Lab 2: Edge detection & Hough transform

Oskar Erlandsson
Kildo Alias

November 27, 2019

0.1 Difference operators

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: We expect to see the edges in the horizontal or vertical directions depending on if we use deltax or deltay.

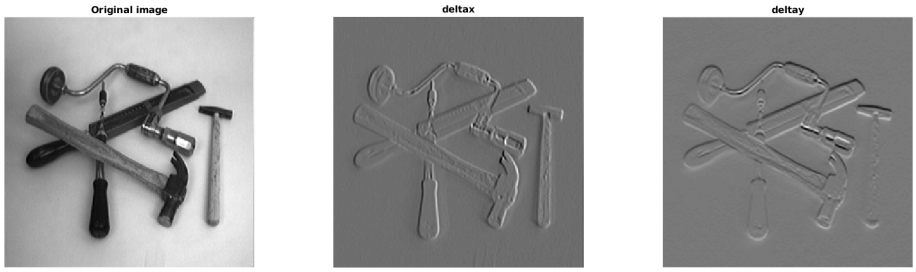

Figure 1: Sobel operator

If we observe the most-right hammer in the original picture it can easily be observed that for when applying deltax the edge between the background and shaft, in the x direction, can be observed whereas for delta y the edge between the head and the shaft can be observed. This is because the Sobel operator computes the gradient of the image intensity function. The size of dxtools and dytools are 2 pixels smaller in width and height, this is because the conv2 function used only returns parts that are computed without the zero-padded edges if the valid flag is raised.

0.2 Point–wise thresholding of gradient magnitudes

Question 2: Is it easy to find a threshold that results in thin edges? Explain why or why not!
Answer: It is hard because thin edges in one part of the image may result in other parts not showing up at all and if more edges are included, i.e lower threshold, some of the edges will be thicker.

Question 3: Does smoothing the image help to find edges?
Answer: It helps reduce noise, due to the filter removing high frequency content. Applying a gaussian filter with too large variance will on the other hand remove both edges and noise. But if you tune the parameters, in this case the variance t and the threshold you will get a more observable picture than in the case where you only tune the threshold.

0.3 Computing differential geometry descriptors

Question 4: What can you observe? Provide explanation based on the generated images.
Answer: The images shows different scales. Higher values of t, ie more blurring, shows more rough shapes and less details. The lower the scale the more details are shown but also more of the noise is present. The lines are thinner than the first derivative, this is because only max and min values are plotted.

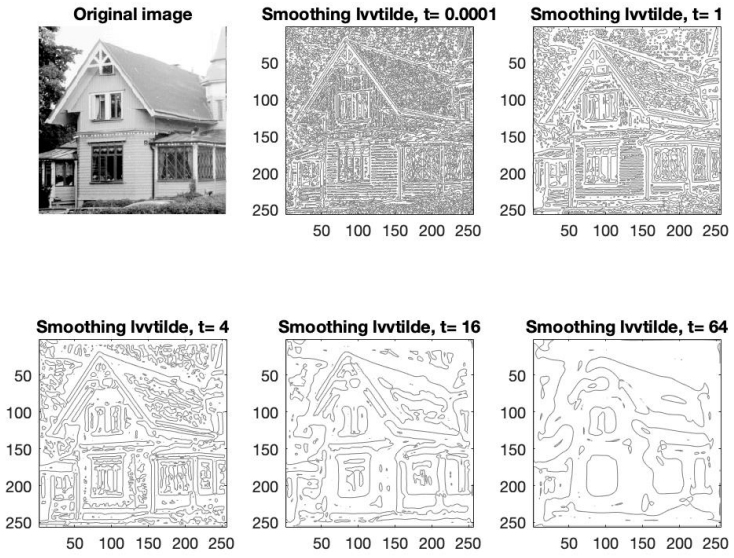

Figure 2: The second derivative of the picture in scale space

About Question 5: Study the sign of the third order derivative in the gradient direction by loading the image $tools = few256$ and show the result of showgrey(Lvvvtilde(discgaussfft(tools, scale), 'same') < 0) for the same values of scale. What is the effect of the sign condition in this differential expression?

Question 5: Assemble the results of the experiment above into an illustrative collage with the subplot

command. Which are your observations and conclusions?
Answer:

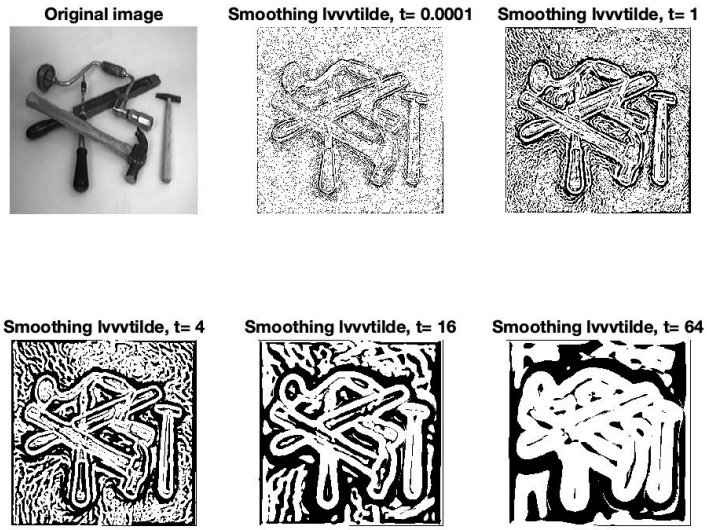

Figure 3: The third derivative of the picture in scale space

For small values of the variance we get more noise on the other hand for large variance the area of which

$$\tilde{L}_{vvv} = L_x^3 L_{xxx} + 3L_x^2 L_y L_{xxy} + 3L_x L_y^2 L_{xyy} + L_y^3 L_{yyy} < 0 \tag{1}$$

holds will be larger due to the gaussian smoothing and will therefor give thicker edges.

Question 6: How can you use the response from $\bar{L}_{vv}$ to detect edges, and how can you improve the result by using $\tilde{L}_{vvv}$?
Answer: By using $\bar{L}_{vv}$ we can detect maximums and minimums, with the constraint $\tilde{L}_{vvv} < 0$ we get the areas for where there are a maximums. By combining $\bar{L}_{vv}$ with $\tilde{L}_{vvv} < 0$ we can get the maximum and the exact point for where the edge take place.

0.4 Extraction of edge segments

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

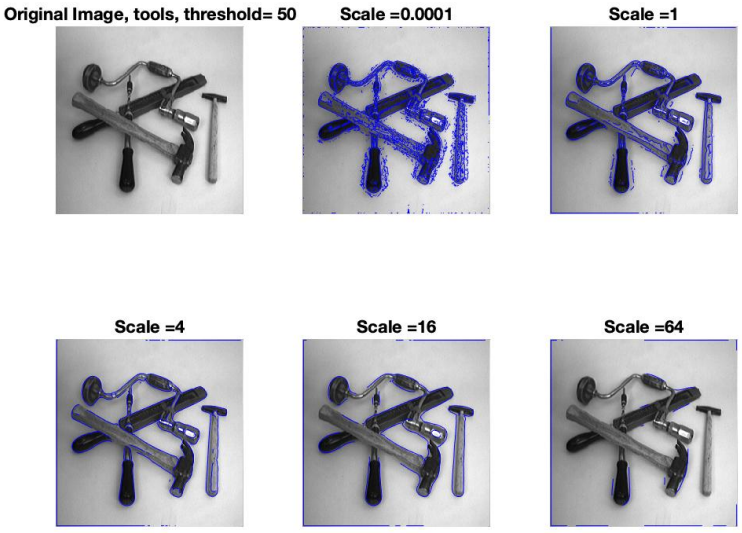

Figure 4: Edges for different values of scale

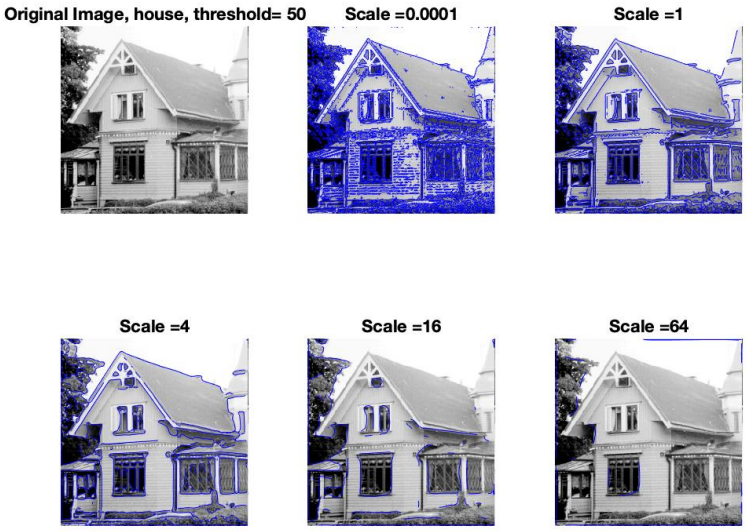

Figure 5: Edges for different values of scale

For the tools set we think that when $scale = 4$ we get the best result whereas for the house we think that when $scale = 1$ give the best result where we, although a bit noisy, find the most edges.

0.5 Hough Transform

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.
Then apply the same procedure to the test image houghtest256.
Show the results of your procedure houghedgeline applied to the images few256, phonecalc256 and godthem256.

Answer: The strongest peaks in the accumulator corresponds to point where a lot of lines go through, therefor giving the best line segments for the given picture.

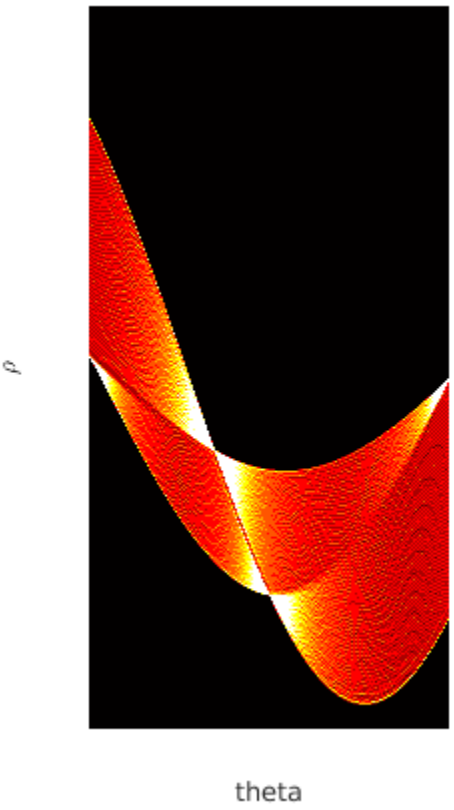

Figure 6: imshow(accumulator) for triangle128

The light areas in Fig. 6 corresponds to the best lines that describes the edges of the image , theta is discretizized between $(-\frac{\pi}{2}, \frac{\pi}{2})$. The rightmost light area gives the vertical line, the light area in the middle corresponds to $\theta = 0$ and therefor gives the vertical line, the left are that is light gives the angled line.

Figure 7: Hough transform applied to triangle128

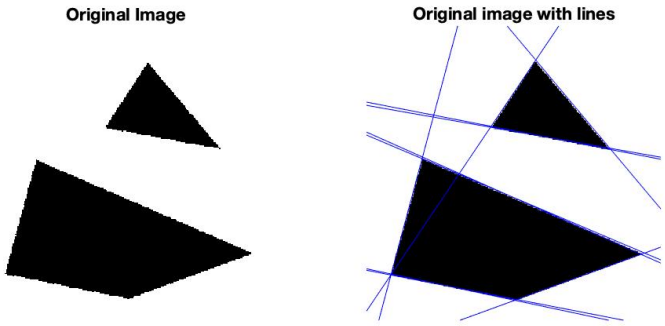

Figure 8: Hough transform applied to houghtest256

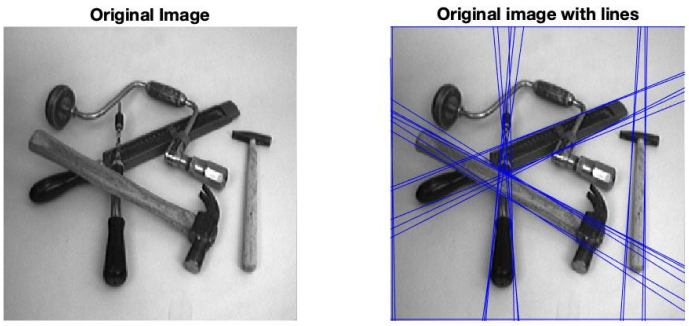

Figure 9: Hough transform applied to few256

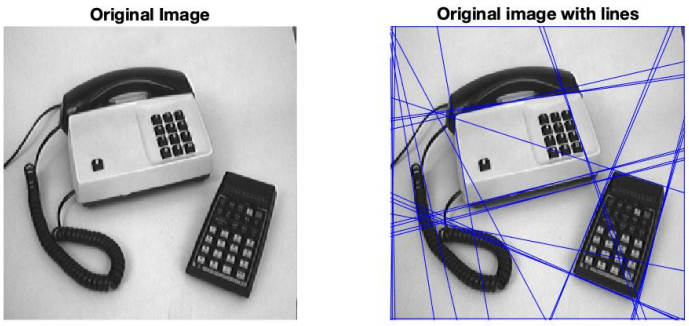

Figure 10: Hough transform applied to phonecalc256

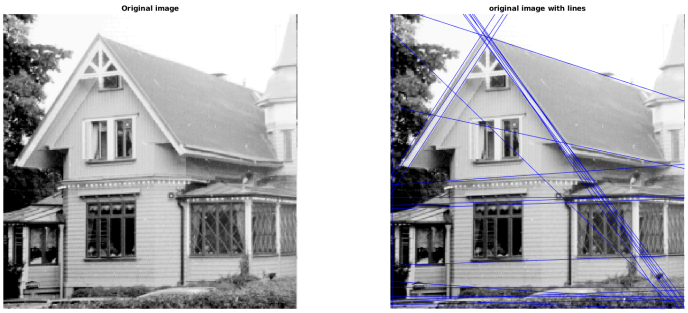

Figure 11: Hough transform applied to godthem256 with scale t=4

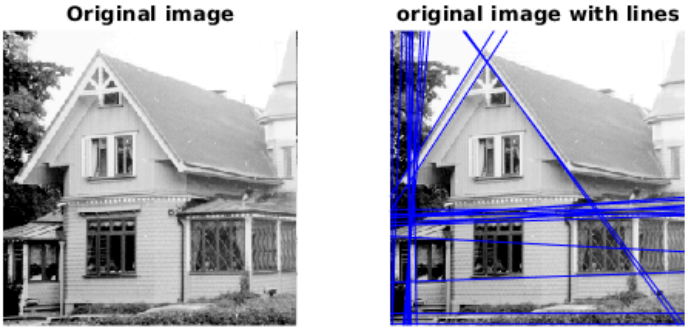

Figure 12: Hough transform applied to godthem256 with scale t=4 and magnitude increment

Question 9: How do the results and computational time depend on the number of cells in the accumulator?
Answer: Increasing the accumulator space by increasing ntheta will increase comptutational time but will give better solutions.

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: Instead of letting $increment = 1$ we instead let $increment = magnitude(x, y)$ where the magnitude is given by Lv(inpic,shape). The magnitude increment puts more significance on the contrast of the edge. It can therefore better detect smaller but more clear edges. The $increment = 1$

puts more weight on the length of an edge and therefore better can detect longer but more gradual edges. Squaring the magnitude further increases this effect. For reference see Fig. 11 and 12. With magnitude incriment the short white pillars with a black background in the lower left corner of the picture are detected but not the grey long roof with the white background.