DD2423 Lab 2
Edge detection Hough transform

Andrej Wilczek - andrejw@kth.se
Ilian Corneliussen - ilianc@kth.se

November 2019

1

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?*

Answer: When we apply the difference operator (Sobel) in the x dimension we detect edges mainly along the y direction and vice versa. This is because we detect change in these directions which corresponds to edges in the other dimension. Where we have very subtle change the image becomes grey.
The size of the image after the difference operator is applied is two pixels smaller in both dimensions. This is due to the 'valid' argument in the `conv2` function in Matlab which only returns the parts of the convolution that where computed without zero-padded edges and since we use a 3x3 kernel we loose one line of pixels on each edge.

Figure 1: Image of tools with first order difference operator in x and y dimension.

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

Answer: Finding a threshold that results in thin edges without removing some of the less defined edges in the image is hard. In Figure 3 we can see that the lower edge of the hammers handle has been removed but the other edges are still quite thick. The histogram in Figure 2 can give us a indication of a reasonable initial guess for the threshold but further tuning was required in order to improve results.

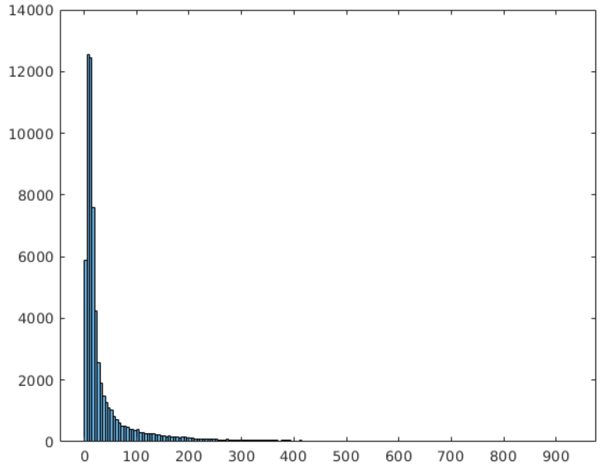

Figure 2: Histogram of the magnitude of the image.

Figure 3: First order derivative with thresholding.

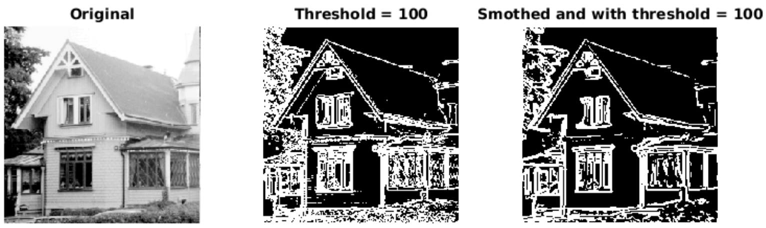

Figure 4: Image with thresholding and smoothing.

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

Answer: With smoothing the edges are a little easier to find, as can be seen in Figure 4 some of the noise in the image has been removed by the smoothing. The problem however is that when we increase the smoothing we begin to distort the true structures in the image.

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

Answer: The zero crossings of $\tilde{L}_{vv}$ plotted in Figure 5 shows us all points where the gradient is zero i.e. both local maximum and minimum points for the gradient. Due to noise, for the images with little smoothing, this occurs not only when we actually have an edge resulting in lots of unwanted lines in the plot. When we increase the smoothing we get less of the false edges but with too much smoothing we start to loose the actual edges as well.

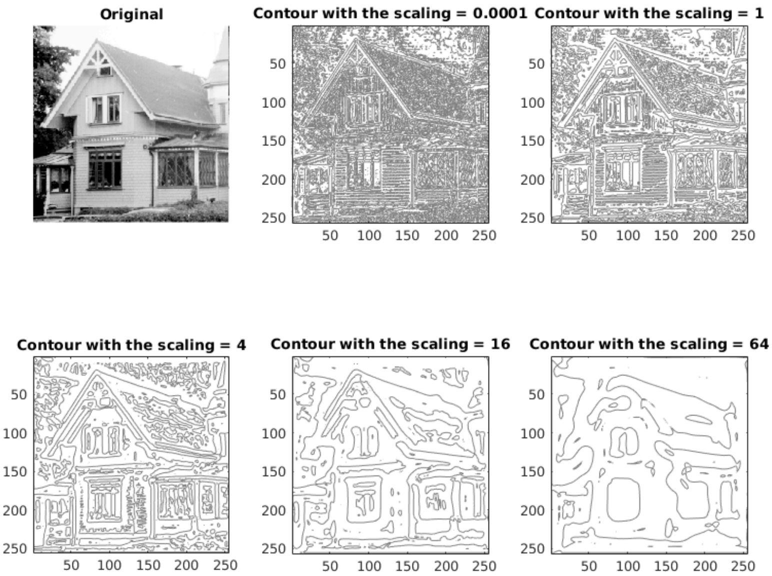

Figure 5: Second order derivative with different scaling.

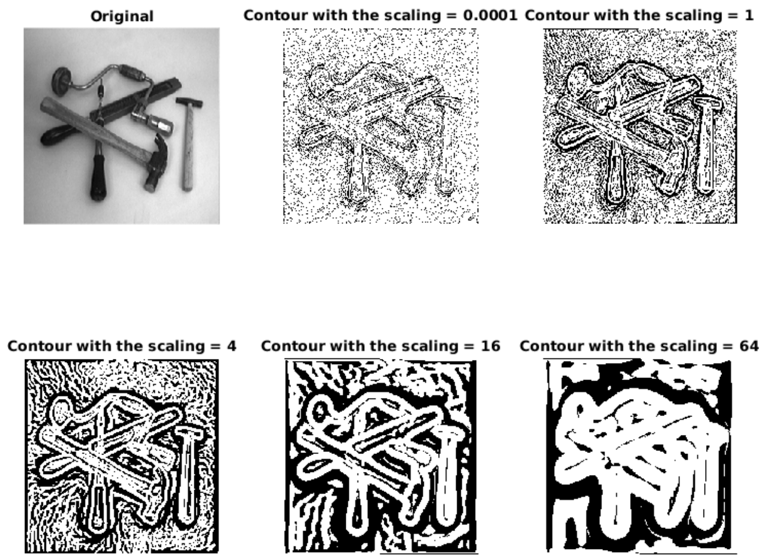

Figure 6: Third order derivative with different scaling.

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

Answer: The larger scaling/smoothing factor the more of the details in the pictures are blurred out. For example one could notice that in Figure 6 with scaling 64, the most distinct edges are left from the original picture and further, the smaller scaling we apply the more of the edges from the details are displayed. The more smoothing the larger the concave areas in the first derivative are therefore the thicker the edges detected by the third derivative, see Figure 7.

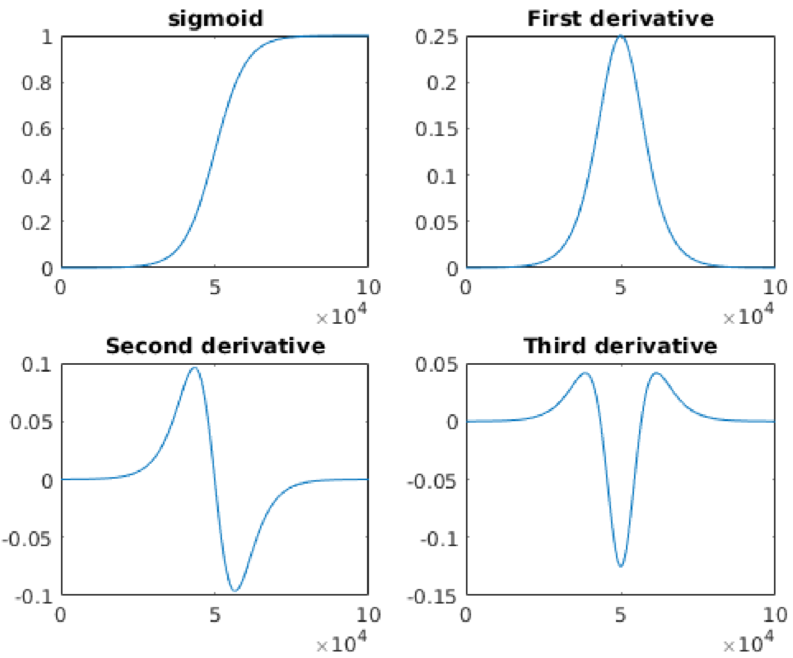

Figure 7: Edge function and it's first, second and third order derivative.

Question 6: *How can you use the response from $\tilde{L}_{vv}$ to detect edges, and how can you improve the* *result by using $\tilde{L}_{vvv}$?*

Answer: By computing all the zero crossings of $\tilde{L}_{vv}$ we can detect edges but in order to avoid false positives we should also introduce the constraint that $\tilde{L}_{vvv}$ should be less than zero which will give us all the maximum points of the gradient which corresponds to the edges.

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

Answer: The best results where obtained with a scaling factor of 4 and threshold of 40 for the house image and for the tools a scaling factor of 16 with a threshold of 40.

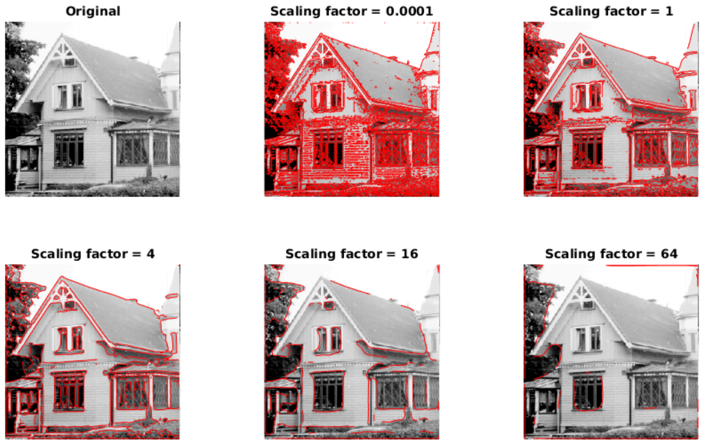

Figure 8: Results of extractedge with different scaling.

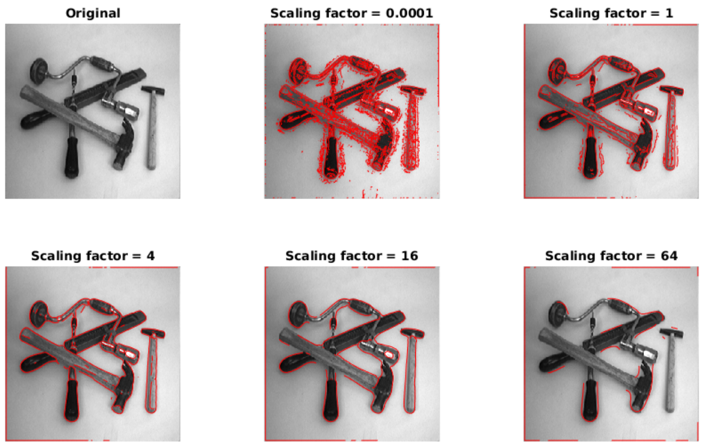

Figure 9: Results of extractedge with different scaling.

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 in one or more figures.*

Answer: As can be observed in Figure 10 we have three peaks that corresponds too $(\rho, \theta) = (43, -27.5^o), (115, 0^o)$ and $(4, 90^o)$ which gives the corresponding lines in cartesian space see Figure 11. One could be sure of that the transformation between parametric space and the cartesian space by doing the transformation by hand as shown in Figure 12.

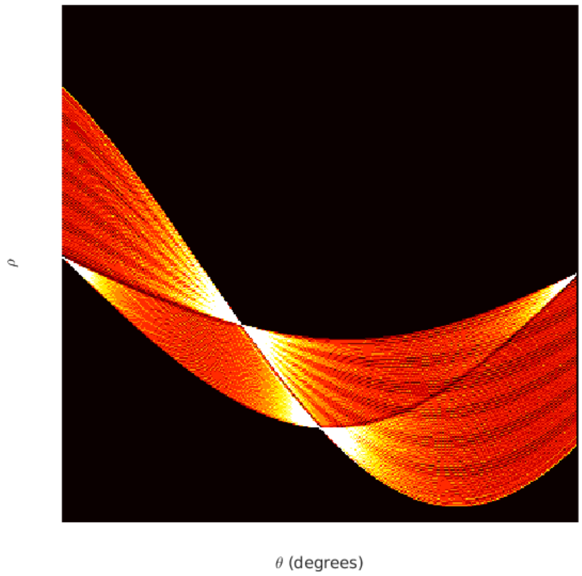

Figure 10: $[\rho, \theta]$ space for the image triangle128.

Figure 11: Hough edge-detection for the image triangle128.

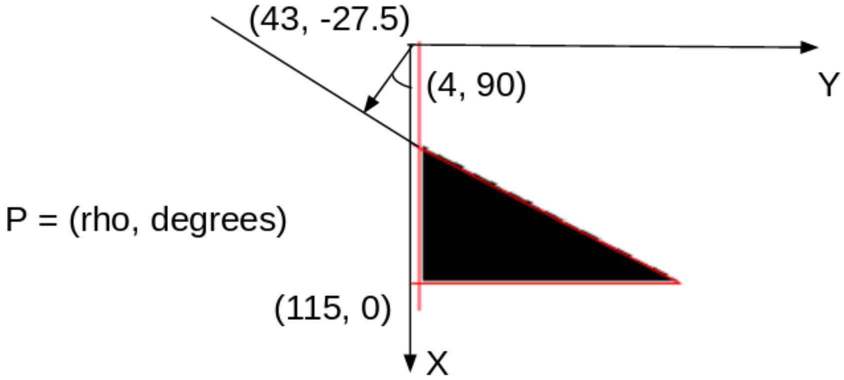

Figure 12: Convinced.

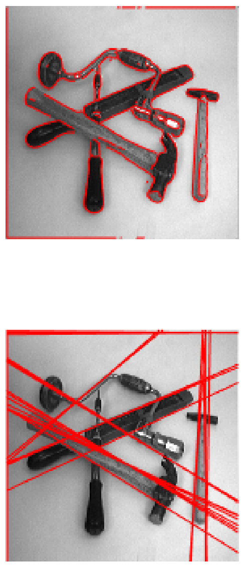

Figure 13: Hough edge-detection for the image few128.

Figure 14: Hough edge-detection for the image phonecalc256.

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

Answer: With a large number of cells more lines are found i.e. we get a spread of lines concentrated around the true lines, however the computational speed is decreased with the amount of cells.

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?*

Answer: One could increment with the magnitude from the first derivative instead of using 1 to enhance the effect from the stronger lines. As can be observed in Figure 15 compared to Figure 16 some of the lines have another priority, such as the lines that correspond in the bushes have been moved to the rooftop. In such way the distinct lines that are shorter have a higher probability to emerge.

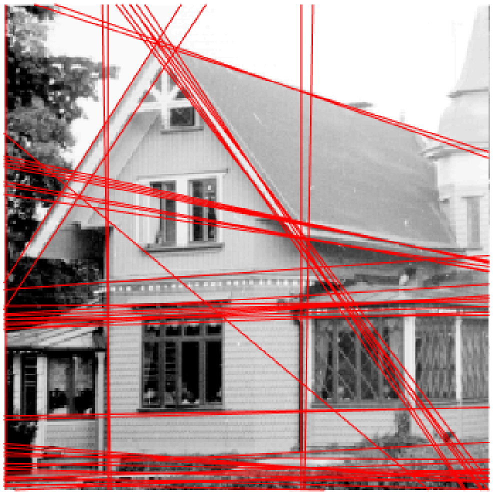

Figure 15: Increment = 1.

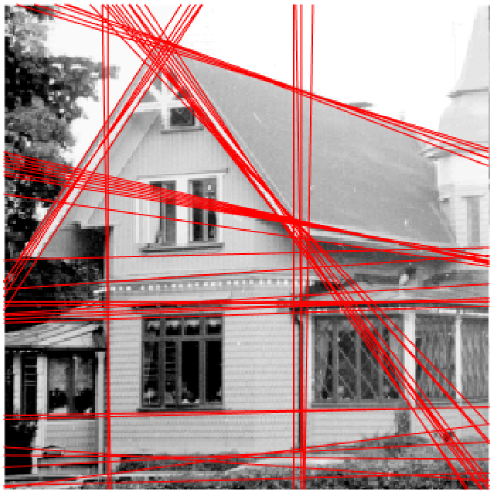

Figure 16: Increment = mag(x,y).