



Line detection

Computer vision - project part 1

Team 4

November 8, 2019

University of Liège



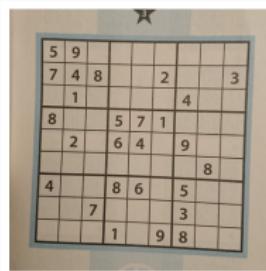
Introduction

- Detect the main **straight** lines of an image



Introduction

- Detect the main **straight** lines of an image
- Sudoku, soccer and road images



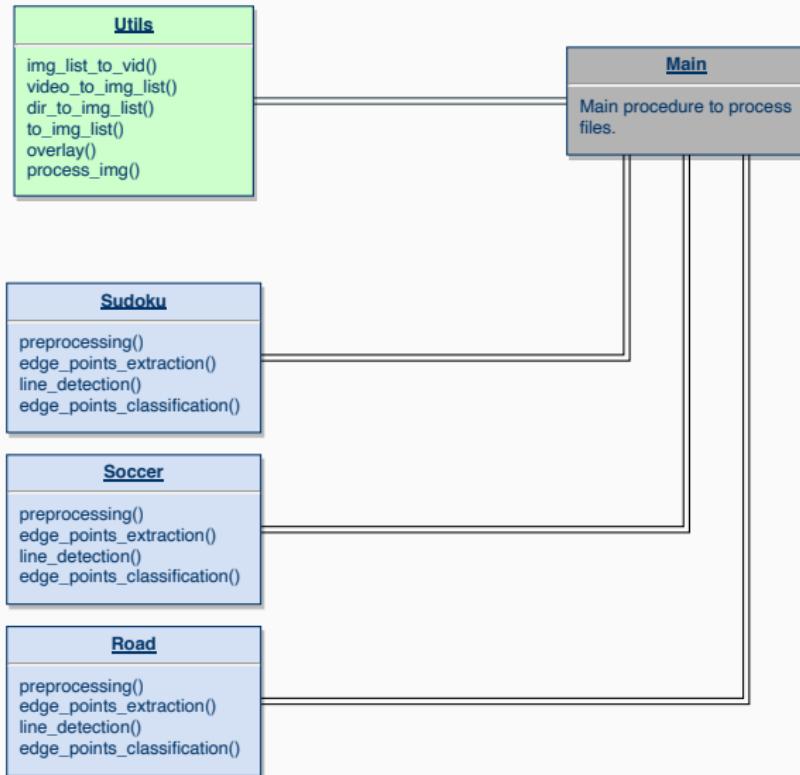
Plan of the presentation



1. Algorithms and main methods
2. Sudoku images
3. Soccer images
4. Road images
5. Other types of images
6. Conclusion

Algorithms and main methods

Algorithms - Structure of our program





Main methods used

Canny's algorithm for **edge extraction**

- Robustness
- Not very sensitive to noise
- Adaptable



Main methods used

Probabilistic Hough transform for **line detection**

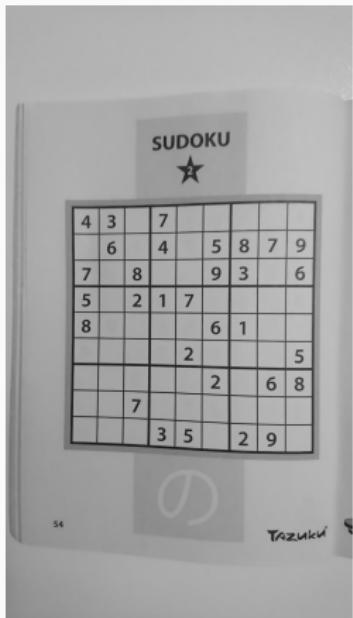
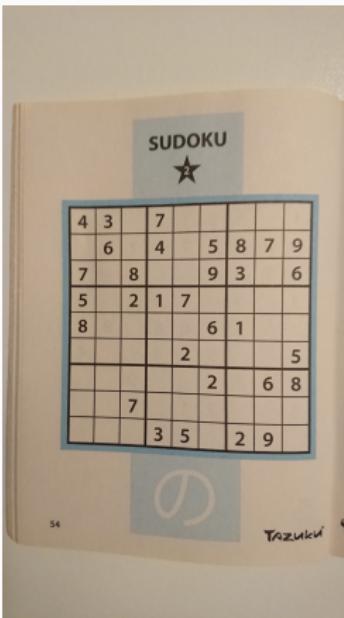
- Powerful
- Robust to noise
- React pretty well to gaps

Sudoku images



Sudoku procedure

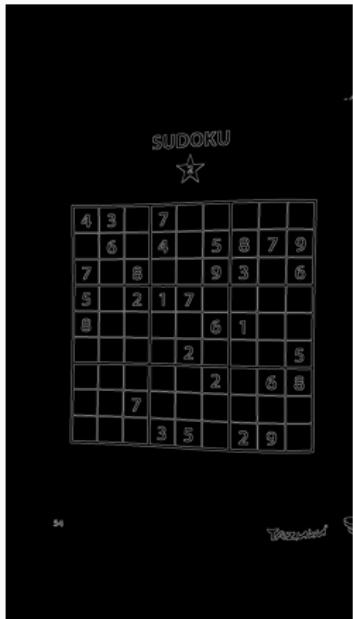
1. Bilateral filter





Sudoku procedure

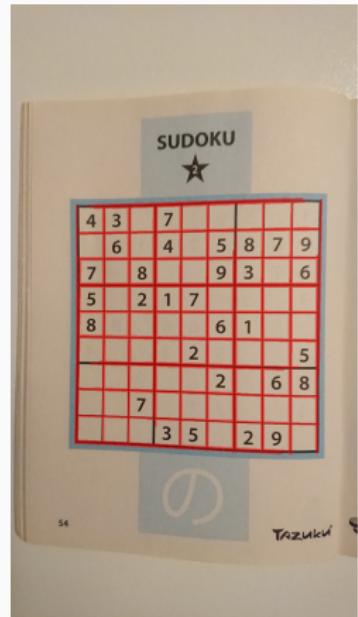
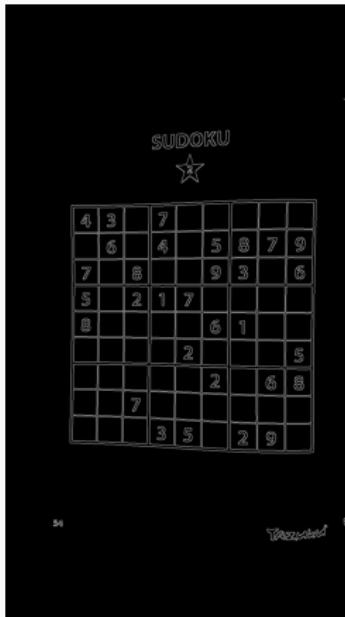
1. Bilateral filter
2. Canny edge detection





Sudoku procedure

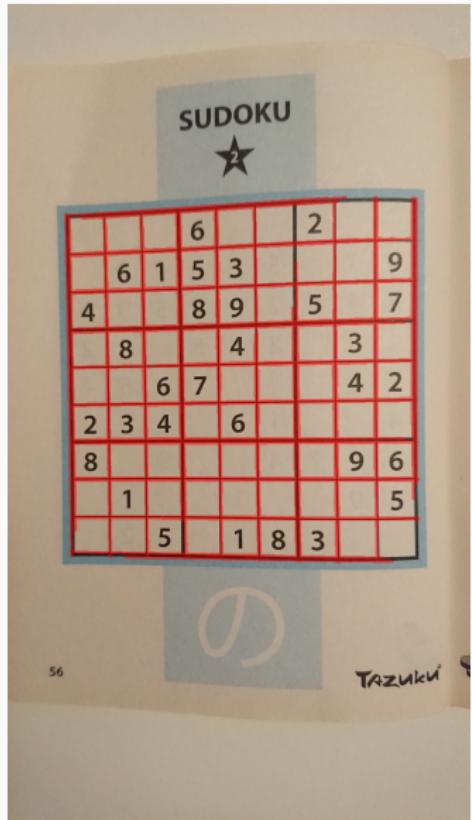
1. Bilateral filter
2. Canny edge detection
3. Probabilistic Hough transform, searching for **long continuous lines**



Sudoku drawbacks



Trouble to recognize curved lines.



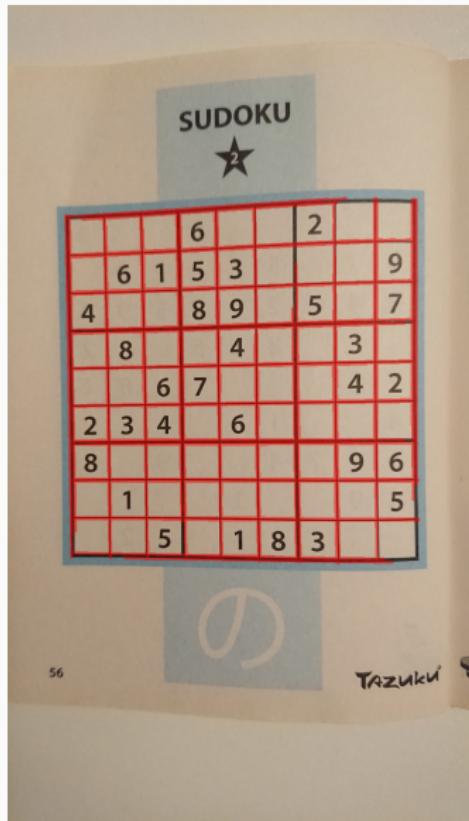
Sudoku drawbacks



Trouble to recognize curved lines.

Solutions

- Post-processing lines



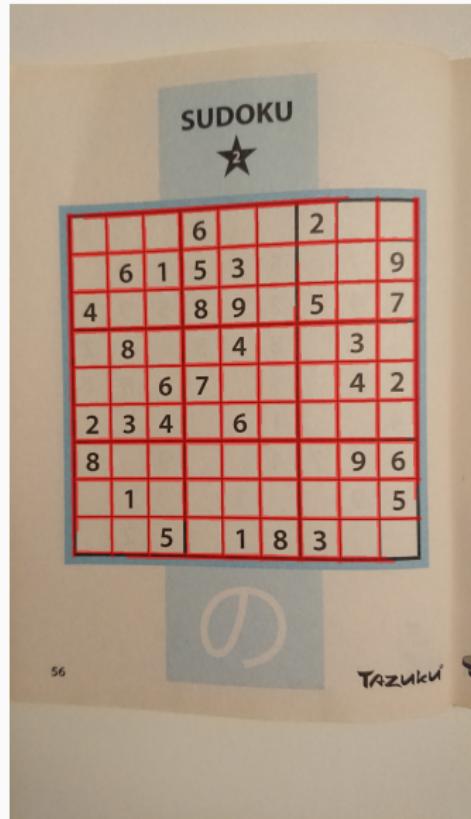
Sudoku drawbacks



Trouble to recognize curved lines.

Solutions

- Post-processing lines
- Improving **acquisition**

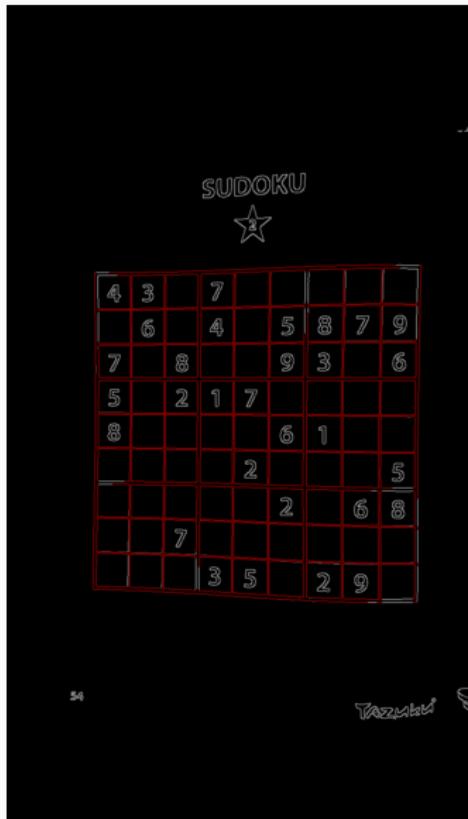


Edges classification



Same procedure for each type of images.

1. Detected lines are traced in white on black image
2. Edges belonging to white zone are selected
3. Selected edges are colored in red



Soccer images

Soccer procedure - First try



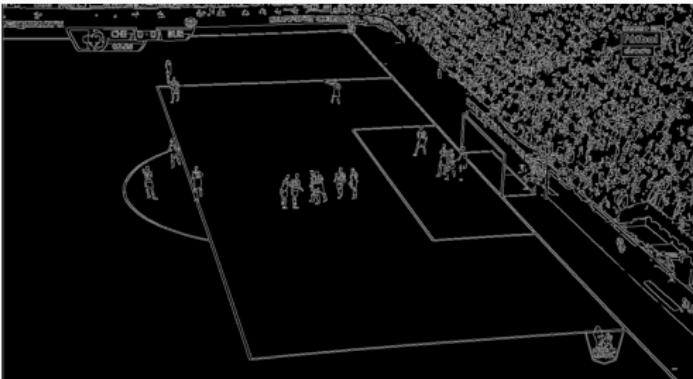
1. Bilateral filter



Soccer procedure - First try



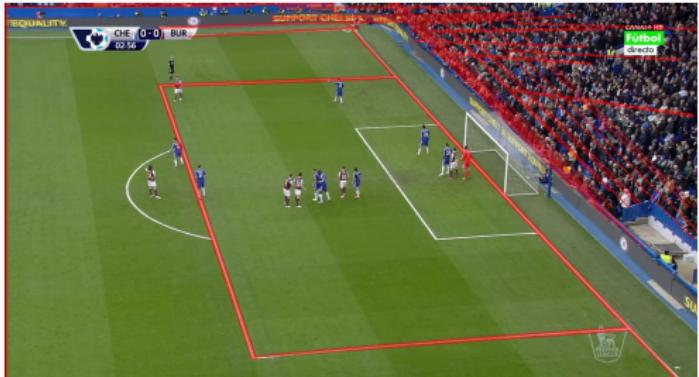
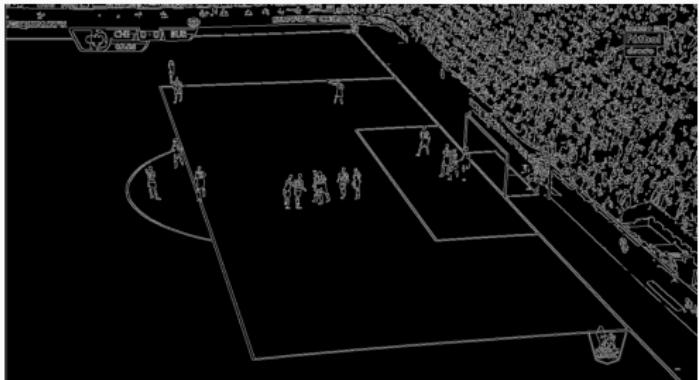
1. Bilateral filter
2. Canny edge detection



Soccer procedure - First try



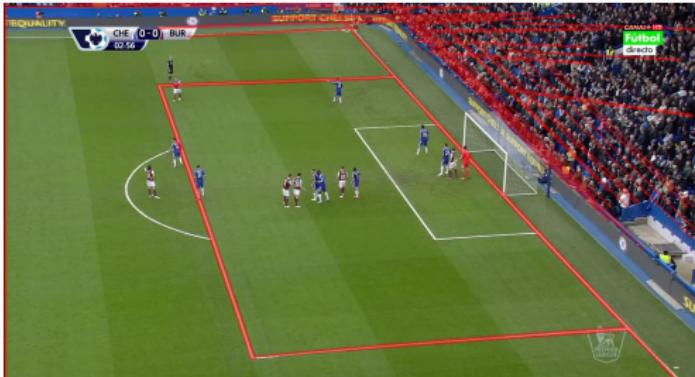
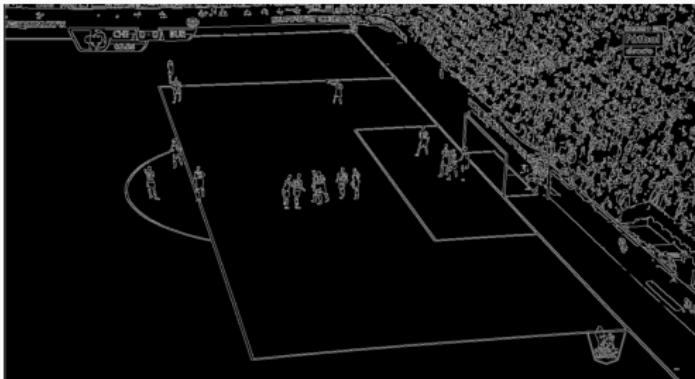
1. Bilateral filter
2. Canny edge detection
3. Probabilistic Hough transform, searching for long lines



Soccer procedure - First try



1. Bilateral filter
2. Canny edge detection
3. Probabilistic Hough transform, searching for long lines



Soccer procedure - Second try



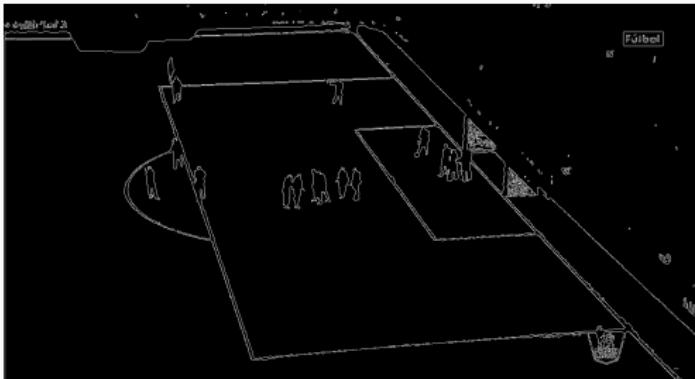
1. Hue filter



Soccer procedure - Second try



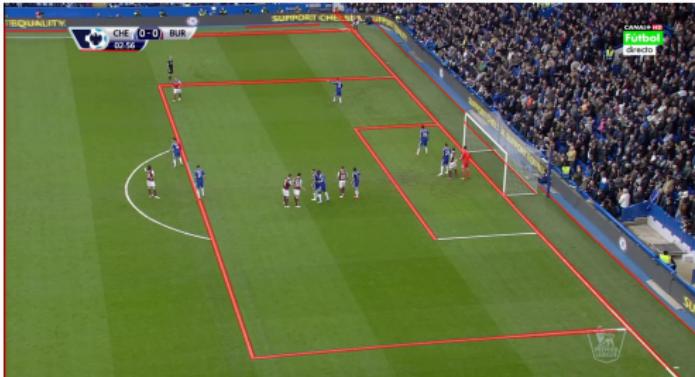
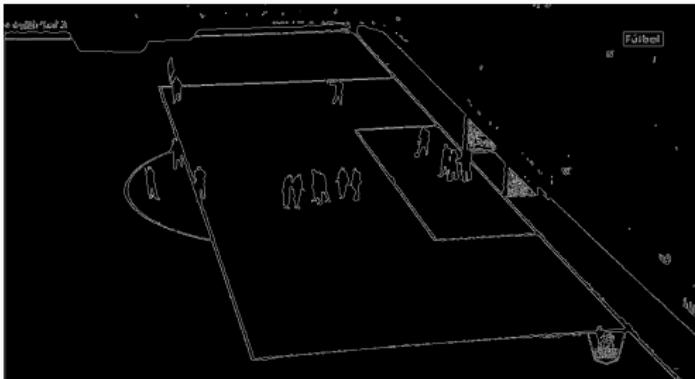
1. Hue filter
2. Canny edge detection





Soccer procedure - Second try

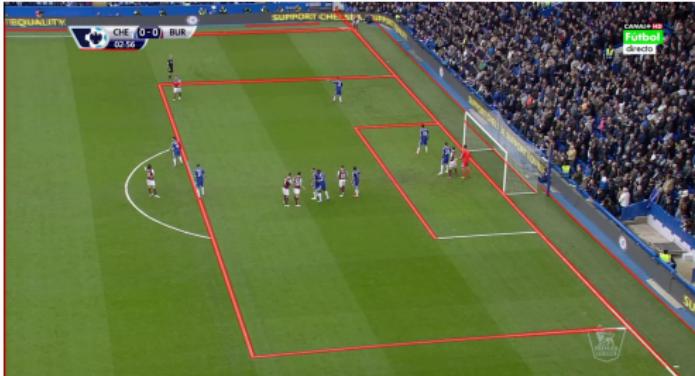
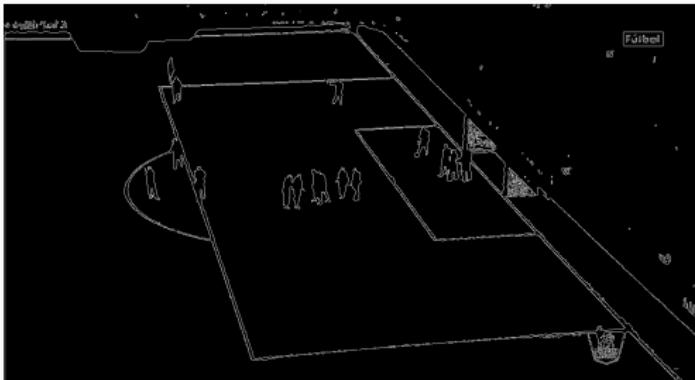
1. Hue filter
2. Canny edge detection
3. Probabilistic Hough transform, searching for long lines





Soccer procedure - Second try

1. Hue filter
2. Canny edge detection
3. Probabilistic Hough transform, searching for long lines



Road images

Road procedure



1. Filter the white components



Road procedure



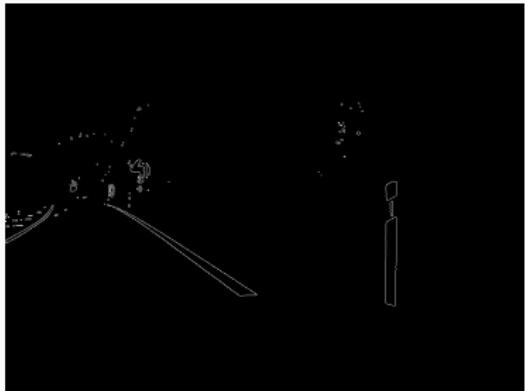
1. Filter the white components
2. Remove big white areas
(findContours)



Road procedure



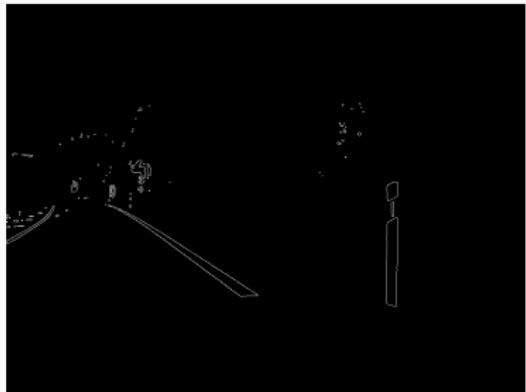
1. Filter the white components
2. Remove big white areas
(findContours)
3. Canny edge detection



Road procedure



1. Filter the white components
2. Remove big white areas
(findContours)
3. Canny edge detection
4. Probabilistic Hough



Road discussion



Road discussion - Imperfect algorithm





Road discussion

Why we had worse results than with the other classes :

- Images are more diverse
- Camera angle not standardized
- Lanes may curve
- Faded lines
- Overexposure
- ...



Road discussion

How to get better results ? → Acquisition step ! (not possible here)

- Standardized angle and position of the camera
- Region of Interest
- Tweak the parameters better

Other types of images

Building as soccer



PCB as road



Building as sudoku

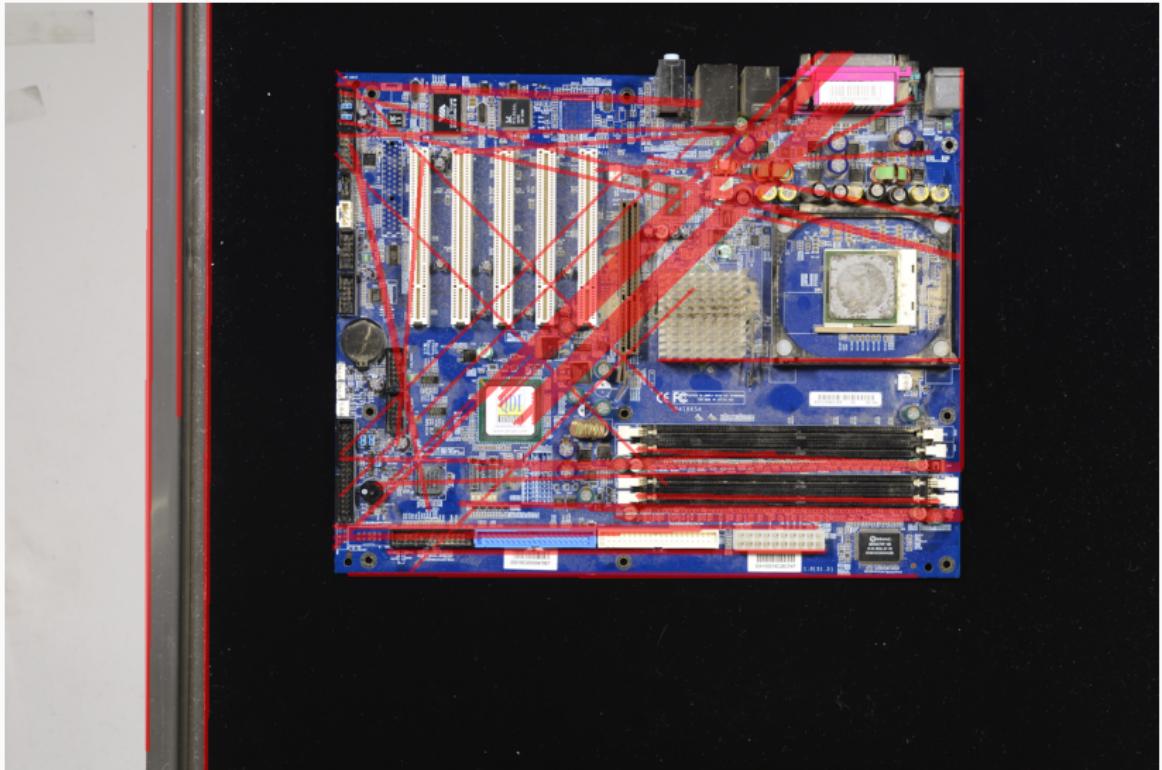


Building as sudoku



Sudoku's class is suitable for **grid-like** objects.

PCB as sudoku



Conclusion

There is no perfect algorithm, only acceptable solutions for particular problems.

Demonstrations