Answers to questions in
Lab 2: Edge detection & Hough transform

Name: Andrea Camilloni Program: ID2222

Instructions: Complete the lab according to the instructions in the notes and respond to the questions stated below. Keep the answers short and focus on what is essential. Illustrate with figures only when explicitly requested.

Good luck!

*All the tests done are in lab2.ipynb, while the built functions in lab2.py

Question 1: What do you expect the results to look like and why? Compare the size of *dxtools* with the size of *tools*. Why are these sizes different?

Answers:

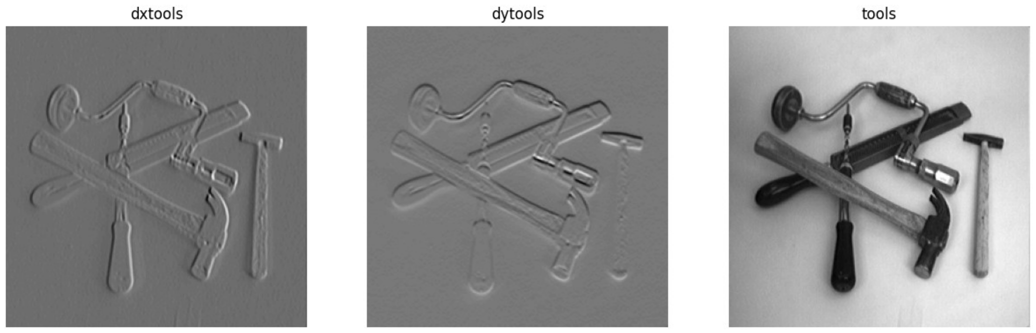

Fig.1: dxtools, dytools, tools images shown with function showgray(). dxtools, dytools partial derivatives of tools

Partial derivatives are computed using Sobel Operator, i.e. by convolving the Sobel kernels with the original image.

Sobel kernels: $K_x = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix}, K_y = K_x^T$

Partial Derivatives: $L_x = f(x,y) * K_x$ and $L_y = f(x,y) * K_y$

The partial derivatives detect discontinuities in the image brightness (EDGES).
The partial derivative along x, L_x, highlights fluctuations along x-axis, and partial derivative L_y highlights fluctuations along y-axis.

A first attempt to detect edges, using first derivatives, is shown in fig.1.

Matrix *dxtools* is of shape 254x254, while *tools* is 256x256. This is due to the **valid convolution** with the kernel, that does not use any padding on the input.

Given the NxN image, the kxk kernel, a valid convolution results in a matrix of shape [N-k+1]x[N-k+1]. We are loosing [k-1]x[k-1] pixels (i.e 2x2 pixels).
We cannot compute the derivatives at the edges.

Question 2: Is it easy to find a threshold that results in thin edges? Explain why or why not!

Answers:
We have to find a threshold s.t. the Magnitude of the gradient is larger: $|\nabla L| \geq \tau$
And for which only 'stronger' edged are retained.

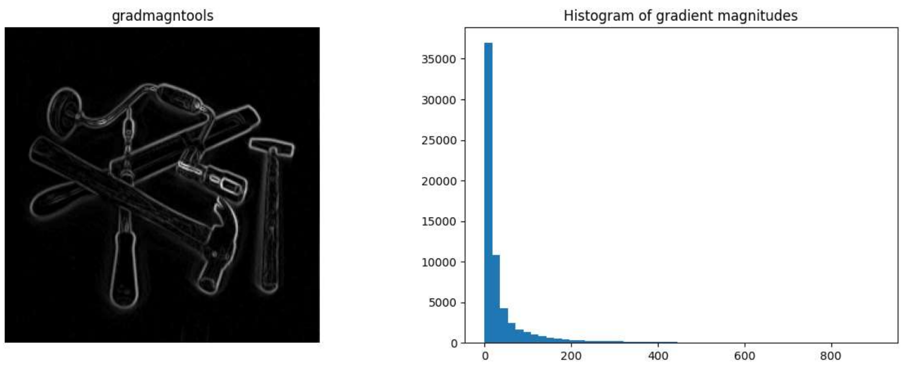

Fig. 2: Left:Magnitude of the gradient of tools. Right: histogram of gradient magnitudes.

In fig. 2 we can see that most of the values for the Magnitude are in the range [0,100]. So a threshold greater than 100 would retained too much 'info', resulting in only edges with higher contrast. Viceversa, low values of the threshold would retained thick edges and the noise.

This results can be observed in fig.3.

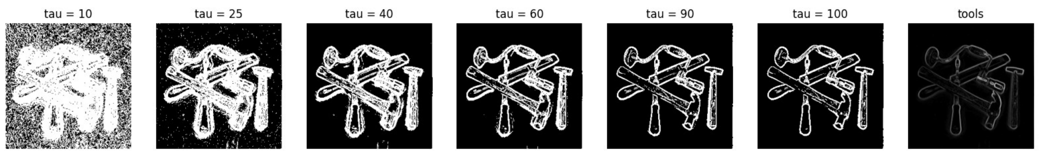

Fig. 3: Thresholding for different values of tau.

Question 3: Does smoothing the image help to find edges?

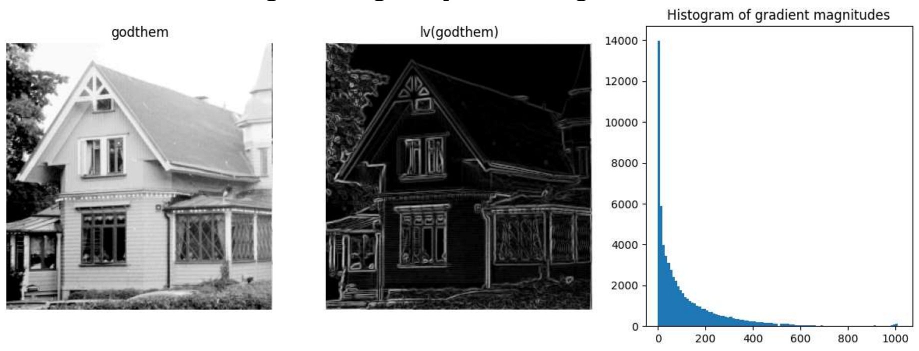

Fig.4: Left: original image. Middle: Magnitude of gradient. Right: Distribution of gradient magnitudes

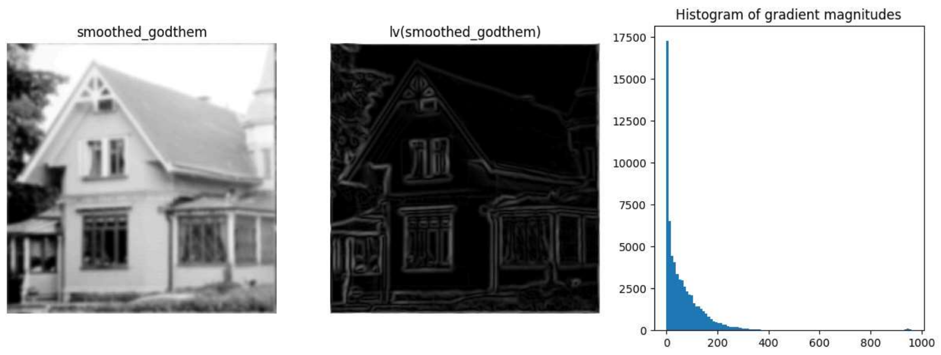

Fig.5: Left: smoothed image. Middle: Magnitude of gradient. Right: Distribution of gradient magnitudes

Answers:
What happens when we smooth the image is that we are removing the noise, but smoothing too much can lead to a huge loss of the edges because they are smoothing with the noise.

After smoothing the image, the high frequency are removed, and we can observe from figures 4 and 5 that also values of the gradient magnitudes become smaller. This affect the choice of the threshold; we now need a smaller threshold than the original case.

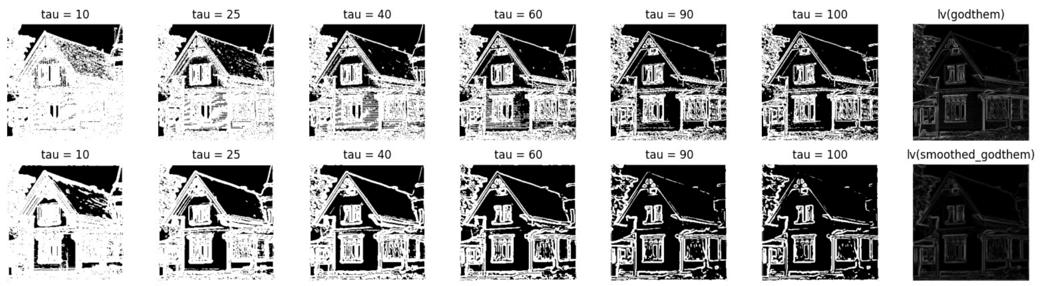

Fig.6: thresholding for both magnitudes of the original and smoothed image

Fig.6 shows the differences between thresholding in case of the magnitude of the gradient of the original image and of the smoothed one.

Question 4: What can you observe? Provide explanation based on the generated images.

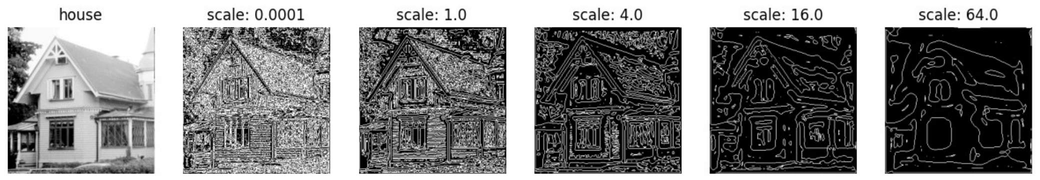

Fig.7: 2^{nd} derivative of the image 'house' at different scales

Answers:
What we can observe wrt to first derivatives is that in this case even thin edges are retained, but as soon as the scale increases, they disappear, due to the smoothing. What we are looking at is the 2nd derivative $L_{vv} = 0$, where we can easily spot lot of zero-crossings values. The problem with sero crossing values is that detects also "false" edges.
The smoothing is reducing those zero-crossing values but is also dropping important details about the edges, which at larger smoothing scales look distorted (e.g. scale 64).

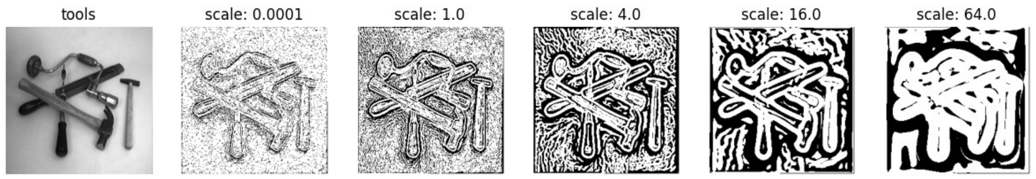

Fig.8: 3rd derivative of the image 'tools' at different scales

We are now considering the 3rd derivative instead of the 2nd, thus we are now looking for $L_{vvv} < 0$, i.e. white areas(max points). Lot of noise is present in the image, and even by smoothing is not possible to reduce it. Increasing the scale too much mess up with the edges, but we are not loosing them.

Question 5: Assemble the results of the experiment above into an illustrative collage with the *subplot* command. Which are your observations and conclusions?

Answers:
By smoothing more, we get thicker edges!
Smoothing affects the way in which the derivatives change. The first derivative is hence lower and wider. The same factors affect both the second derivative and the third derivative.

Question 6: How can you use the response from *Lvv* to detect edges, and how can you improve the result by using *Lvvv*?

Answers:
$L_{vv} = 0$ is used to find the points where the magnitude of the first derivative has a zero-crossing value, but this is finding both maximum and minimum points of L_v . $L_{vvv} < 0$ is instead used for finding only maximum values. Using them together can be a solution for detecting edges.

Question 7: Present your best results obtained with *extractedge* for *house* and *tools*.

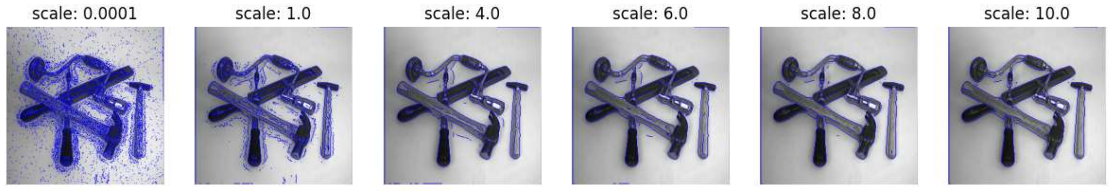

Fig.9a: results obtained by extractedge() on tools, with a threshold = 20

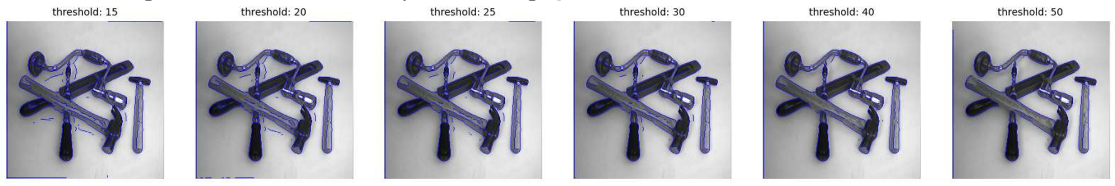

Fig.9b: results obtained by extractedge() on tools, with a scale = 4

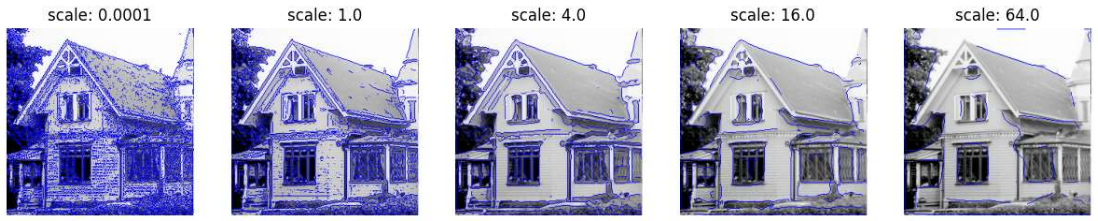

Fig.10a: results obtained by extractedge() on house, with a threshold = 20

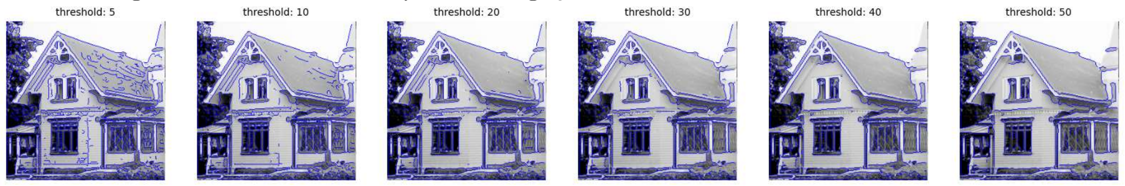

Fig.10b: results obtained by extractedge() on house, with a scale = 4

Answers:
Smaller scales results in retaining most of the noise, while greater scales would smooth too much, at the point that we do not detect any edges. A good value for the scale spotted in the experiments is 4, see Figures 9a and 10a. By smoothing the right amount we should get less noise and better/thinner edges.
Once we set a scale (=4), in figures 9b and 10b we can observe results for different threshold values.
Lower threshold values would retain even "false" edges, while higher values might keep too much information. Thresholding works well when the noise has lower gradient magnitude than the edges.

Question 8: Identify the correspondences between the strongest peaks in the accumulator and line segments in the output image. Doing so convince yourself that the implementation is correct. Summarize the results of in one or more figures.

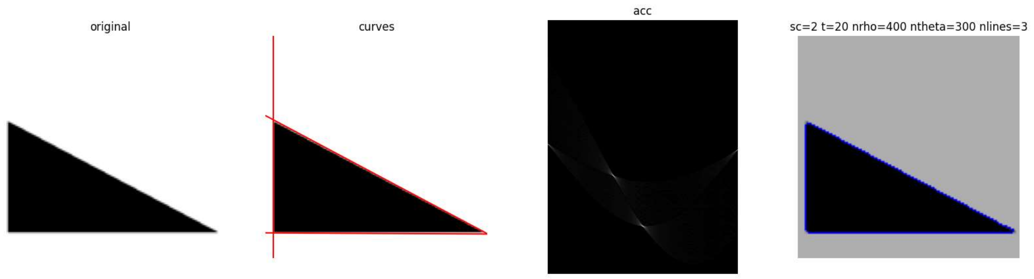

Fig.11: Original image, curves, accumulator space, segments

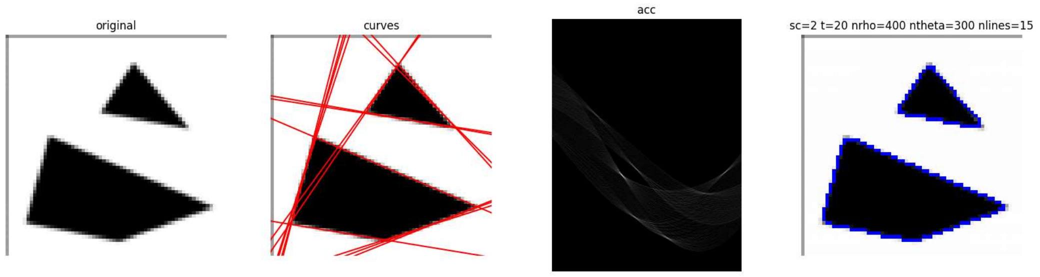

Fig.12: Original image subsampled twice, curves, accumulator space, segments
Answers:
Looking at figures 11 and 12 we can identify a line in the image domain for each peak in the accumulator space.

Let’s consider Figure 11, the 3 lines correspond in the accumulator space to the following coordinates: (r1,t1), (r2,t2), (r3,t3).
The mapping between the two spaces is shown in figure 13. For instance if we consider the yellow line, this will correspond in the accumulator space to a peak of coordinate (r1,t1)

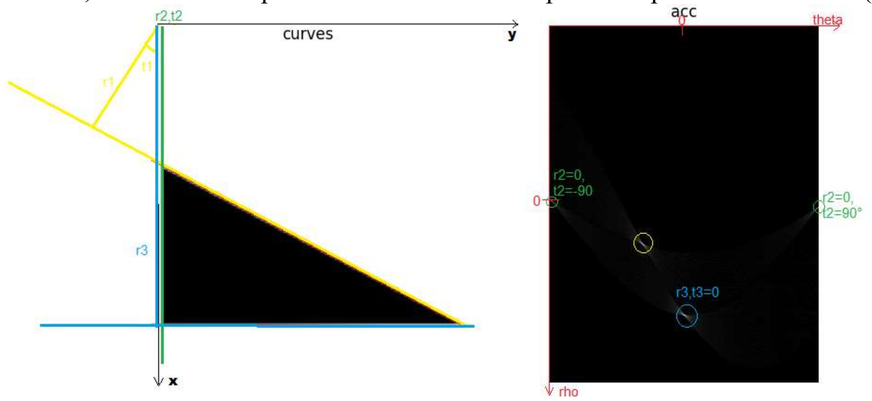

Fig.13: Image space and accumulator space.

Question 9: How do the results and computational time depend on the number of cells in the accumulator?

Answers:
The computational time doesn’t depend on the number of rhos but depends on the number of thetas. As we increase them the computational time increases. This is because we have to compute the lines for each value of the angles. Both the numbers of rhos and thetas determine the number of cells. Let’s say that an increasement of the number of cells(i.e. thetas) results in an increasement of the time, and determines also higher accuracy.

Question 10: How do you propose to do this? Try out a function that you would suggest and see if it improves the results. Does it?

Answers:
The idea id to introduce a weighted voting system depending on the magnitude of the gradient in that point.
Using a logarithm voting, instead of the voting()=1, leads to have smaller difference between high and low intensities. The log will increase the possibility of edges with lower values of gradient magnitudes.
