Image Processing report 3

Boogard Darian, Marogna Lorenzo

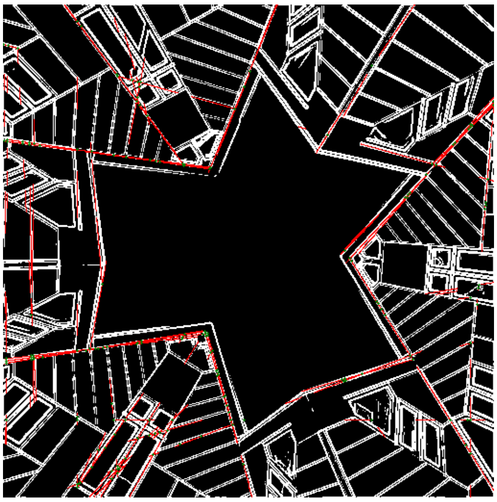
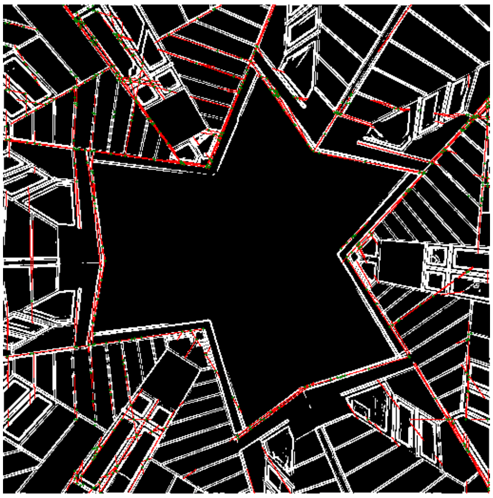
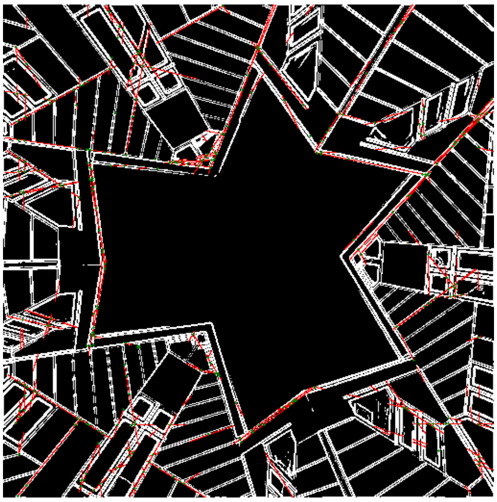
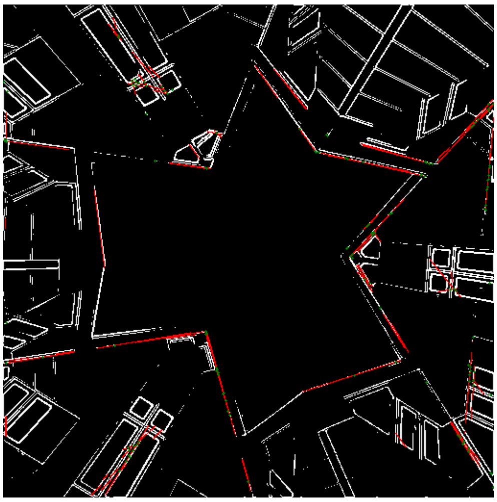
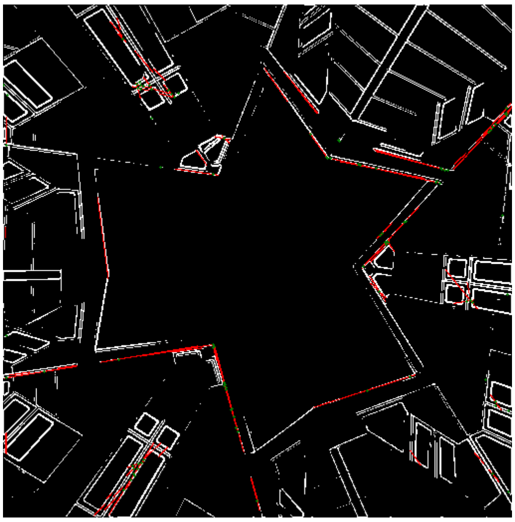
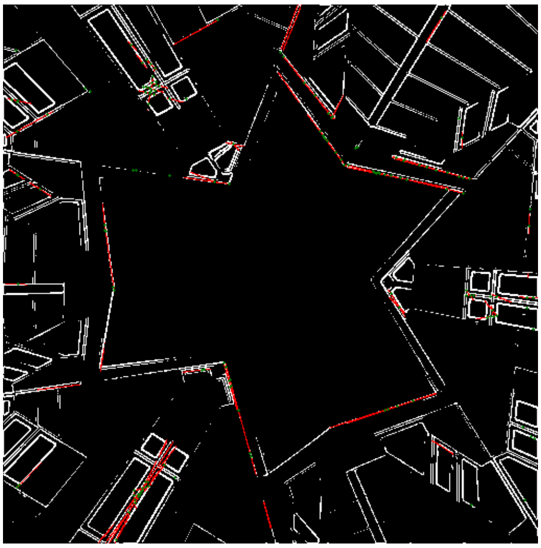
In this third image processing assignment we implemented some functions to detect lines on an image through the hough transformation. In order to show the effects of our operations, we needed an example.

Starting from image “A” from the first assignment, we obtained a new image by applying the edge detection function from PA1 and thresholding the image.
With this new binary image, in the first part of the assignment we applied the hough transformation and, in order to find the peaks corresponding to different detected lines, we thresholded the result (threshold may change), applied closing operation on the image and detected the different regions with the flood fill from PA2. For each detected region, we computed the centroid coordinates and used these values to describe the detected line.

Here is a table with different results:

Image	edge threshold	hough threshold	structural element	Minimum intensity threshold	Minimum segment length	Maximum gap
ht1	80/255	70%	SQ-11x11	127 / 255	10 px	5 px
ht2	80 / 255	65%	SQ-11x11	127 / 255	10 px	5 px
ht3	80 / 255	60%	SQ-25x25	127 / 255	10 px	5 px
ht4	200 / 255	55%	SQ-15x15	127 / 255	10 px	5 px
ht5	200 / 255	55%	SQ-31x31	127 / 255	10 px	5 px
ht6	200 / 255	50%	SQ-31x31	127 / 255	10 px	5 px

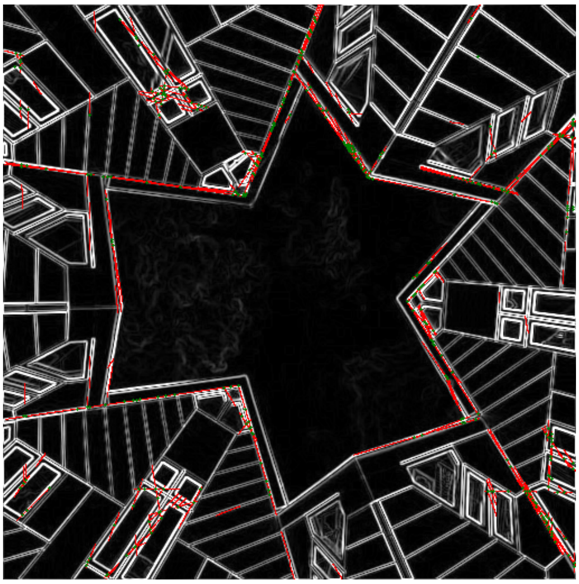
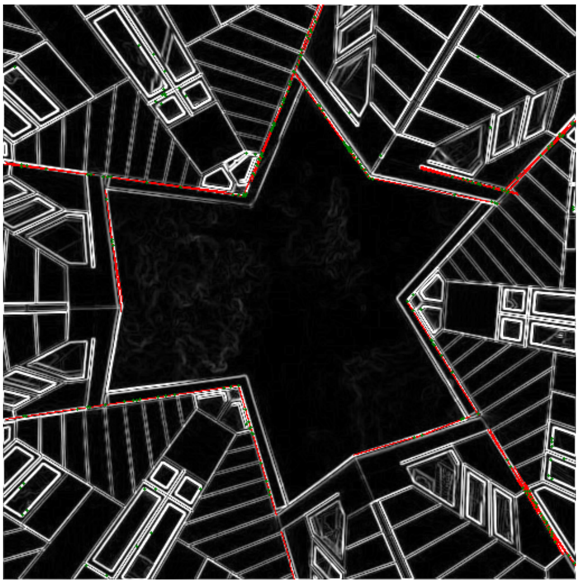
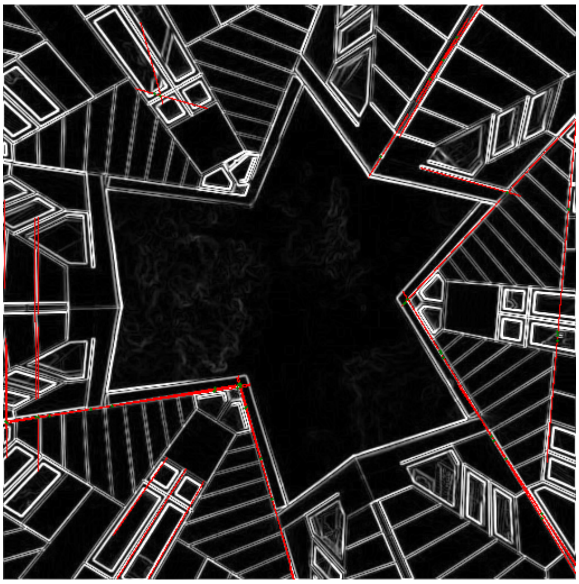
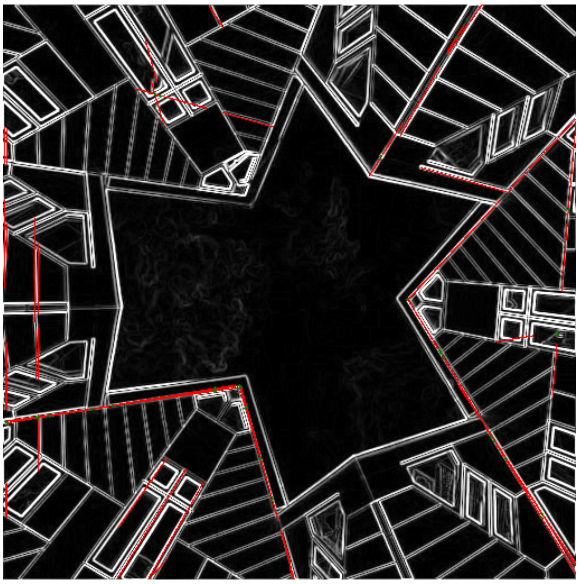
What we can say is that setting a higher edge threshold on the edges allows us to focus more on the most important edges in the image, while with the lower threshold the hough transform has peaks that are difficult to find because there is a lot of noise due to the fact that there a lot of little lines all with the same importance.
Another important thing that we noticed is that if the hough threshold gets lower the structural element that we use to close the image and find the peaks needs to be higher, in order to merge together points that are further between each other but still describing the same line. important lines have actually been detected. We think the best result among these 6 images is number 4 because a good amount of the important lines have been detected. Number 5 is also very similar, even though the Structural Element gets much bigger. So the reason why we choose number 4 is that the high edge threshold focuses more on the important edges in the image, the hough threshold allows us to find a good amount of pixels and at the same time keep a good precision while calculating the centroids and the 15x15 kernel is faster to apply to the image then the 31x31, giving the same result. All the other parameters stayed the same because 1. Minimum intensity is useless with a binary image; 2. 10 px for segment length and 5 px for maximum gap have always given a good result since the first image.

 1	 2
 3	 4
 5	 6

In the second part of the assignment we started again from image “A” and repeated the same experiments using the image with edge magnitude. The gray-scale image is giving an idea of how intense the edge is for each pixel, so the line detection is finding the most important edges that can be found in the image.
Here is a table with different results:

Image	hough threshold	structural element	Minimum intensity threshold	Minimum segment length	Maximum gap
ht7	60%	SQ-11x11	127 / 255	10 px	5 px
ht8	60%	SQ-11x11	127 / 255	70 px	5 px
ht9	70%	SQ-11x11	70 / 255	70px	10 px
ht10	70%	SQ-11x11	70 / 255	20px	5 px

The first thing we noticed in the second part is that the computation time to compute the hough transform is a lot higher than thresholding the image beforehand. This is because with a binary image a lot of BG pixels can be skipped, but now every pixel with intensity >= 1 / 255 is “important” to calculate. It could probably be possible to set some threshold at this level, ignoring every pixel with intensity < Th to save time (if intensity >= Th the pixel is NOT mapped to 255, otherwise it will be the same thing as before). Here we tried to play with Minimum intensity threshold value, putting it to 70 in some cases, but we didn’t notice a lot of difference. We also tried to raise the Minimum segment length and this allowed us to focus on the longest lines, giving a cleaner result. We also tried to raise the Maximum gap to 10 px but the behavior was not nice as it can be seen in the picture. The hough threshold hasn’t varied a lot in this part of the assignment, we noticed that both can be used and the difference is smaller than the one we got in the first part, due to the fact that peaks are usually easier to find in this case because smaller edges get “cut out” because of their importance and so they produce less noise. The structural element has been the same through the whole experiment in order to better compare other parameters. We think ht10 is the best result we got, even though it could also be possible to obtain even better results with different parameters.

7 	8 
9 	10 

Extra functions:

- Circle detection (input radius from GUI)
- Crossing lines (Green plus-shaped marker automatically inserted in the result if the crossing is two active (red) segments.

Buttons:

- Hough transform: apposite button
- find peaks: hough_line_detection shows every line and prints parameters in stdout.
- Find and Visualize segments: Line detection

- Hough Transform Angle limit (set with 45 and 90) : HTAngleLimit
- 3 Other buttons for utilities in generating images and evaluating results