Answers to questions in Lab 2: Edges detection and Hough transform

Benedith Mulongo, master year 2 Machine learning

September 2019

1 Tasks

Question 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?

Figure 1: Sobel operator effect on the image

- **Size image :** [256,256]
- **Size Sobel.X :** [254,254]
- **Size Sobel.Y :** [254,254]

In the convolution operation, we ca specify to either get a output matrix/image which have bigger, smaller och same dimension

The expectation is to see image where the rapid variations and transitions are highlighted. The size of the output image is smaller than the input image because we use convolution with No zero padding, with means that the output image inevitable becomes smaller. However when using zero padding we can create an output image of a size bigger or equal to the input image.

Question 2

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

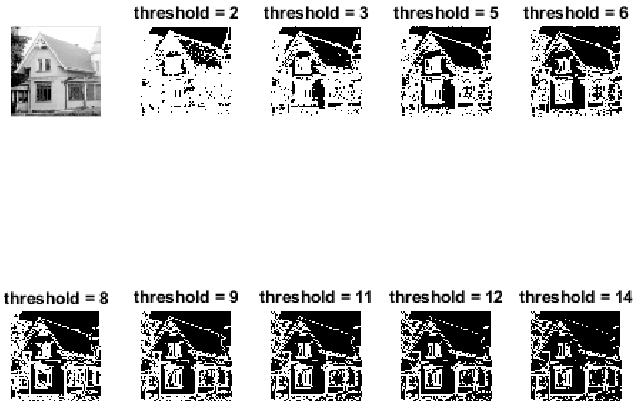

Figure 2: Image magnitude threshold

It is not easy to find a threshold that result to an image with thin edges. When looking to figure 3 below, depicting the magnitude strength of the image. We see that an edge can be understand as a continuous flow, the magnitude region is not very clear distinguishable. Any threshold will result to an image where the edges of lower strength will be absent and many regions will correspond to that threshold which result to a thick edges.

Figure 3: Image magnitude

Question 3

Does smoothing the image help to find edges?

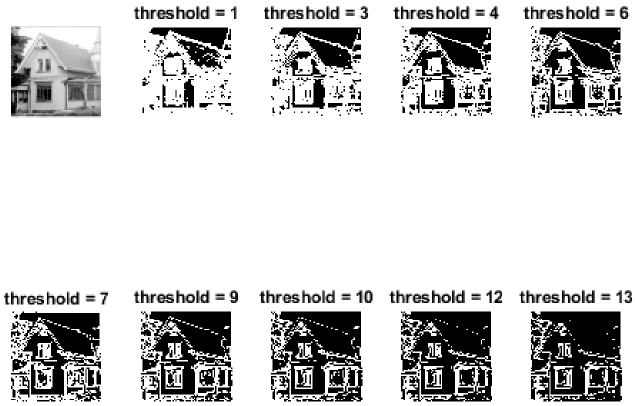

Figure 4: Image magnitude threshold with Gaussian kern.

The effect of smoothing is that the resulting image are less noisy, however the thickness of the output image are not different.

Question 4

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

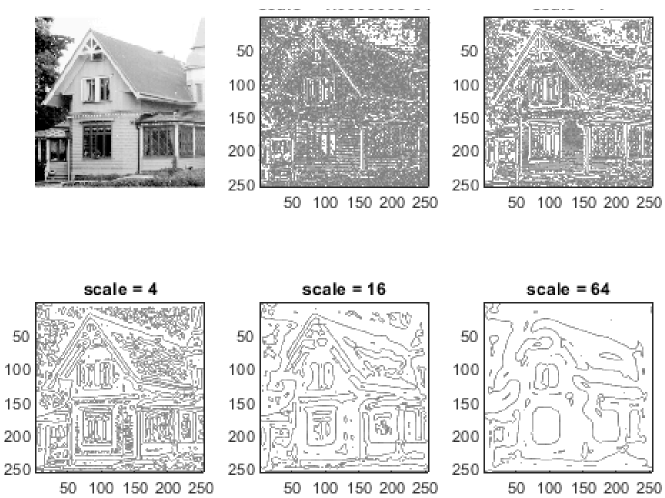

Figure 5: LvvTilde

What we observe is that resulting image are thin edges, not as when we were using the threshold on the image magnitude strength. The zero crossing is found by evaluation the point that for which the following conditions are true :

$$\tilde{L_{vv}} = L_v^2 L_{vv} = L_x^2 L_{xx} + 2 L_x L_y L_{xy} + L_y^2 L_{yy} = 0 \tag{1}$$

Every points for the conditions are true are collected with result to a thinner image compared to the grey zone of the threshold method on the image magnitude.

Question 5

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

The third derivatives is defined as :

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

In Matlab this condition has the effect to create a new binary matrix where there is a 1 when $f(x, y) < 0$ and vice versa.

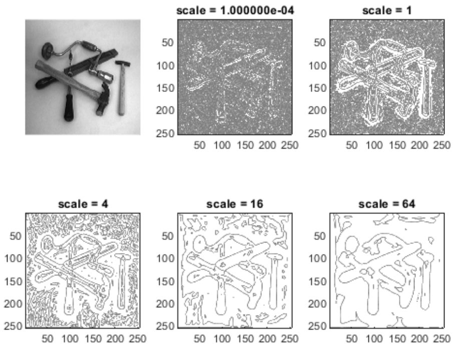

Figure 6: Zero crossing

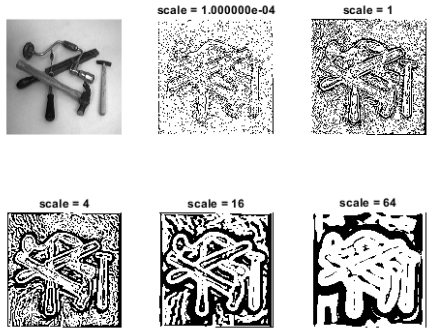

Figure 7: Third derivatives

The observation is that when we increase the variance of the Gaussian smoothing the region where the object containing become wider and wider. with a low variance the image are less clear and more noisy. The third derivative image are not as clear as the second order image.

Question 6

How can you use the response from L_{vv} to detect edges, and how can you improve the result by using L_{vvv} ?

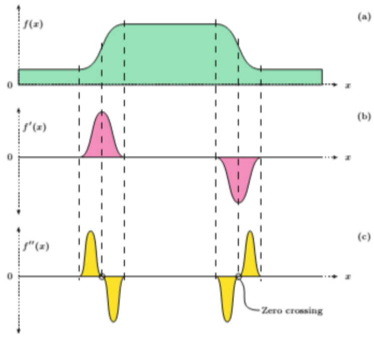

Figure 8: The second derivative

Image 8 shows the result for different derivatives. We can use the response from L_{vv} by calculating the zero crossing points and then checking that those points are negatives for the third derivatives are negative that we find a local maximum with the both conditions fulfilled.

Question 7

Present your best results obtained with extractedge for house and tools.

Figure 9: House edge-lines

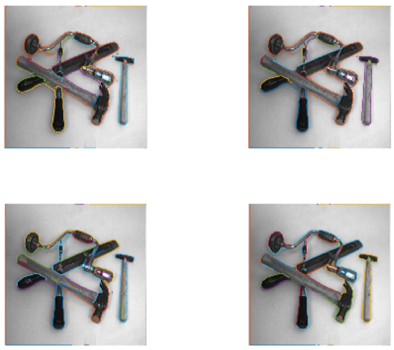

Figure 10: Tools edge-lines

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.

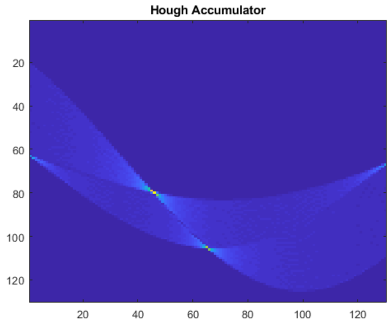

Figure 11: Accumulator

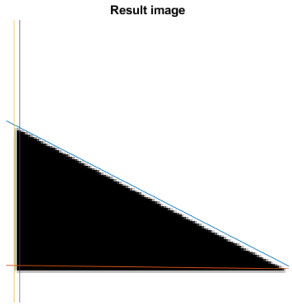

Figure 12: Resulting edges

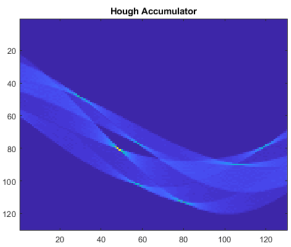

Figure 13: Accumulator

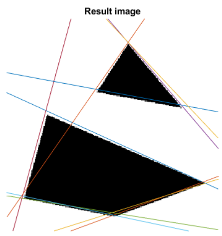

Figure 14: Resulting edges

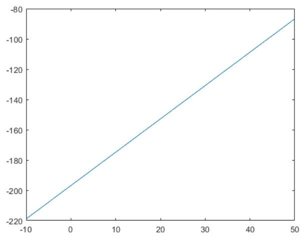

Figure 15: Strongest line

The strongest line has the following parameters :
$\rho = 81.3885$ $\theta = -0.4262$

$$Y = \frac{\rho - X * \cos\theta}{\sin\theta} \tag{3}$$

using the Hessian normal form line representation in equation (3), we can calculate the line representation in normal form, which is shown in figure 15.

Question 9

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

The complexity time for the given implementation is given by $N|\theta|$ where N is the length of the curves given as a parameter and θ is the number of angles to check in the accumulator ρ is calculated implicitly. For a small accumulator the computational time is small and the result less accurate, for a big accumulator the computation time also increase.

For a big number of θ we calculate the edges near to each other and for a small we calculate more distanced edges.

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?

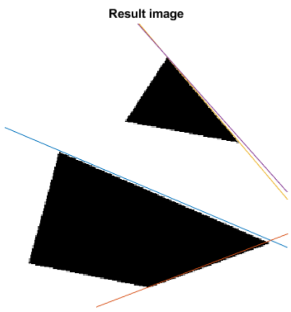

Figure 16: Result with increment by 1

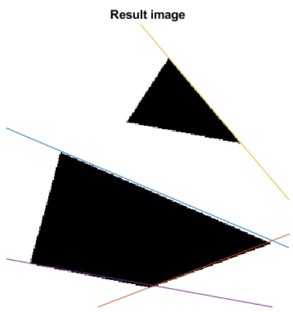

Figure 17: Result with increment by $log(|E(u, v)|)$

We see that when we increment the accumulator with the logarithm of the absolute value of the magnitude of the image, the resulting edges found are more accurate.

References