Shape Categorization

Computer vision project

Senthilkumar Sockalingam Kathiresan & Yannick Weitz

January 12, 2018

Task description

Find the objects and the orientation that fit into the cavity tiles placed in the PPT platform.

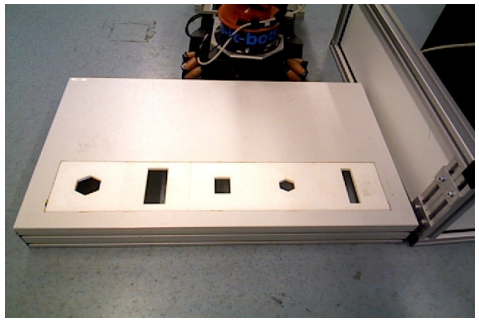
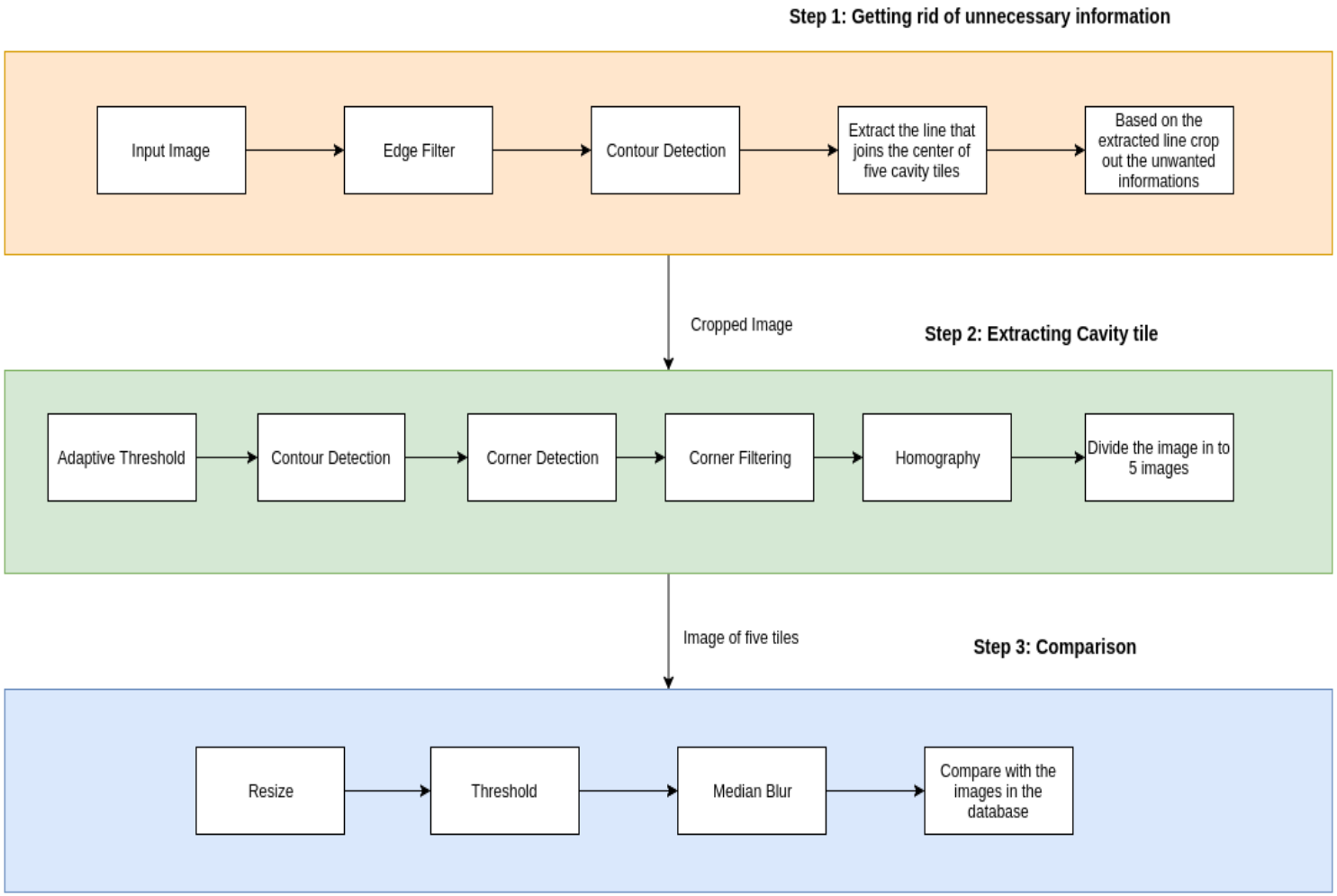


Figure 1 : Left: Precision Placement Test Platform. Right: Objects that need to be placed in the cavities.

Assumptions

1. All cavities and corners are visible
2. There is always one object that fits in the cavity tile.
3. The YouBot is standing in front of the table and provides an almost horizontal image
4. The tiles properties and parts are defined by the robocup@work rules

Flow Chart


Step 1: Getting rid of unnecessary information
Input Image
Edge Filter
Contour Detection
Extract the line that joins the center of five cavity tiles
Based on the extracted line crop out the unwanted informations
Cropped Image
Step 2: Extracting Cavity tile
Adaptive Threshold
Contour Detection
Corner Detection
Corner Filtering
Homography
Divide the image in to 5 images
Image of five tiles
Step 3: Comparison
Resize
Threshold
Median Blur
Compare with the images in the database

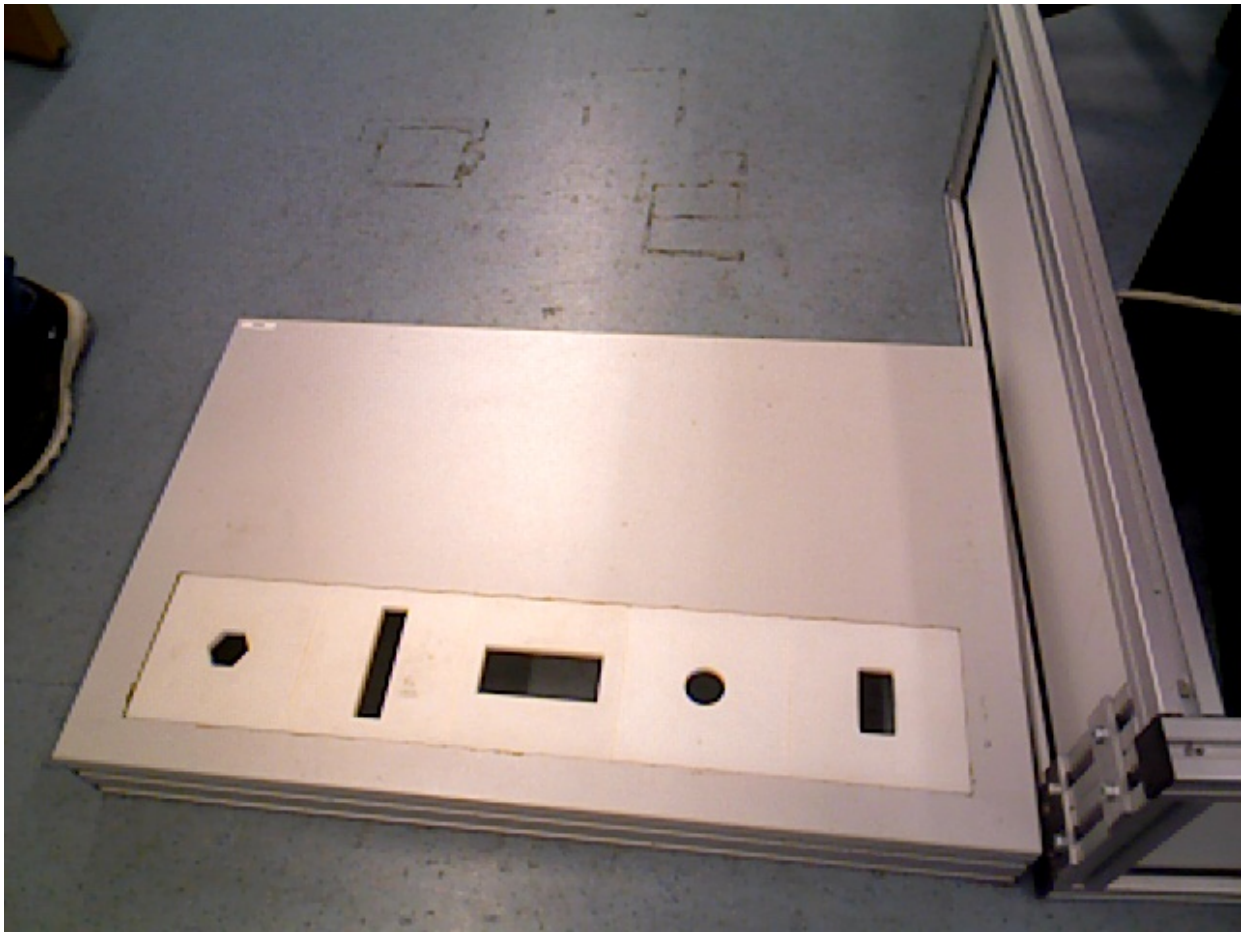


Figure 2 : Original image of 5 tiles like the camera would percept it.

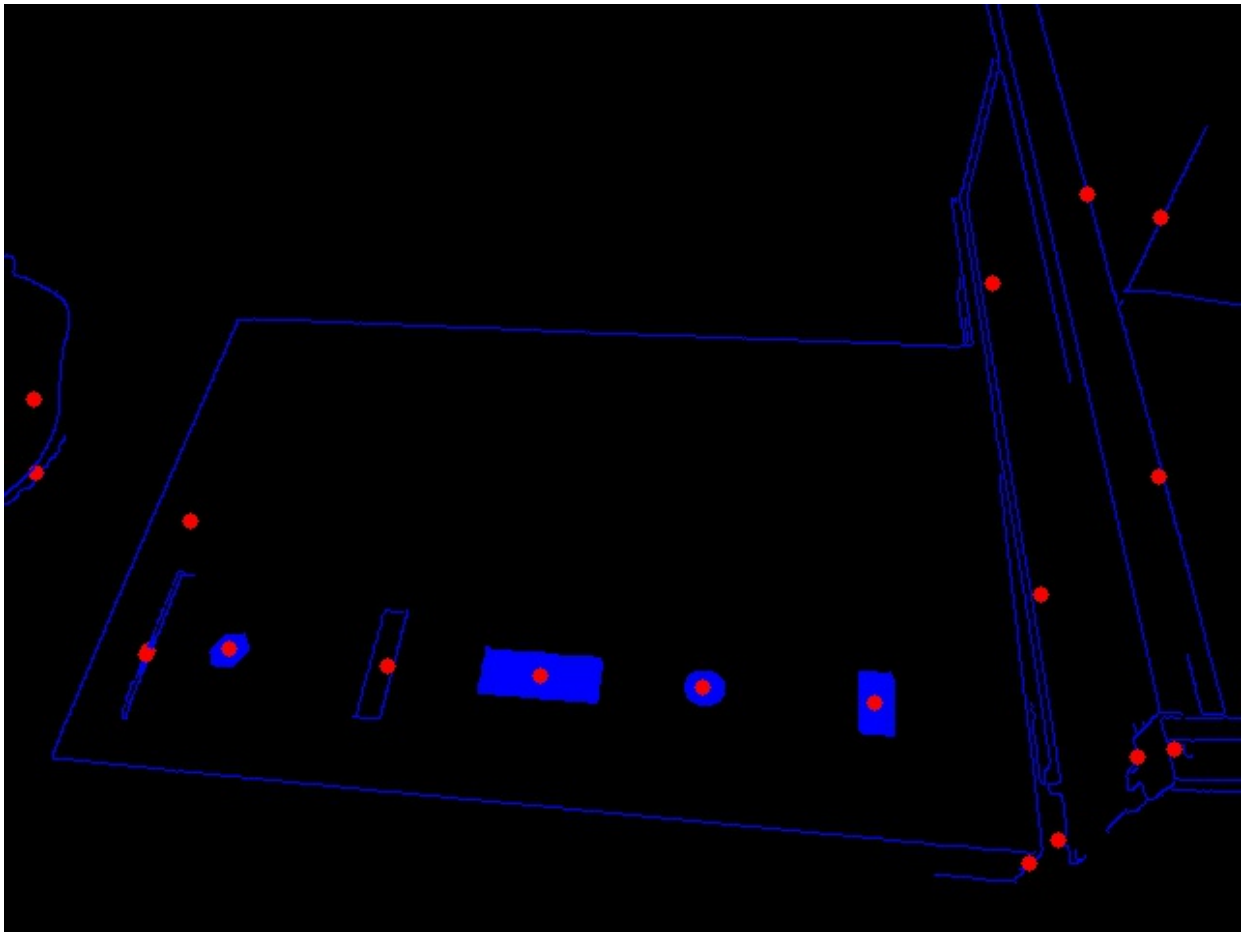


Figure 3 : The found contours after applying the findContour algorithm.

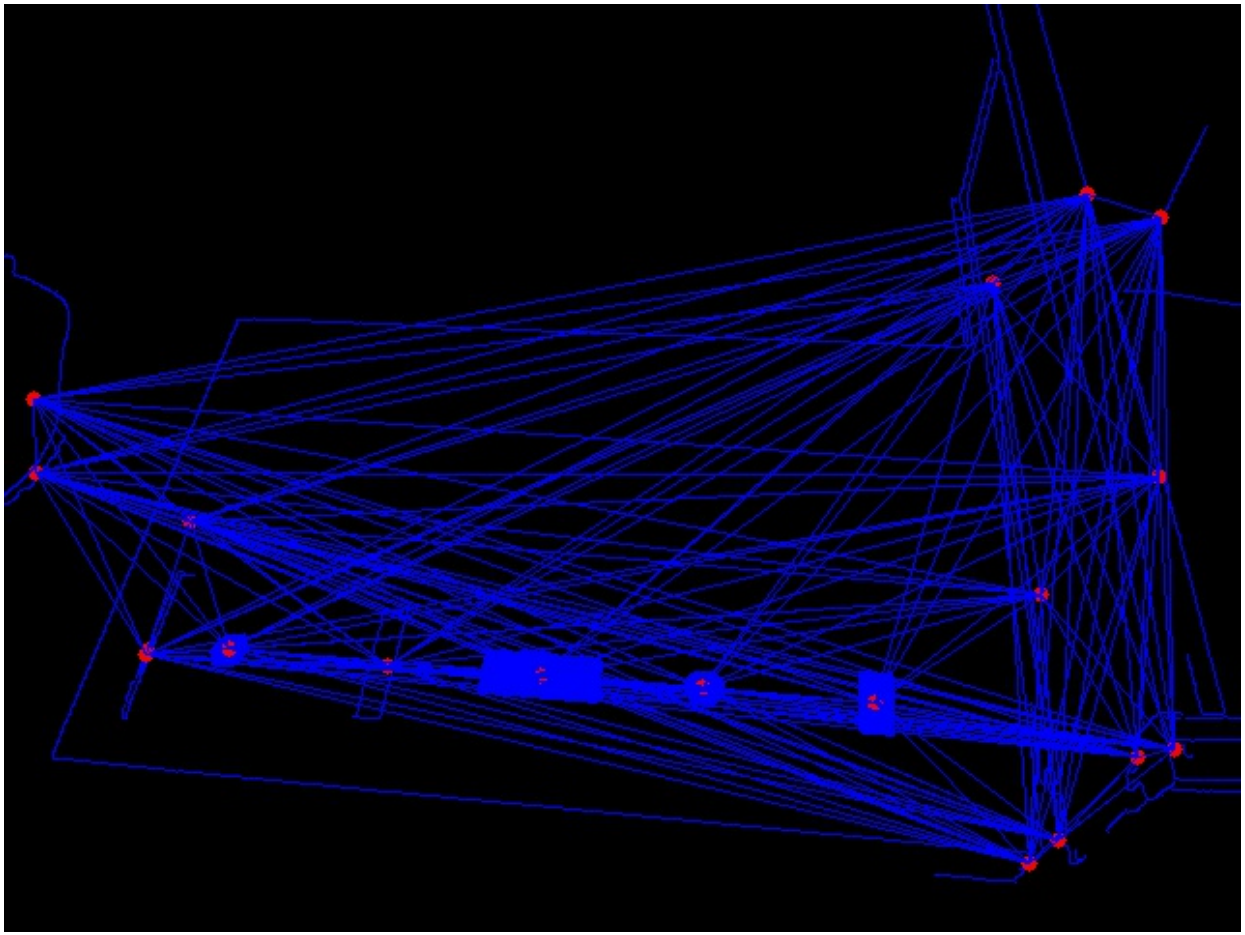


Figure 4 : All lines between the centers of the contours.

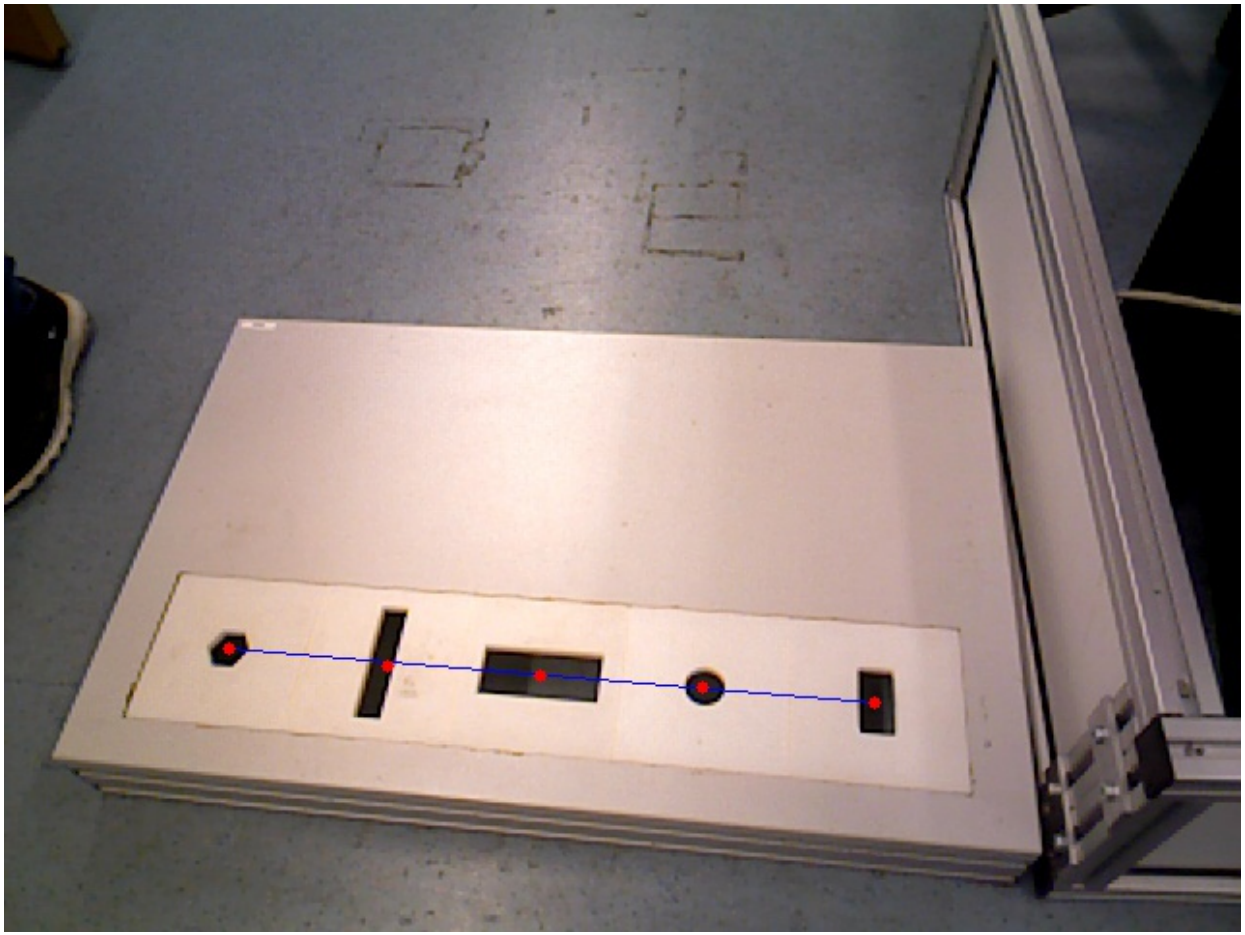


Figure 5 : The final line that corresponds to the least error.

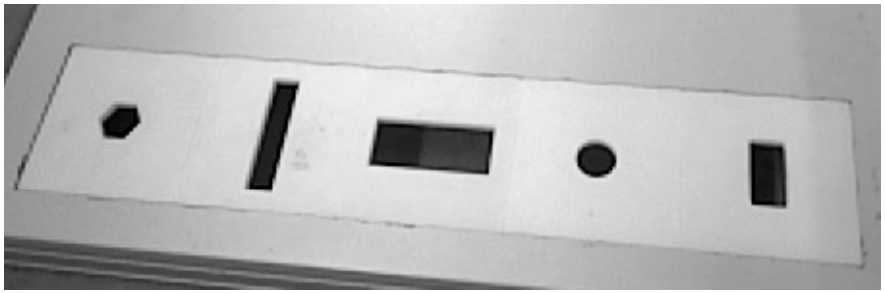


Figure 6 : The cropped image without less unnecessary information.

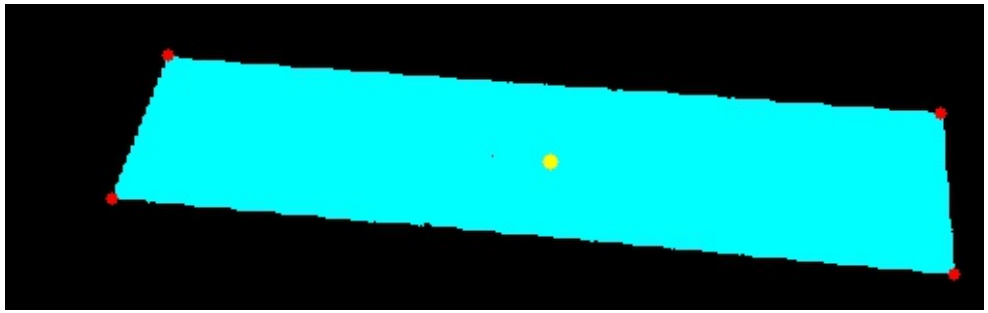


Figure 7 : The contour of the tile after applying an adaptive threshold and finding the corners of the tile using Harris Corner Detection.

step 2: Extracting cavity tile

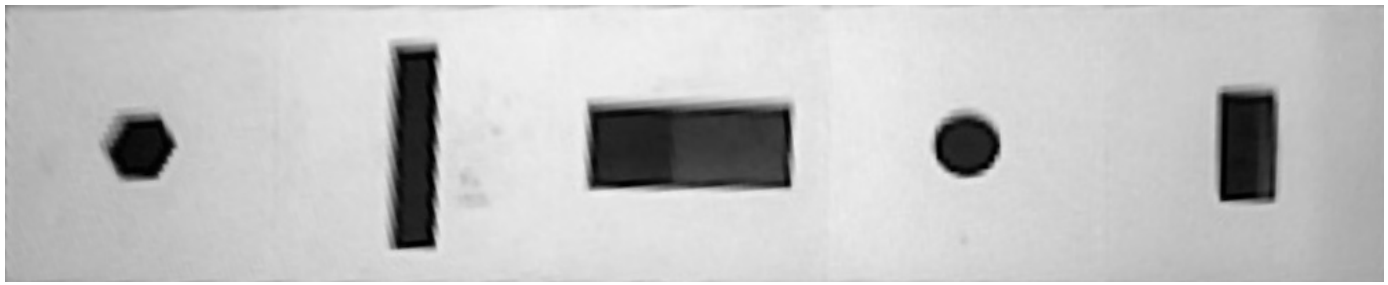


Figure 8 : The rectified row of tiles.

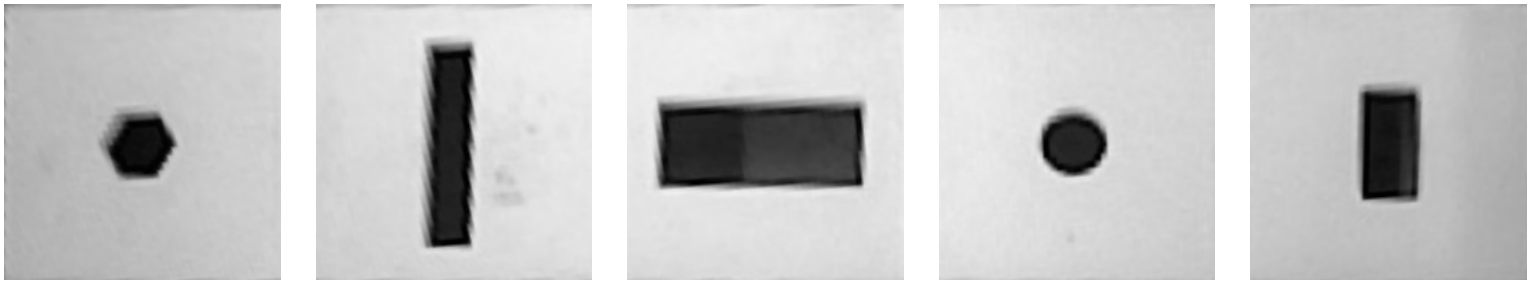


Figure 9 : The final five cavity tiles extracted from the PPT platform image.

step 3: Comparison

The extracted tiles are

- Resized
- Applied Threshold
- Applied Median Blur

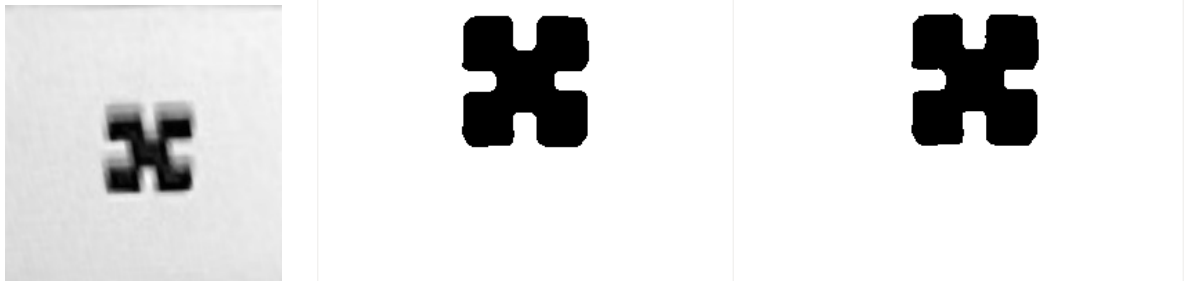
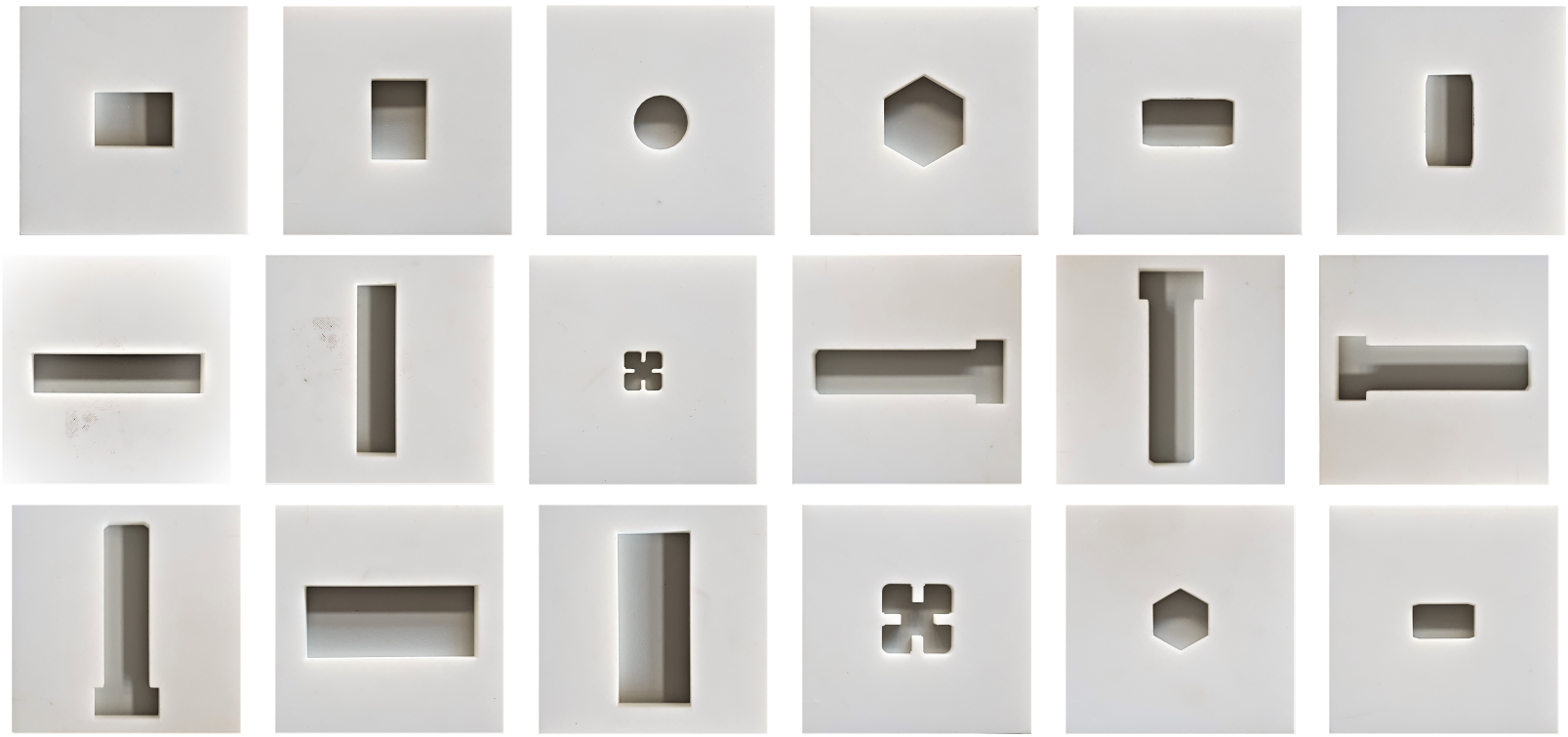


Figure 10 : **First**: The output extracted from the algorithm. **Second**: The input image is resized and applied threshold. **Third**: The image after median blur

step 3: Comparison

Database:



Mean Square Error(MSE)[3]

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[I(i,j) - K(i,j)]^2 \quad (1)$$

Structurality Similarity Index Measure (SSIM)[3]

$$SSIM(x,y) = \frac{(2\mu_x\mu_y + C_1)(2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)(\sigma_x^2 + \sigma_y^2 + C_1)} \quad (2)$$

where, $\mu - mean$ $\sigma - variance$ $c - covariance$

Step 3: Comaprison

Figure 11 : First: Input image. Second: Best match found using MSE. Third: Best match found using SSIM

Figure 12 : Both SSIM and MSE are able to differentiate the cavities of same shape but different size

Setup & Data

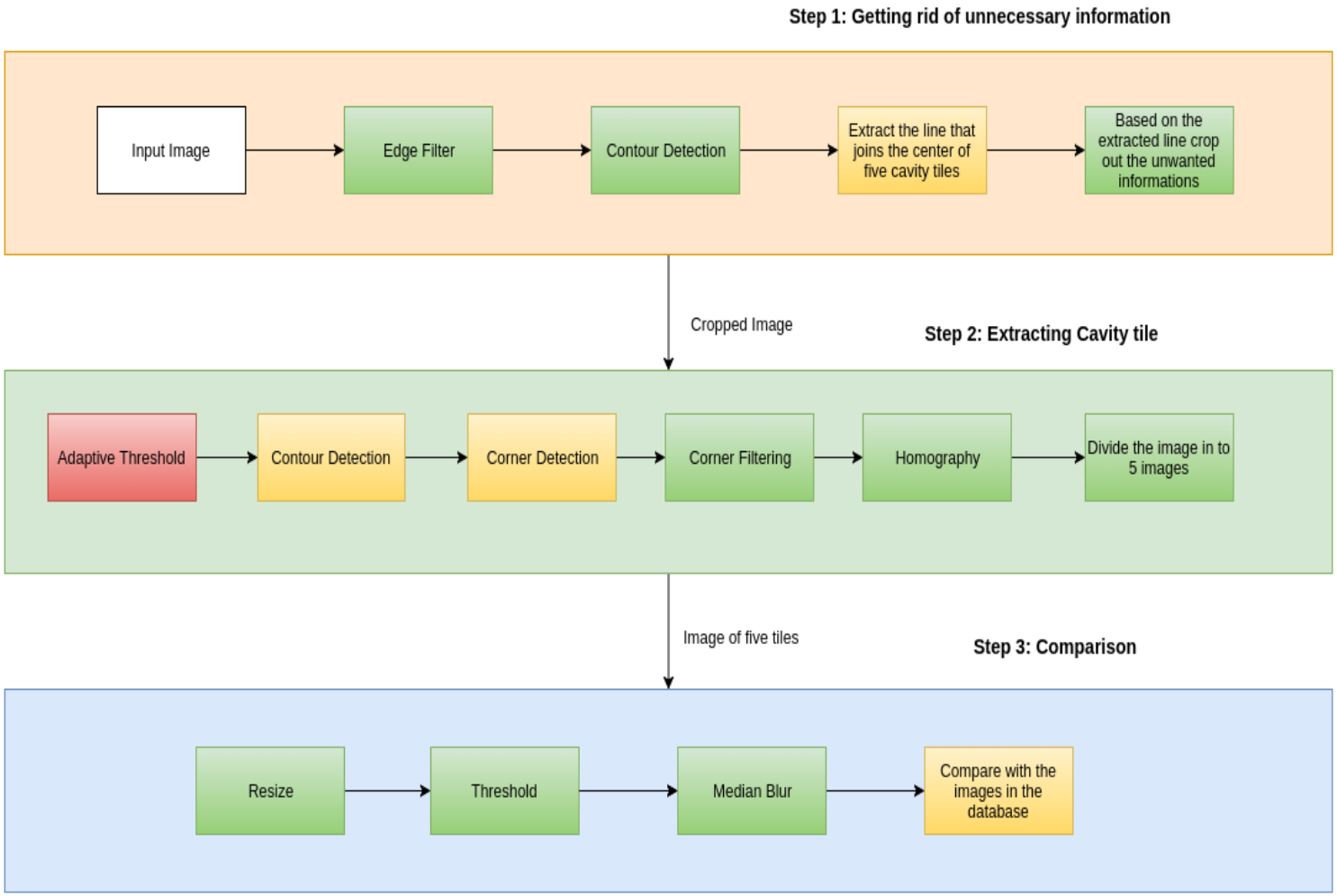
- 20 images that show different compositions of the tile row, from different perspectives.
- 12 images that show special conditions (lights out, 4 tiles and additional dark objects).

Results

Table 1 : Results of the Tests.

Performance	Normal cases	No lights	Add. Objects	4 tiles	Combined
Step 1	73%	50%	100%	100%	74%
Step 2	81%	66%	66%	33%	65%
Step 3	98%	30%	80%	100%	85%
Total	60%	10%	26%	33%	39%

Conclusion


Step 1: Getting rid of unnecessary information
Input Image
Edge Filter
Contour Detection
Extract the line that joins the center of five cavity tiles
Based on the extracted line crop out the unwanted informations
Cropped Image
Step 2: Extracting Cavity tile
Adaptive Threshold
Contour Detection
Corner Detection
Corner Filtering
Homography
Divide the image in to 5 images
Image of five tiles
Step 3: Comparison
Resize
Threshold
Median Blur
Compare with the images in the database

Conclusion

Step 1

- Good Performance, can account for additional dark objects
- Edge detection for dark images doesn't work satisfying
- Errors: No or wrong line extracted.

Step 2

- Adaptive threshold need to be improved
- Corner detection and homography: good performance

Step 3

- Best Performance
- No problem with the same shape in different size
- Errors: Dark images, similar shapes

Improvements & Future work

- Line extraction by distance between adjacent points
- Replace adaptive thresholding
- Predict position of corner points if not found or shown
- Add comparison using area of the contour and shape of the contour
- Select the best match using voting method

What we have learned

- Importance preprocessing of images using filters
- Sometimes manually constructed methods are more effective than cv libraries
- Incorporating real world knowledge in programming

Any questions?

Sources

1 https://www.aliexpress.com/price/intellectual-geometry-toy_price.html
2 https://www.generationrobots.com/de/402093-kuka-youbot-omnidirektionaler-mobiler-roboter-mit-arm.html
3 https://www.pyimagesearch.com/2014/09/15/python-compare-two-images/

The End