- Simulated 2nd floor:



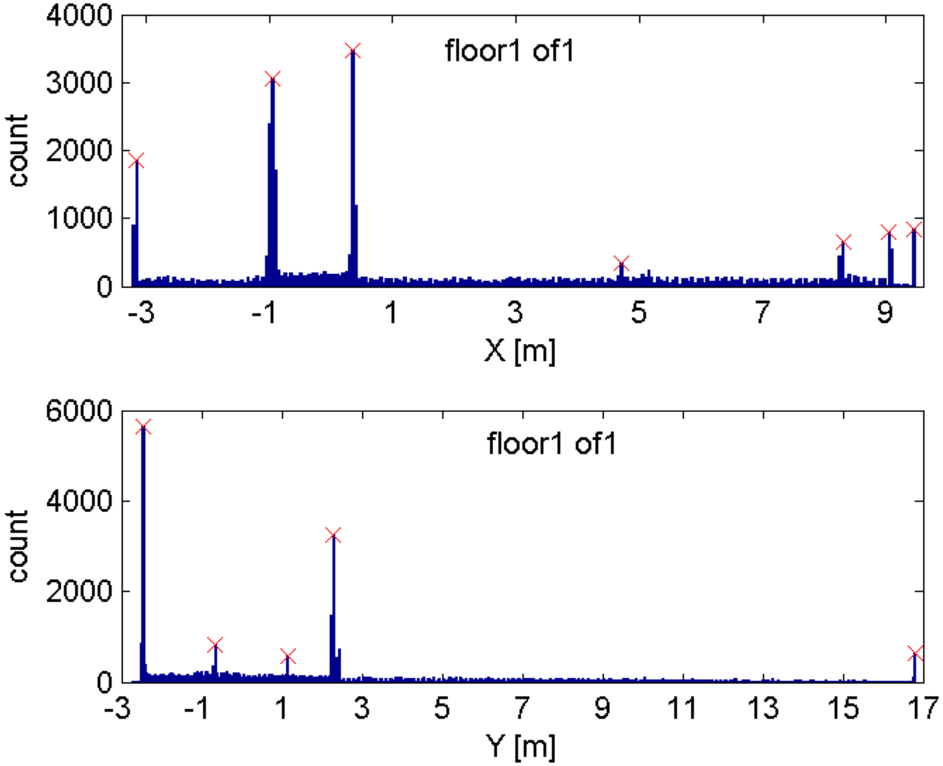
Connected cuboids (rule 2)

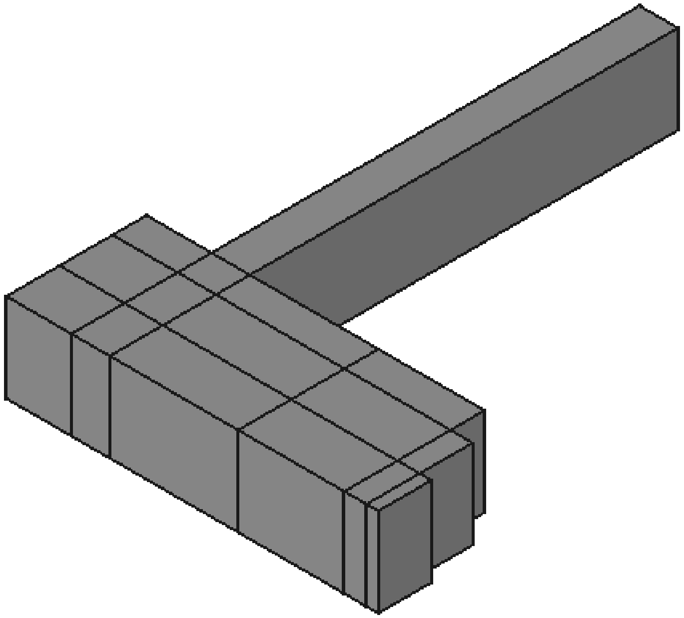


Merged cuboids (rule 3)

RESULTS

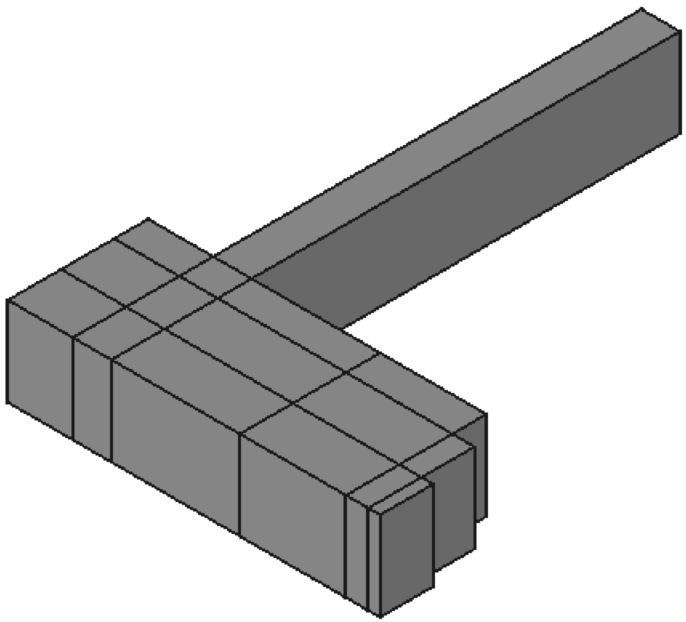
- Real point cloud:



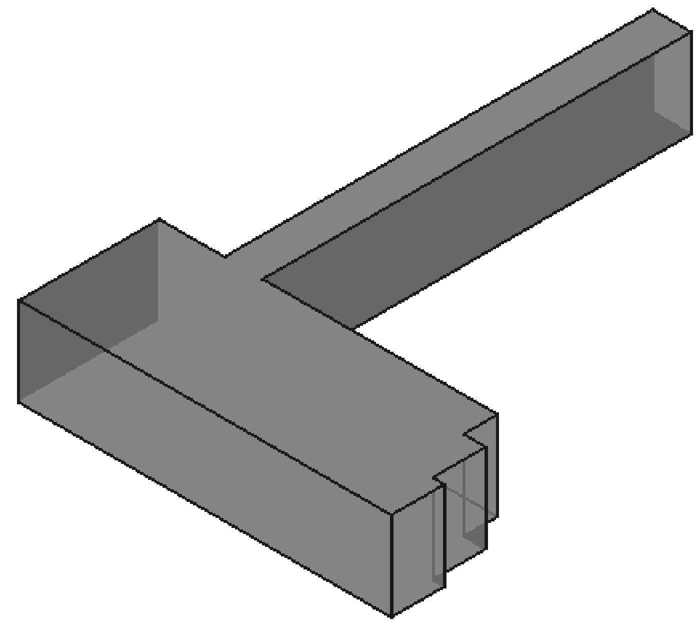


RESULTS

- Real point cloud:



Connected cuboids (rule 2)



Merged cuboids (rule 3)

CONCLUSIONS

- Automated indoor modeling by learning grammar rules from a point cloud;
- Convenient method to create BIM/cityGML models (can provide volumetric solids + surfaces + semantics)
- Adjacency relationships are inherent in the cuboids and can be transferred across rules;
- Non-navigable spaces can be modeled as well using the points-on-wall index.

- Future work:
  - Addition of semantic rules;
  - Extension to handle non-Manhattan-World structures.

THANK YOU!

3 PhD positions related to photogrammetry and laserscanning

INACHUS

Technological and Methodological Solutions for Integrated Wide Area Situation Awareness and Survivor Localisation to Support Search and Rescue Teams.
20+ partners. Our task:
UAV-based photogrammetry and satellite remote sensing to support first responders in finding victims
Call for applications open till July 1st (one position)

Position estimation of mobile sensors using airborne imagery

Mitigating the well-known GPS outage problem in "urban canyons" for mobile mapping platforms through integrated position and attitude estimation using nadir and oblique airborne imagery. Close cooperation with leading industry partners like Cyclomedia, Fugro and Topcon Sokkia.

Two PhD positions, focusing on mobile laser scanning and mobile imagery, respectively.
Call for applications open soon, please watch www.itc.nl, or contact Markus Gerke.

SEVENTH FRAMEWORK PROGRAMME

STW Nieuwe technologie mogelijk maken

NWO Dutch Science Foundation

Contact: Markus Gerke, m.gerke@utwente.nl

DEMO – 1ST FLOOR

```
FreeCAD.getDocument("countRules_1stFloor").getObject("Box90").Placement = App.Placement(App.Vector(7.81687

FreeCAD.getDocument("countRules_1stFloor").addObject("Part::Box","Box91")
FreeCAD.getDocument("countRules_1stFloor").getObject("Box91").Height = 3.000000
FreeCAD.getDocument("countRules_1stFloor").getObject("Box91").Length = 0.400000
FreeCAD.getDocument("countRules_1stFloor").getObject("Box91").Width = 0.400000
FreeCAD.getDocument("countRules_1stFloor").getObject("Box91").Placement = App.Placement(App.Vector(9.81687

FreeCAD.getDocument("countRules_1stFloor").addObject("Part::Box","Box92")
FreeCAD.getDocument("countRules_1stFloor").getObject("Box92").Height = 3.000000
FreeCAD.getDocument("countRules_1stFloor").getObject("Box92").Length = 0.400000
FreeCAD.getDocument("countRules_1stFloor").getObject("Box92").Width = 0.400000
FreeCAD.getDocument("countRules_1stFloor").getObject("Box92").Placement = App.Placement(App.Vector(9.81687

FreeCAD.getDocument("countRules_1stFloor").addObject("Part::Box","Box93")
FreeCAD.getDocument("countRules_1stFloor").getObject("Box93").Height = 3.000000
FreeCAD.getDocument("countRules_1stFloor").getObject("Box93").Length = 0.400000
FreeCAD.getDocument("countRules_1stFloor").getObject("Box93").Width = 0.400000
FreeCAD.getDocument("countRules_1stFloor").getObject("Box93").Placement = App.Placement(App.Vector(9.81687

FreeCAD.getDocument("countRules_1stFloor").addObject("Part::Box","Box94")
FreeCAD.getDocument("countRules_1stFloor").getObject("Box94").Height = 3.000000
FreeCAD.getDocument("countRules_1stFloor").getObject("Box94").Length = 0.400000
FreeCAD.getDocument("countRules_1stFloor").getObject("Box94").Width = 0.400000
FreeCAD.getDocument("countRules_1stFloor").getObject("Box94").Placement = App.Placement(App.Vector(3.81687

FreeCAD.getDocument("countRules_1stFloor").addObject("Part::MultiFuse","Box95")
```

Rule 1: place cuboids