FACULTY OF SCIENCE
PAVOL JOZEF ŠAFÁRIK UNIVERSITY
IN KOŠICE

ics.upjs.sk/indora

PAVOL JOZEF ŠAFÁRIK
UNIVERSITY
IN KOŠICE

Map Model Extraction from Image Floor Plans

Miroslav Opiela, Martina Hrehová, František Galčík

miroslav.opiela@upjs.sk

Institute of Computer Science, P. J. Šafárik University, Faculty of Science, Košice, Slovakia

FACULTY OF SCIENCE
PAVOL JOZEF ŠAFÁRIK UNIVERSITY
IN KOŠICE

ics.upjs.sk/indora

PAVOL JOZEF ŠAFÁRIK
UNIVERSITY
IN KOŠICE

Map Model Extraction from Image Floor Plans

Miroslav Opiela, Martina Hrehová, František Galčík

miroslav.opiela@upjs.sk

Institute of Computer Science, P. J. Šafárik University, Faculty of Science, Košice, Slovakia

Floor plan input

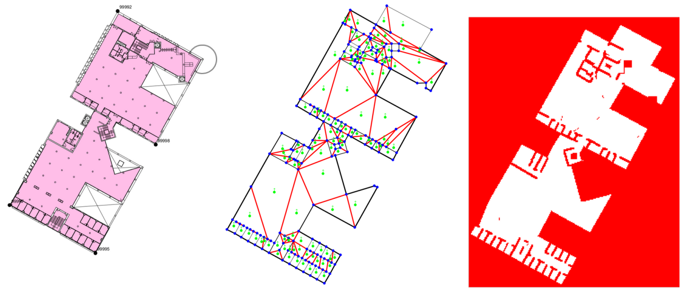
- raster image
- map scale and rotation
- georeferenced position of a specified point

Vector model

- annotated map
- points, lines, polygons
- convex zones requirement for automatic processing

Grid output

- binary image
- accessible and inaccessible positions

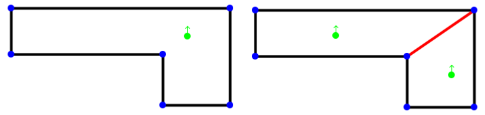


Library building, IPIN 2020 competition, Castellón de la Plana, Spain

Convex zones

- Zones in form of convex polygons.
- **Essential condition** for automatic extraction of vector model

*A **polygon** is **convex** if any line segment between two arbitrary points inside polygon is inside this polygon as well. Moreover, internal angles do not exceed 180°.*



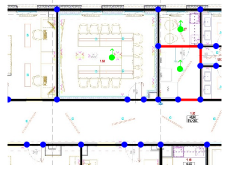
Point types

- **Zone point** - inside zone, includes semantical information about the zone
- ● **Point** - endpoint of connections

Connection types

- ▬ **Solid** - walls
- ▬ **Transitional** - doors
- 🟥 **Transparent** - artificial border for separating zones

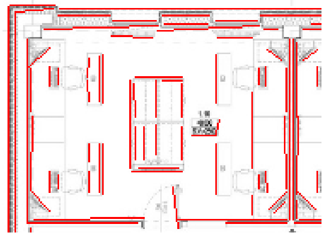
I. Map Annotation



- Custom Java application (similar to GIS software).
- **Manual** time-consuming **process**. May be replaced by computer vision approach.
- Annotation over **floor plan background image.**
- Automatic zone convexity checker
- Semantic representation:
  - **zones** as structural units
  - **rooms** as semantical units
  - A room is a collection of zones.

Computer vision approach for walls annotation

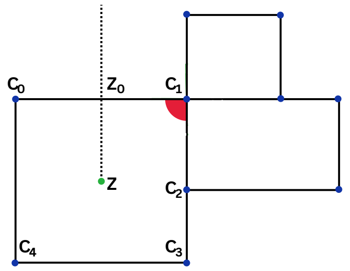
1. **Text is removed** from image using OCR (Optical Character Recognition).
2. LSD (**Line Segment Algorithm**) finds lines.
3. **Eliminate duplicate lines** - endpoints of detected lines are clustered using mean shift algorithm and replaced by respective cluster positions.
4. **Points alignment** using mean shift algorithm separately on each dimension.



Additional manual adjustment may be needed to acquire accurate annotation.

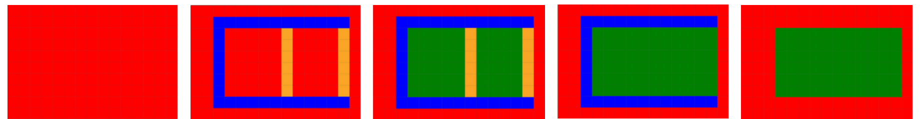
II. from **annotated convex zones** to **vector model**

1. Construct vertical line from zone point **Z**.
2. Find point $\mathbf{Z_0}$as intersection between existing connection and vertical line. The distance between $\mathbf{ZZ_0}$is minimal. Denote the line as $\mathbf{C_0C_1}$.
3. Next point $\mathbf{C_{i+1}}$: connection $\mathbf{C_iC_X}$ with minimal angle $\mathbf{C_{i-1}C_i\,C_X}$ is selected as $\mathbf{C_iC_{i+1}}$.
4. Repeat previous step until $\mathbf{C_{n-1}C_0}$ is found.



III. from **vector model** to **grid**

1. Set grid scale based on size (e.g. using 33x33cm per a grid cell) and fill two-dimensional array with **(1) inaccessible value**
2. Iterate over **all connections**. Calculate grid cell positions. Apply **line drawing algorithm**. Put **(2)** close value for walls or (3) open value for doors and transparent connections.
3. Iterate over **all zones**. Apply **flood-fill algorithm** (BFS) from zone point to fill area bound by **(2)** and **(3)** values. Insert **(0) accessible value.**
4. Change (3) to **(0).**
5. Change **(2)** to **(1).**



Evaluation

- IPIN competitions buildings:
  - IPIN 2018, shopping mall, 4 floors
  - IPIN 2019, research institute, 3 floors
  - IPIN 2020, library, 5 floors
- Faculty building, 5 floors/areas
- Computer vision approach evaluated also on datasets:
  - ROBIN
  - CubiCasa5k
  - CVC-FP
- Single floor IPIN 2019 competition building:
  - more than 800 lines
  - **40 minutes** manual annotation
  - **5 minutes** computer vision approach + manual adjustment

Floor plan input

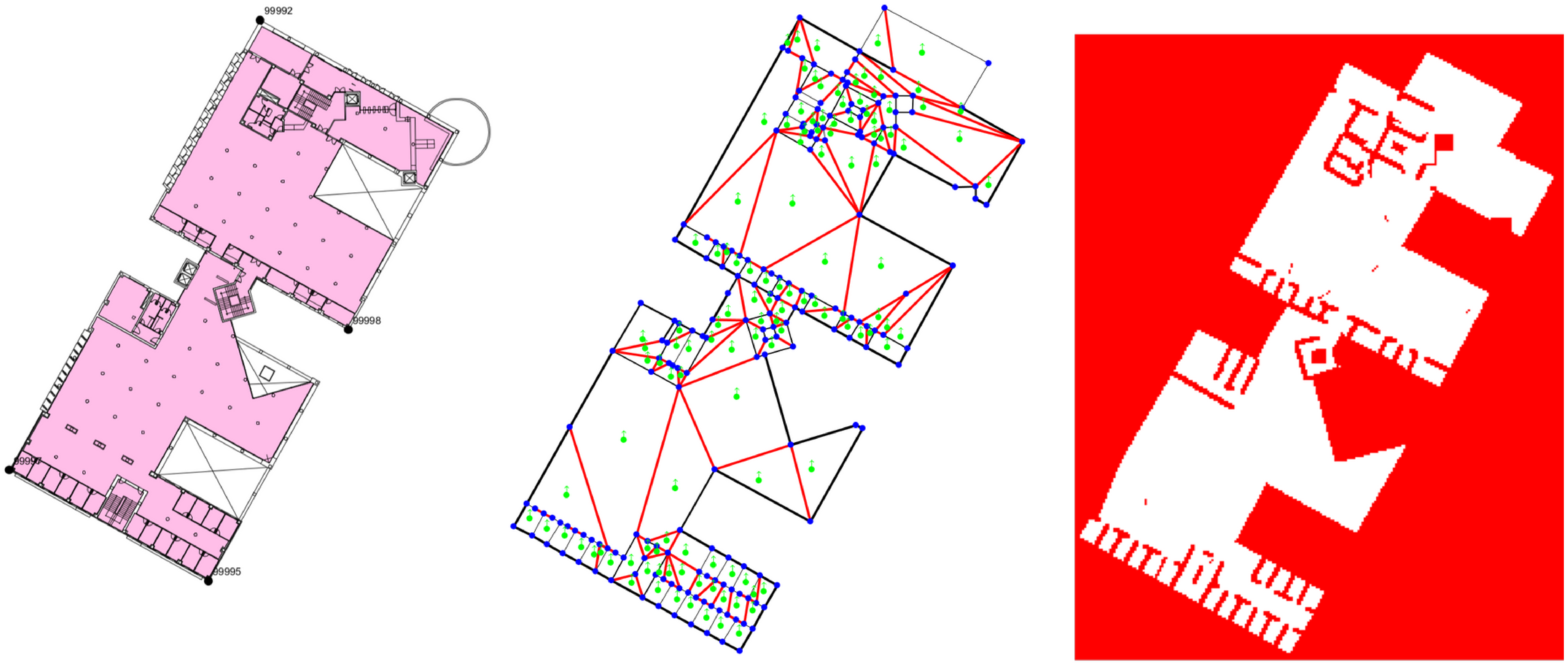
- raster image
- map scale and rotation
- georeferenced position of a specified point

Vector model

- annotated map
- points, lines, polygons
- convex zones requirement for automatic processing

Grid output

- binary image
- accessible and inaccessible positions

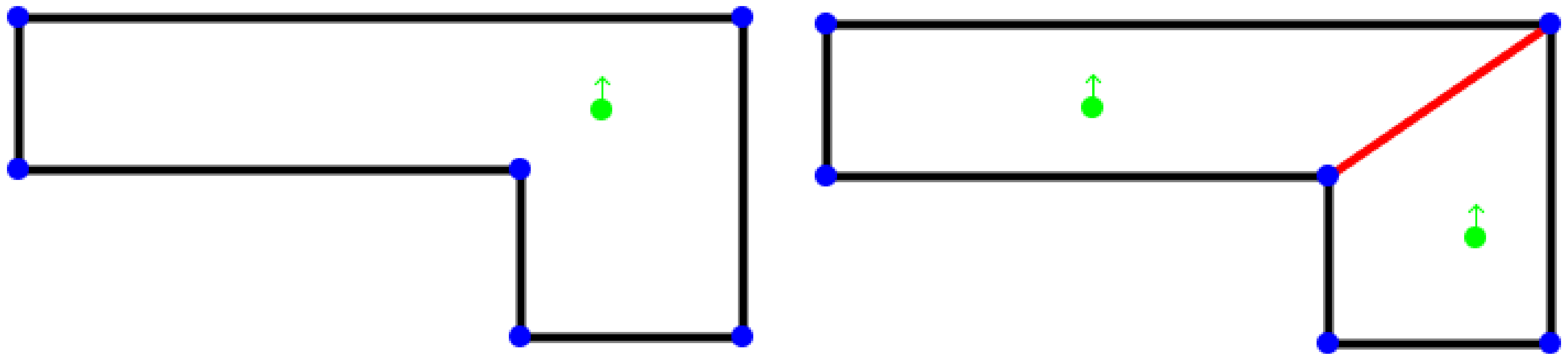


Library building, IPIN 2020 competition, Castellón de la Plana, Spain

Convex zones

- Zones in form of convex polygons.
- **Essential condition** for automatic extraction of vector model

*A **polygon** is **convex** if any line segment between two arbitrary points inside polygon is inside this polygon as well. Moreover, internal angles do not exceed 180°.*



Point types

- ● **Zone point** - inside zone, includes semantical information about the zone
- **Point** - endpoint of connections

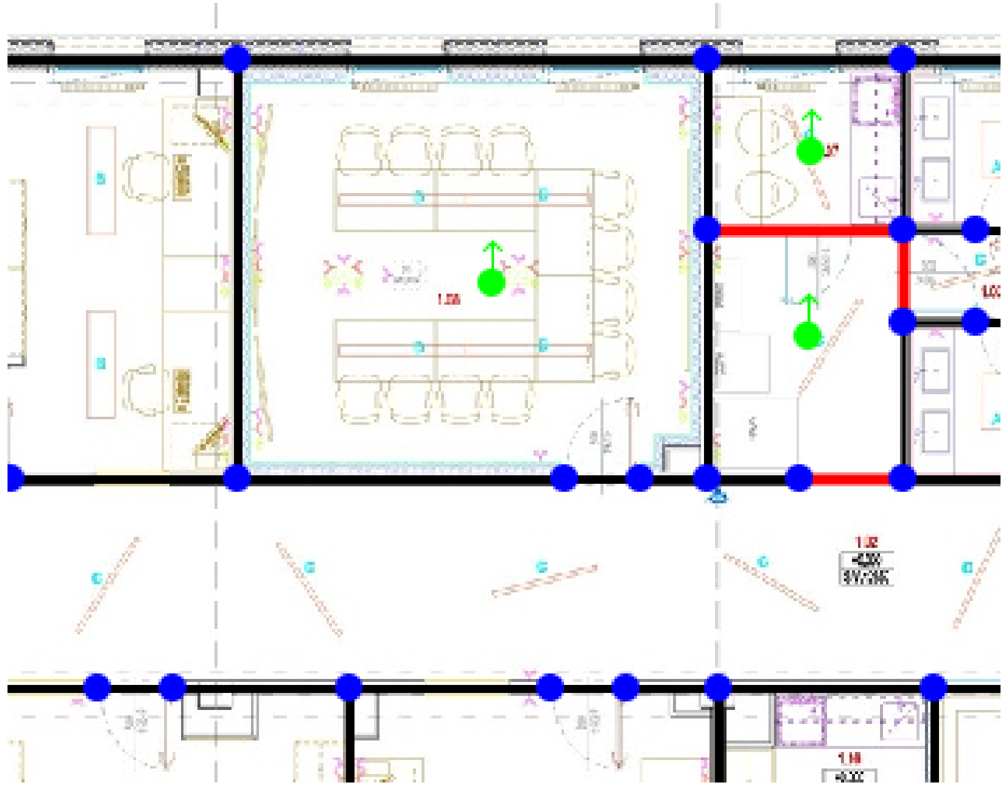
Connection types

▬ **Solid** - walls

▬ **Transitional** - doors

- **Transparent** - artificial border for separating zones

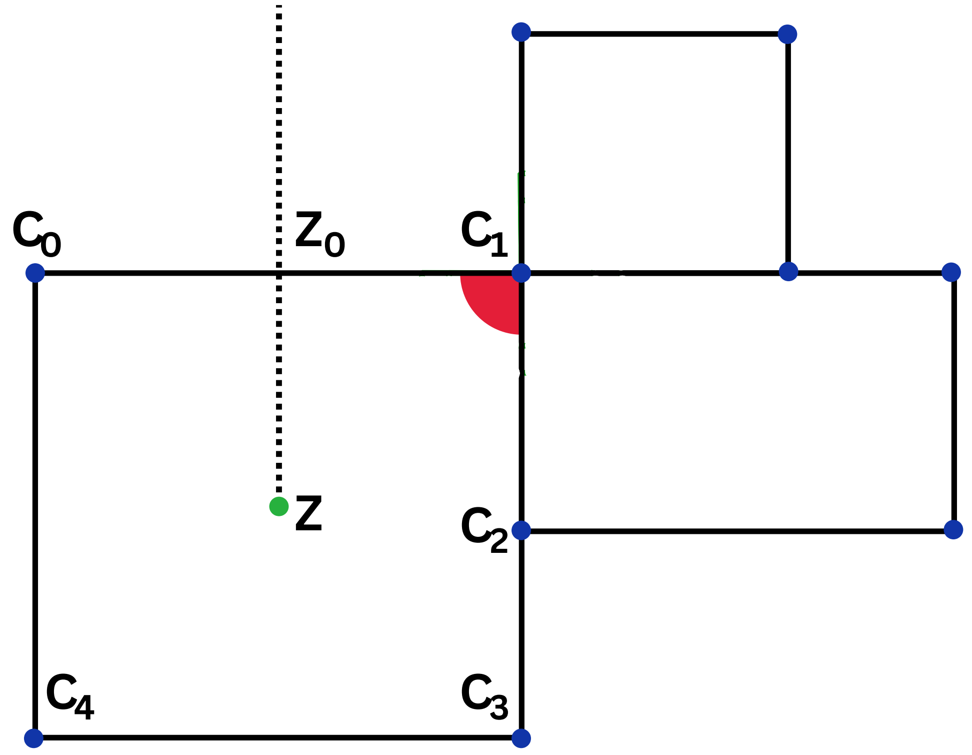
I. Map Annotation



- Custom Java application (similar to GIS software).
- **Manual** time-consuming **process**. May be replaced by computer vision approach.
- Annotation over **floor plan background image.**
- Automatic zone convexity checker
- Semantic representation:
    - **zones** as structural units
    - **rooms** as semantical units
    - A room is a collection of zones.

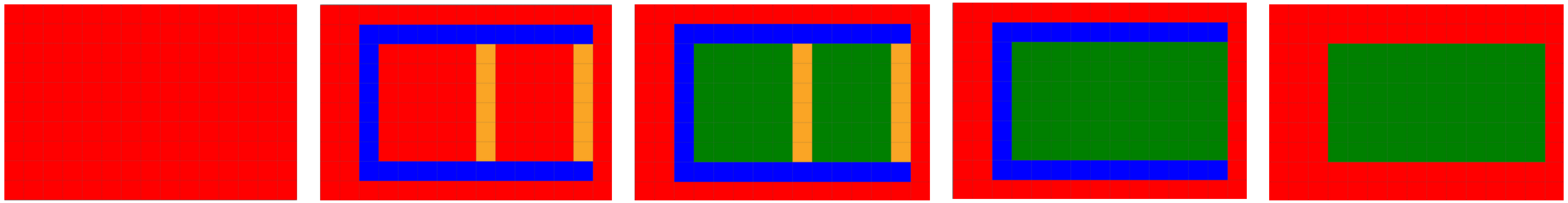
II. from **annotated convex zones** to **vector model**

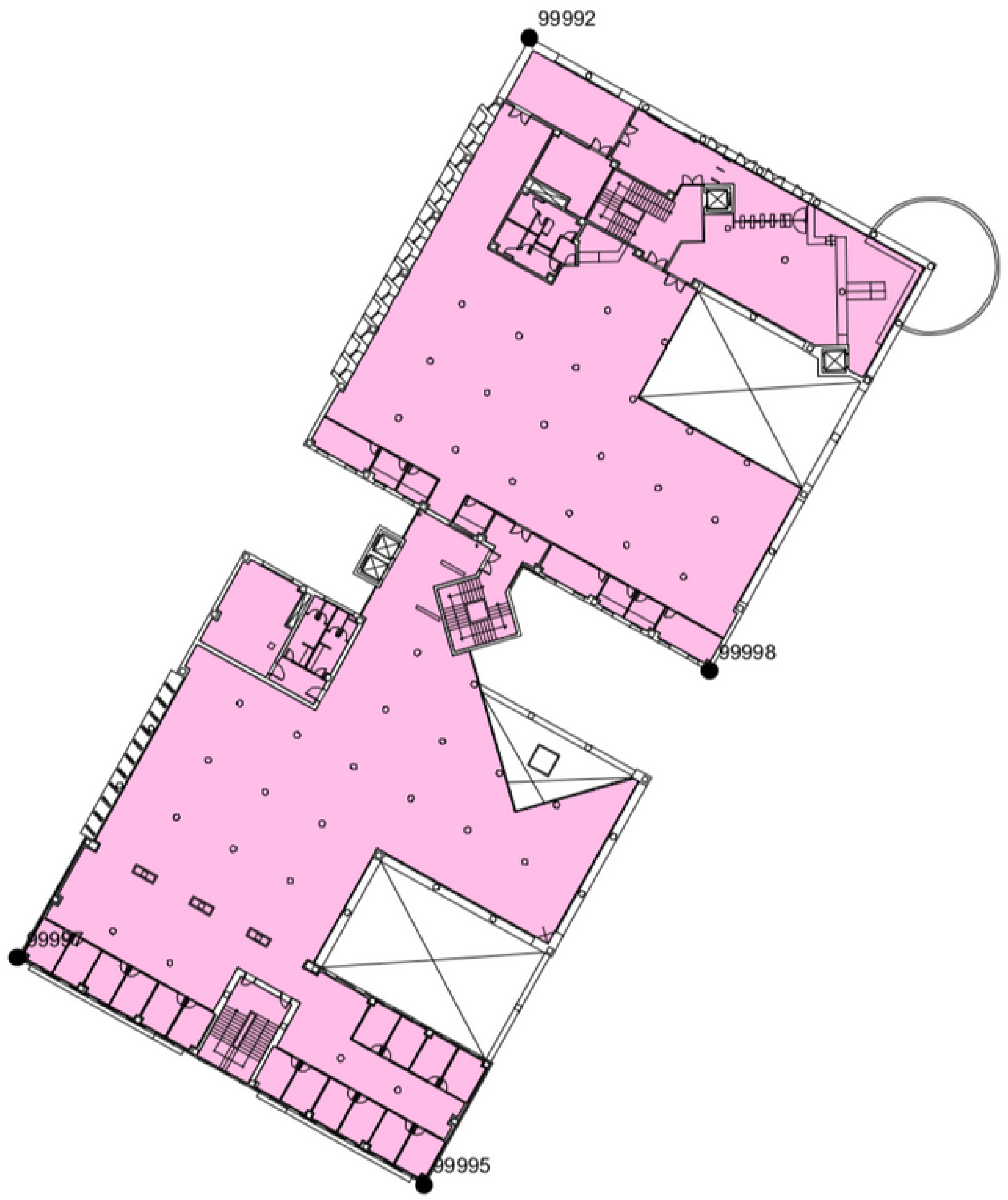
1. Construct vertical line from zone point **Z**.
2. Find point $\mathbf{Z_0}$as intersection between existing connection and vertical line. The distance between $\mathbf{ZZ_0}$is minimal. Denote the line as $\mathbf{C_0C_1}$.
3. Next point $\mathbf{C_{i+1}}$: connection $\mathbf{C_iC_X}$ with minimal angle $\mathbf{C_{i-1}C_iC_X}$ is selected as $\mathbf{C_iC_{i+1}}$.
4. Repeat previous step until $\mathbf{C_{n-1}C_0}$ is found.

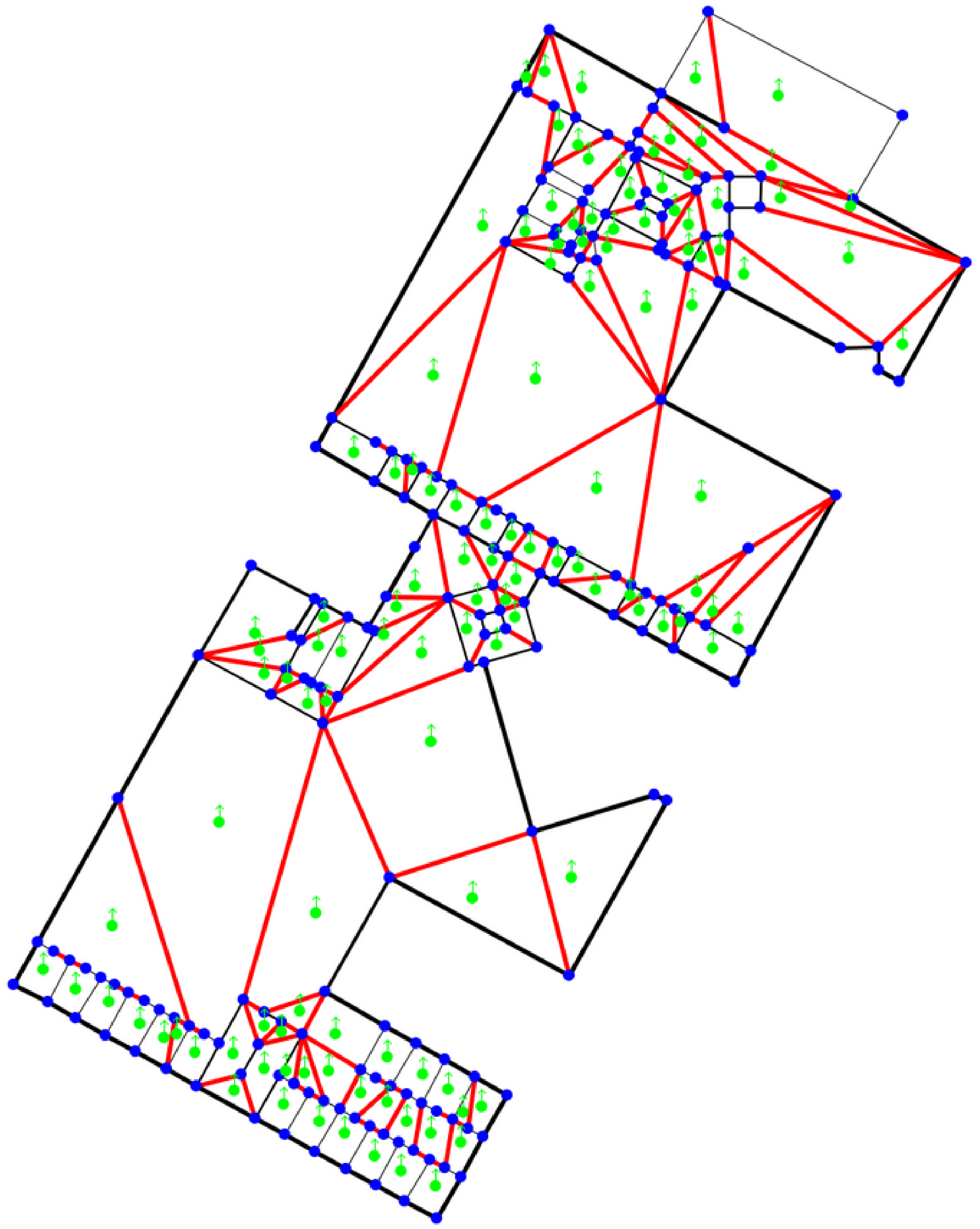


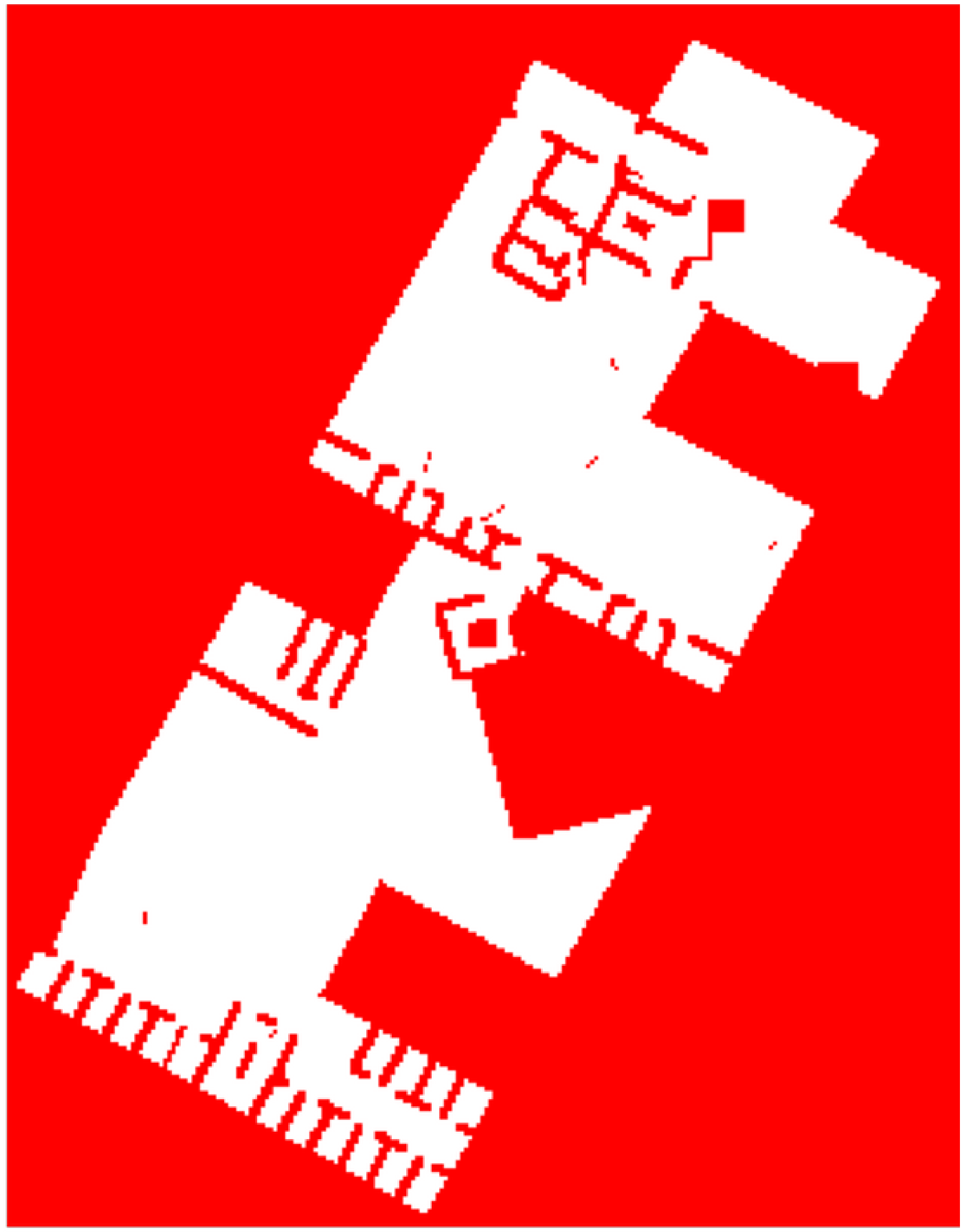
III. from **vector model** to **grid**

1. Set grid scale based on size (e.g. using 33x33cm per a grid cell) and fill two-dimensional array with **(1) inaccessible value**
2. Iterate over **all connections**. Calculate grid cell positions. Apply **line drawing algorithm**. Put **(2) close value** for walls or **(3) open value** for doors and transparent connections.
3. Iterate over **all zones**. Apply **flood-fill algorithm** (BFS) from zone point to fill area bound by **(2)** and (3) values. Insert **(0) accessible value.**
4. Change (3) to **(0).**
5. Change **(2)** to **(1).**



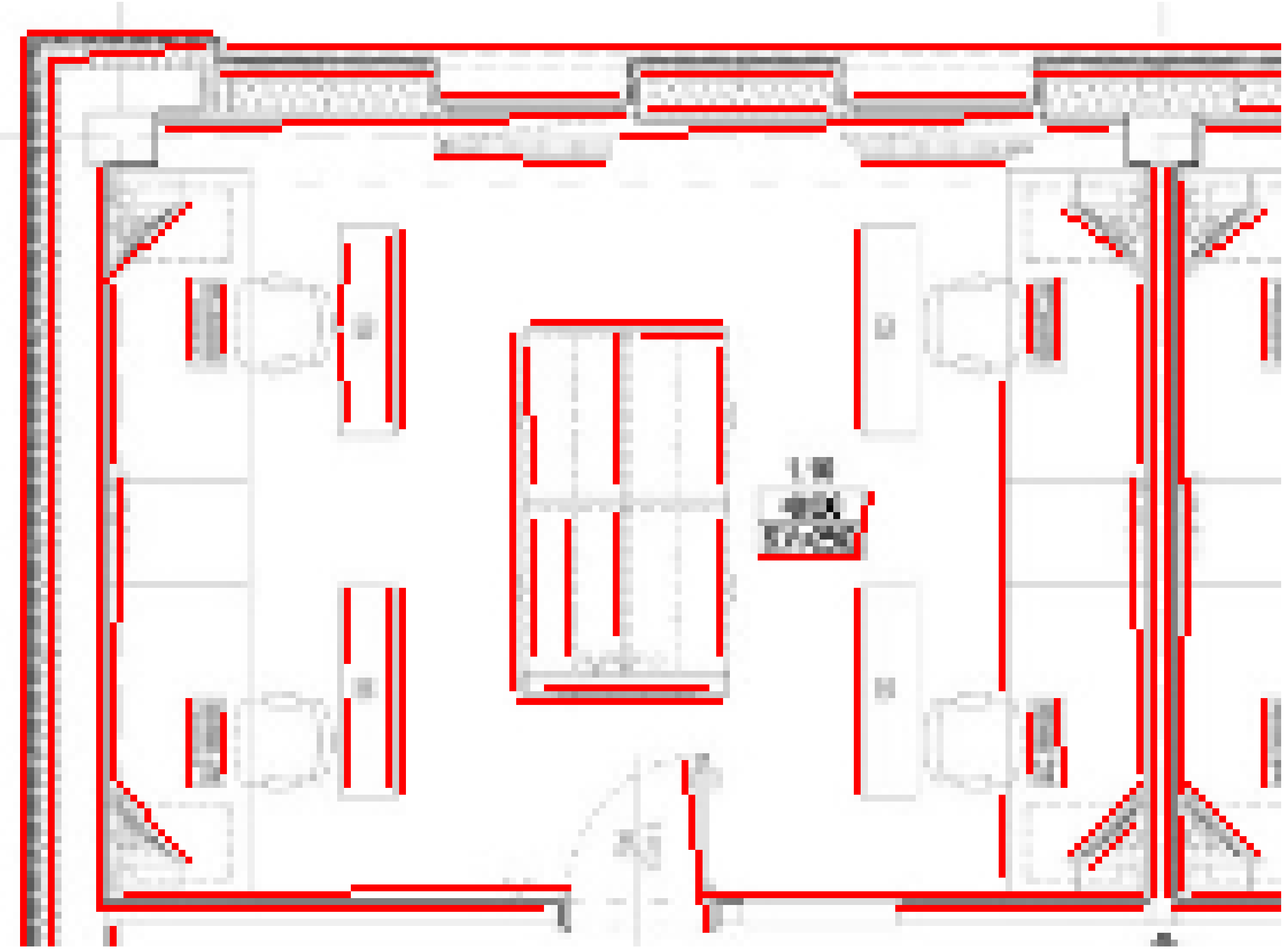

99992
99998
99997
99995





Computer vision approach for walls annotation

1. **Text is removed** from image using OCR (Optical Character Recognition).
2. LSD (**Line Segment Algorithm**) finds lines.
3. **Eliminate duplicate lines** - endpoints of detected lines are clustered using mean shift algorithm and replaced by respective cluster positions.
4. **Points alignment** using mean shift algorithm separately on each dimension.



Additional manual adjustment may be needed to acquire accurate annotation.

Evaluation

- IPIN competitions buildings:
    - IPIN 2018, shopping mall, 4 floors
    - IPIN 2019, research institute, 3 floors
    - IPIN 2020, library, 5 floors
- Faculty building, 5 floors/areas
- Computer vision approach evaluated also on datasets:
    - ROBIN
    - CubiCasa5k
    - CVC-FP
- Single floor IPIN 2019 competition building:
    - more than 800 lines
    - **40 minutes** manual annotation
    - **5 minutes** computer vision approach + manual adjustment

Thank you for your attention.