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

Name: *Javier López Iniesta Díaz del Campo* Program: *TSCRM*
Name: *Mathias Näreaho* Program: *TMAIM*

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?

We decided to use the Sobel operator (Eq. 1 and 2) [1] to perform the derivations.

$$\delta_x = \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix} \tag{1}$$

$$\delta_y = \delta_x^T \tag{2}$$

We expect dxtools to show the variations in the horizontal axis, and dytools to show the variations in the vertical axis. The reason the derivative-images are smaller is because we can not apply the Sobel operator to the edge pixels, so we instead remove them.

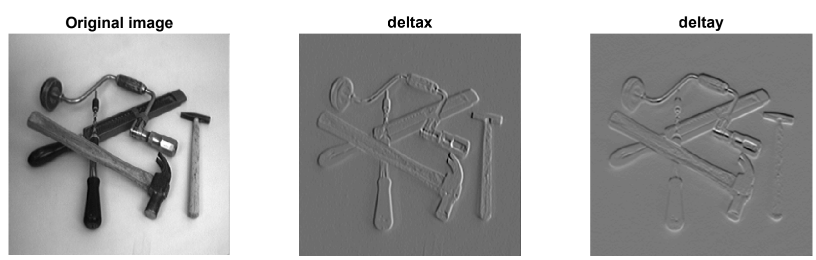


Figure 1: Difference operators.

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

Using the histogram method it is relatively easy to find a threshold, as we want to eliminate the scattered light spots around and on the tools, to only keep the edges. The histogram (Fig. 2) tells us where to find a cutoff point, as the bulk of the large histogram bins are representing the edges.

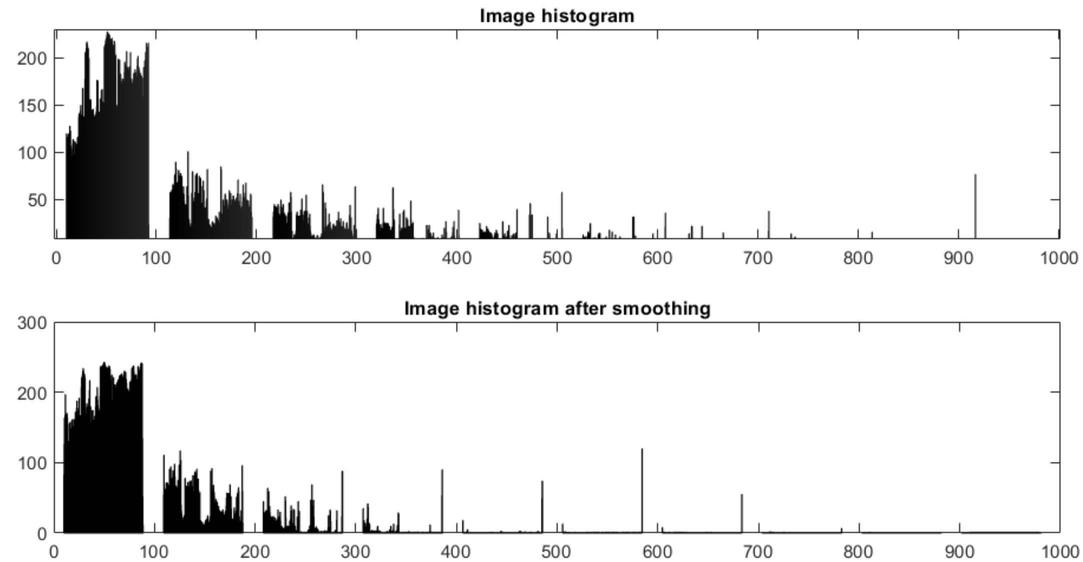
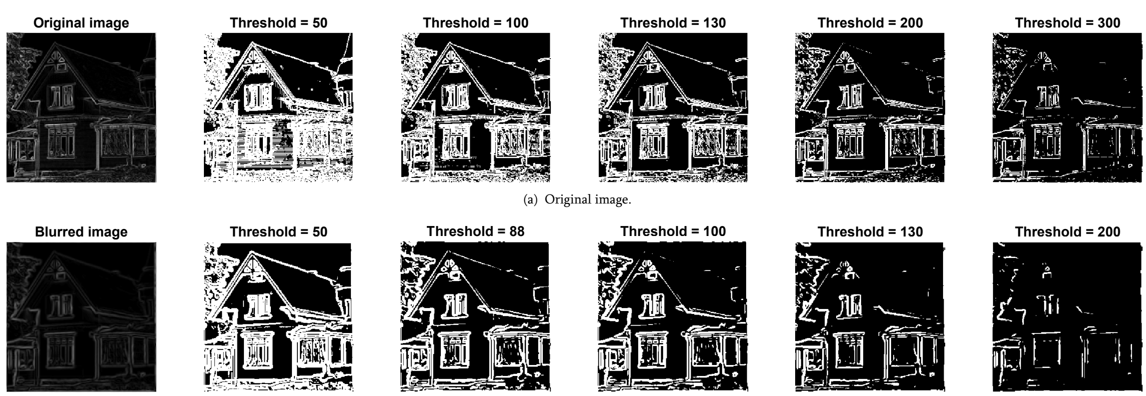


Figure 2: Histograms.

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

Smoothing with a low variance (e.g. $t = 2$) is very useful for finding the edges, but it may smooth them over as well, causing some loss of information. Without smoothing it is very hard to find the edges in the house image. Without smoothing we also get noise. These conclusions have been obtained after observing different threshold levels with and without smoothing the image (Fig. 3).



(a) Original image.

(b) Blurred image.

Figure 3: Point-wise thresholding of gradient magnitudes.

Figure 4 shows different smoothed edges that will disappear when we increase the threshold level. High contrast edges do not cause a lose of information.

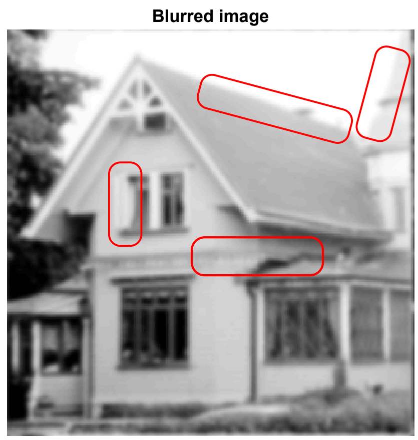


Figure 4: Smoothed image.

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

We can see that as we increase the blur we find less zero-crossings for the second derivative. Many of the zero crossings in the lower scales are not edges, but as we increase the blur we start to see distortions. The edges become warped and much information in the image is lost.

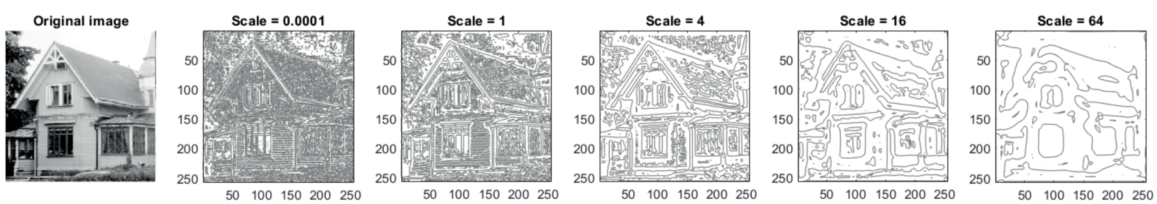


Figure 5: Second order derivative. Zero crossings of $\tilde{L}_{vv}$ on a number of different scales.

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

The sign condition creates a logical array of ones and zeros. Ones where the pixel value is less than zero and zeros elsewhere. This means that the white parts of the L_{vvv} plot in Figure 6 are the pixels of negative third derivative. For the smaller scales, the image becomes very grainy, but as the scale increases (more blurring) we can see that the shape of the tools becomes more apparent. As the scale increases further the edges become very wide, and the tools become harder to distinguish from each other.

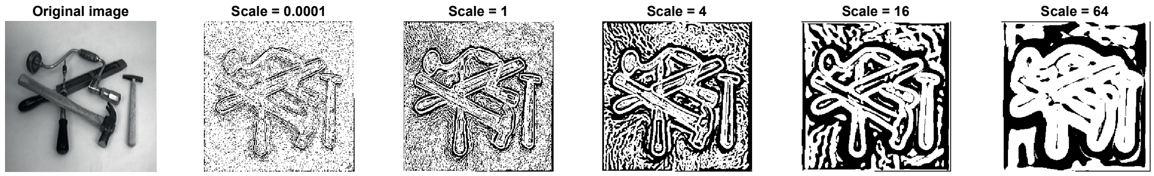


Figure 6: Third order derivative. Sign of $\tilde{L}_{vvv}$ in the gradient direction ($\tilde{L}_{vvv} < 0$).

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}?

While both L_{vv} and L_{vvv} detect edges, they must be combined for the best result. When L_{vv} is zero and L_{vvv} is less than zero, we have a local maximum in L_v (the gradient). When the gradient magnitude reaches a local maxima in the gradient direction, we know we must be at an edge.

Question 7: Present your best results obtained with extract edge for house and tools.

Figure 7 shows our best results when we extract the edges of the house and tools images. We have noticed that the best scale for extract edges is 4, whereas the threshold varies a bit with the image selected.

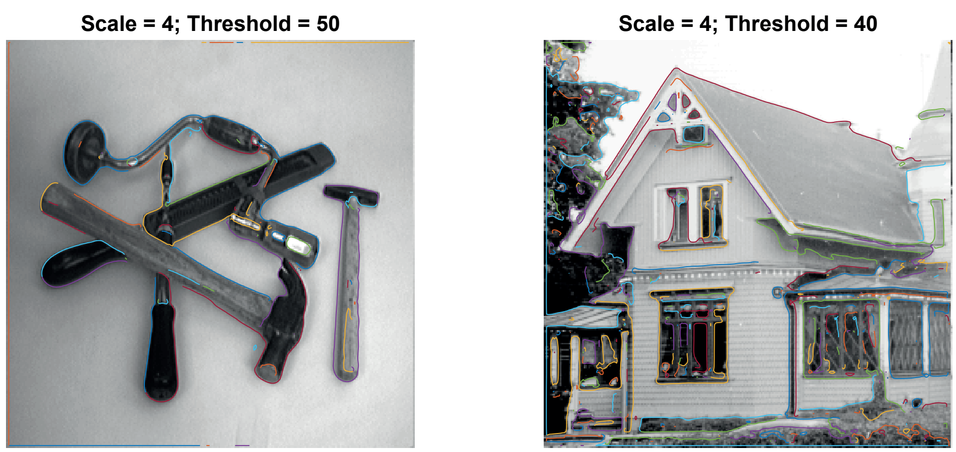
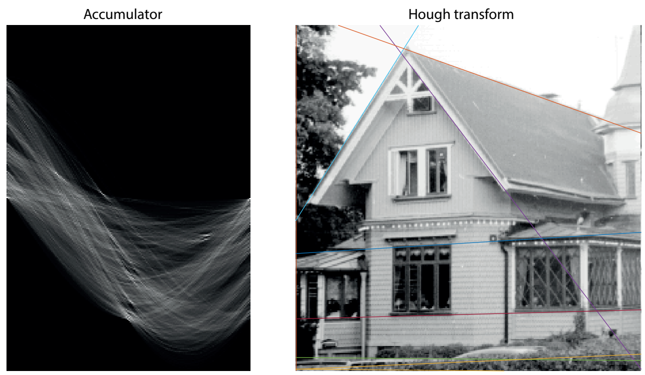


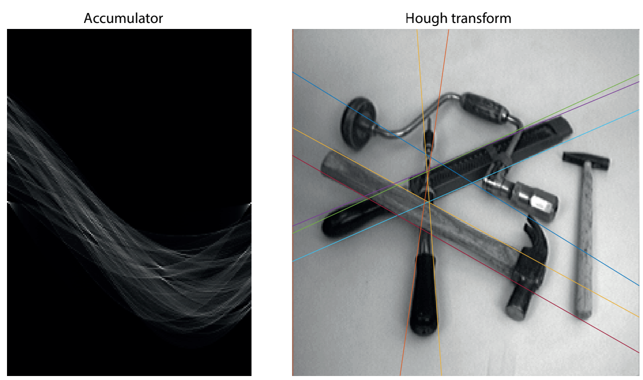
Figure 7: Best results for edge the house and tools images.

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.

It is clear to see that the strongest peaks in the accumulator represent the best lines in the final image. The accumulator space has the angle θ on the horizontal axis, with the zero point in the middle. This means that vertical lines correspond to peaks in the center of the accumulator space, horizontal lines correspond to peaks on the edges of the accumulator space, and tilted lines lie somewhere in between.



(a) `godthem256` image.



(b) `few256` image.

Figure 8: Accumulator and line segments for two different images.

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

The number of cells in the accumulator indicates how many lines we are including in the algorithm. More possible lines means longer computational time (more voting) but may also improve performance.

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?

This can be done by using the magnitude object that is sent as an input to the `houghline` function. We decided to try this while taking h to be the natural logarithm of the magnitude (Eq. 3). Our results did not change much, it moved some lines around but kept most of them the same.

$$\Delta S = \log(|\nabla L|) \tag{3}$$

(a) Accumulator update with +1. (b) Accumulator update with $\log(|\nabla L|)$.

Figure 9: Comparison between different accumulator incrementation function.

References

[1] R. Fisher, S. Perkins, A. Walker, and E. Wolfart, "Sobel Edge Detector," 2003, [Online] Available: https://homepages.inf.ed.ac.uk/rbf/HIPR2/sobel.htm [Accessed: 18-11-2022].