Answers to questions in

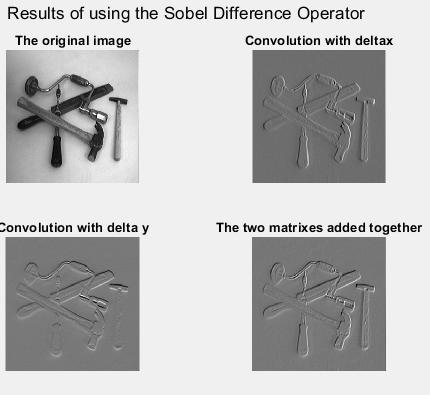
Lab 2: Edge detection & Hough transform

Name: David Dashti Program: CMEDT

**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:

They are done perpendicular to each other so one is in the x-direction and the other is in the y-direction based on how the convolution kernels are designed. So, one will show all derivate changes in x-direction on a greyscale and the other will show all derivate changes in the y-direction on a greyscale.



The reason why the sizes differ is because you are doing a convolution of a 256 \* 256 image with a 3 \* 3 convolution kernel. Effectively returning a 254 \* 254 image. In the function for convolution the argument ‘valid’ is set which means it will return an image where the values are not computed with zero-padding.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Answers:

I found it quite hard. The answer why is twofold.

The image itself has some inherent high frequency noise. When a first order derivate filter is convolved with that image, the noise creates big changes in the derivate. This is because noise is “edgy”. Thus, there will be high intensity values in the histogram purely because of noise.

Secondly, if the image is a very smooth one, then the histogram will be more spread out. Making it hard to find thin edges since the histogram distribution is so spread.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Answers:

Both yes and no. This question is linked to the answer in Question 2.

It helps to counteract the fact that there is high frequency noise which in turn makes it hard to threshold. So, you remove those artefact edges. However, you are at the same time smoothing out the histogram and thus creating thicker edges.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Answers:

What we get is a representation of local extremes in the gradient direction. These are in different scales which refer to increase variance in gaussian smoothing of the image.

When the scale is increased too much, we lose dominant edges and when the scale is too small, the image becomes noisy.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Answers:

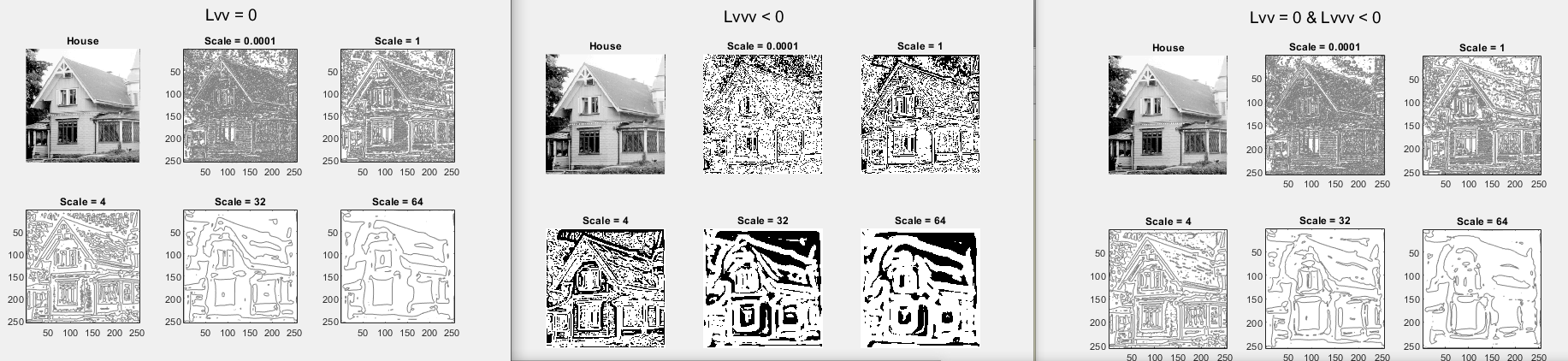


Figure - Results of experiments to find edges on the image nicknamed house

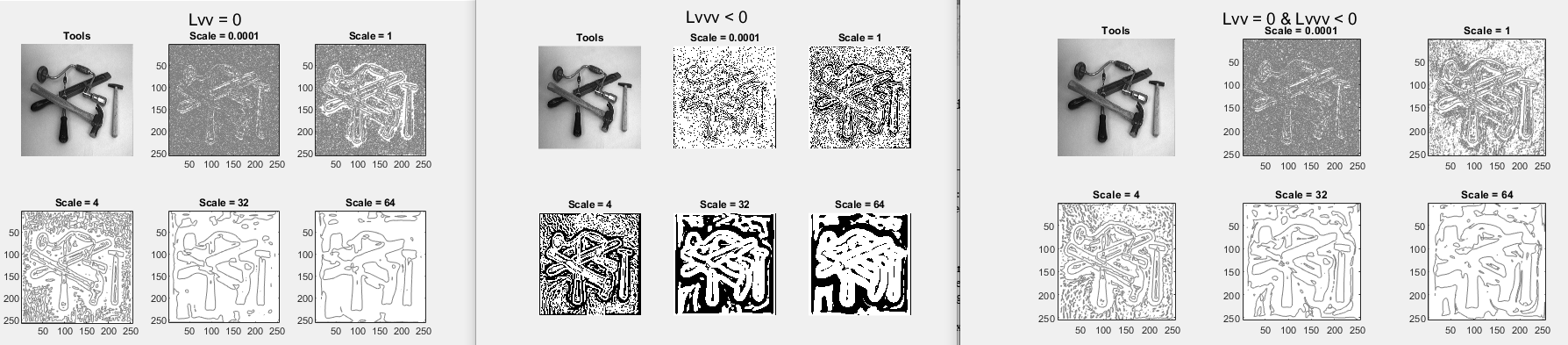


Figure - Results of experiments to find edges on the image nicknamed tools

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Answers:

Both work on their own but when combined they are stronger. Lvv can produce of a lot of extra lines because it is very sensitive to noise. This is because there is a lot of zero crossings in noisy images. Lvvv in turn adds thick white lines where Lvvv < 0.

But having a criterion that takes in account both, should give you only lines or at least a better approximation of where the lines are. Since you know that you have centered the line when both criterions are met.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Answers:

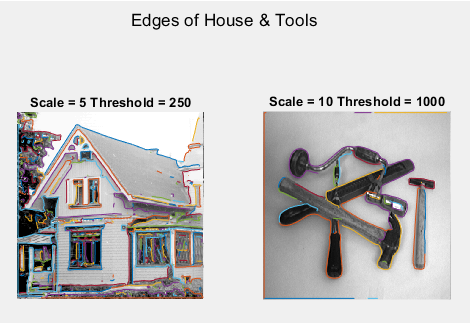
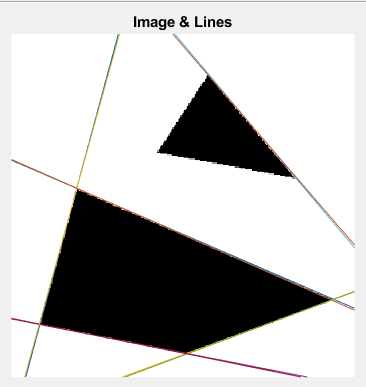


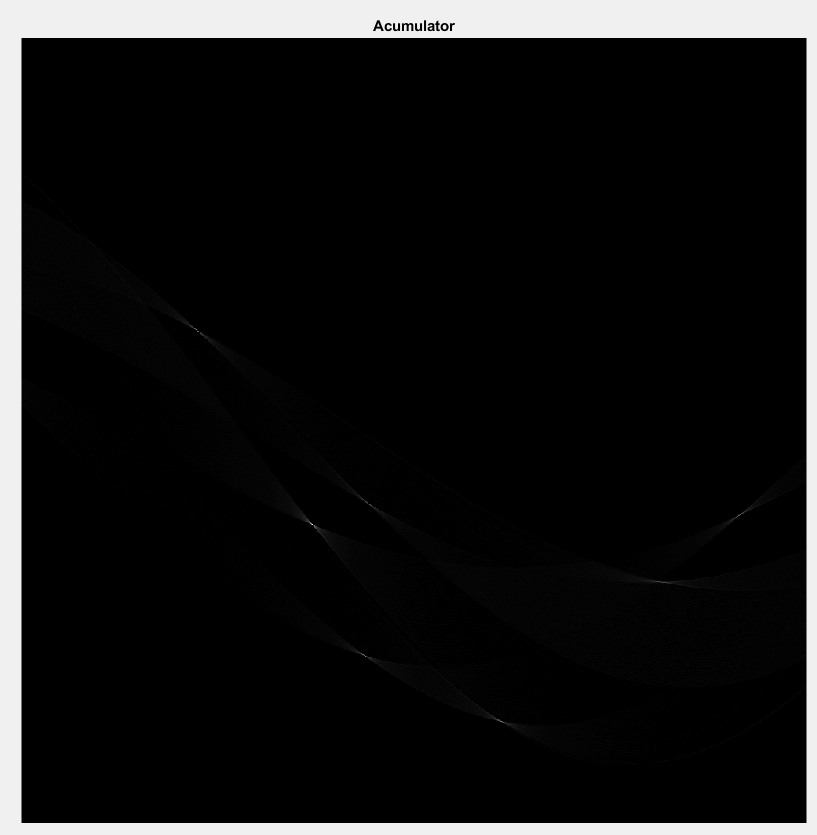
Figure 3 - My best results with extractedge for tools and house

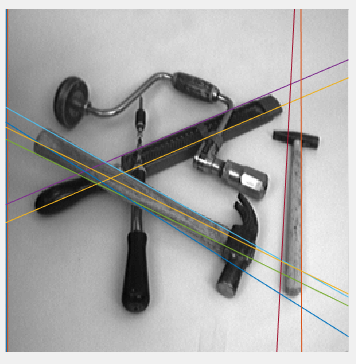
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

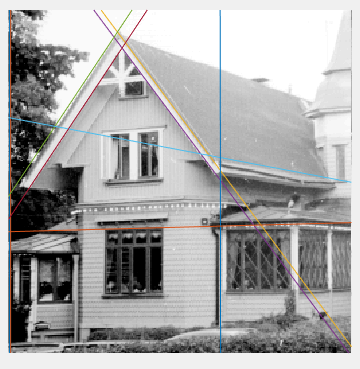
**Question 8**: Identify the correspondences between the strongest peaks in the accu-mulator and line segments in the output image. Doing so convince yourself that the implementation is correct. Summarize the results of in one or more figures.

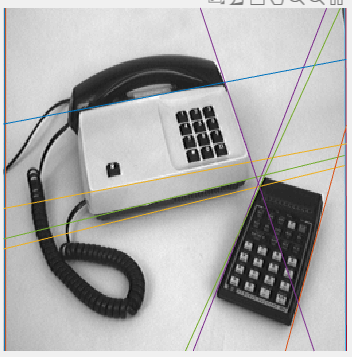
Answers:











\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

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

Answers:

The general answer is that an increasing in the number of cells will cause an increase in the computational time, which is kind of expected. However, the amount of θ cells will affect the computational time in further extent than the amount of ρ cells. Increasing the amount of cells will on the other hand increase the performance of the algorithm.

Decreasing the number of cells will speed up computational time. Also, the accuracy in both direction and placement will be negatively affected if the number of cells drastically decreases. On the other hand, if the number of cells is way too big, you will get a high number of lines from the same maxima.

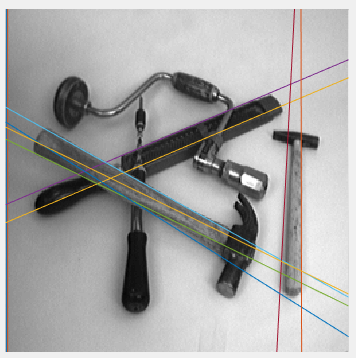
When looking at theta, if the number of cells is high then you will get a lot of points in an area referring to the same line. Thus, your accuracy in edge identification will increase because the vote is higher. However, as said above, the computation takes longer because the inner loop becomes “bigger”.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**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:

**Tools with regular voting**



**Tools with logarithmic voting**



What you could do is a weighted voting where you weigh votes high is the strength of their gradient magnitude is high. Using a logarithmic function will cause lower intensity magnitudes to have a bigger chance of creating a detected edge.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_