Lab 2

Name: Neil Pradhan

Email: npradhan@kth.se

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?

Ans 1:

The results are as follows:

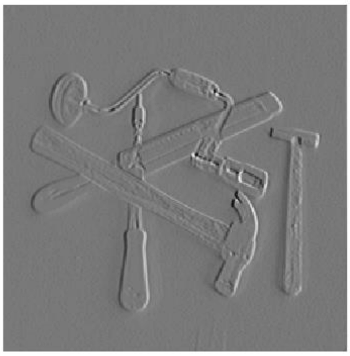


Figure: (left) denotes the delta_x = 0.5*[-1,0,1] where as the (right) denotes delta_y = transpose(0.5*[-1,0,1])

Observation:

Most part in the image will have zero grey value. In the left image the vertical direction has better edge separation and in the right image the horizontal direction has the better image separation.

Because we use the command conv2 with the shape as “valid” it returns the subset of the convolution without including the values obtained from the zero padding of the image. Hence the dx tools and dy tools has difference in dimension. As the delta_x has dimensions 1*3 while we slide through the image for convolution 2 columns are left out or need padding therefore, we have the final image after convolution and neglecting padding of the size 256* 254.

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

Ans:

The choice for edge kernals can be varied, but here I have chosen difference operators(dx = [1,-1] and dy = dx’ to get the results for various thresholds. For the gaussian kernel the standard deviation has been chosen to be 0.1. Also various other operators such as sobel or central difference operators were tried but here only for demonstration purpose I have included only the simple difference operator.

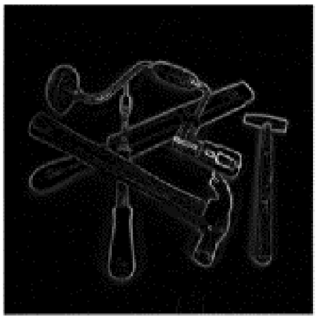


Figure : Image without thresholding

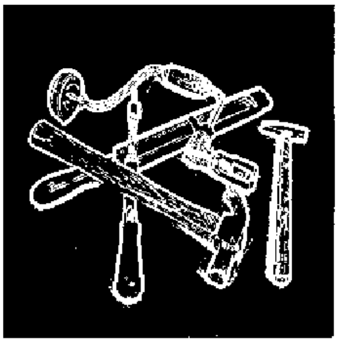


Figure: Image with thresholding =20

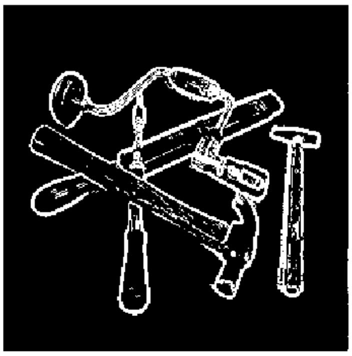


Figure: Image with thresholding = 30

Observation :

A threshold with thin edges is difficult to find because Edge detection is a trade-off between thresholding and gaussian smoothing. Gaussian Smoothing removes noise but blurs edges, and thus making edge localisation difficult. Therefore, we need to choose an appropriate value between standard deviation and threshold in an image for edge detection and localisation. Then there is difficulty due to unwanted noise.

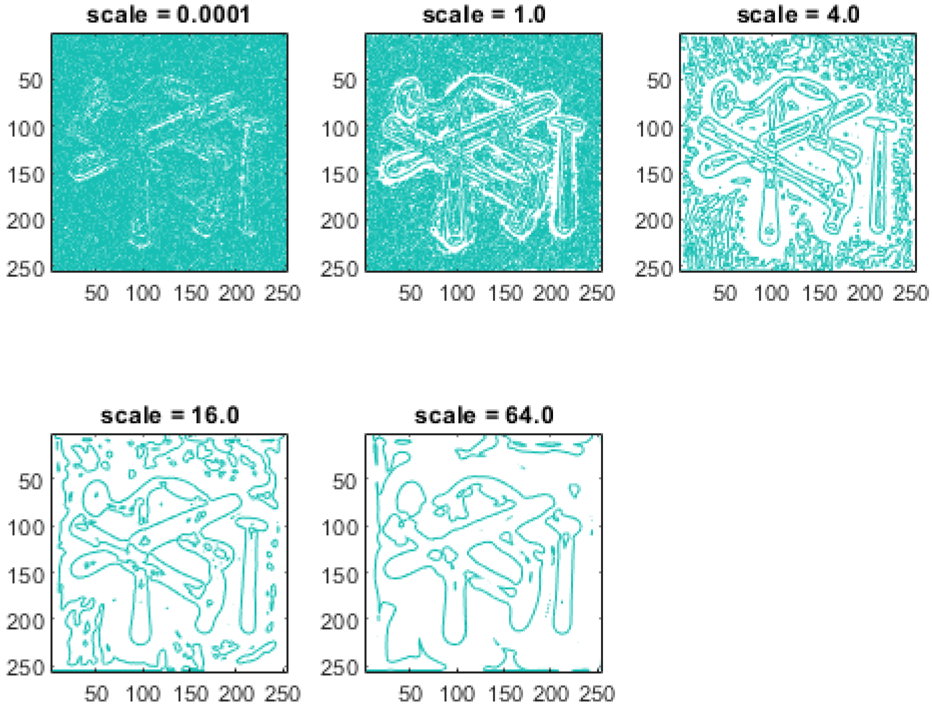
Question 3: Does smoothing the image help to find edges?

Answer:

Smoothing the image with a constant threshold can help remove high frequency noise but it creates problems in deciding the threshold as the pixel values are accumulated over the mean and edges now have less difference than before. Hence smoothing should be carried out with care and simultaneously thresholding should not be overlooked at as smoothing and thresholding are a tradeoff in an image.

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

Answer:

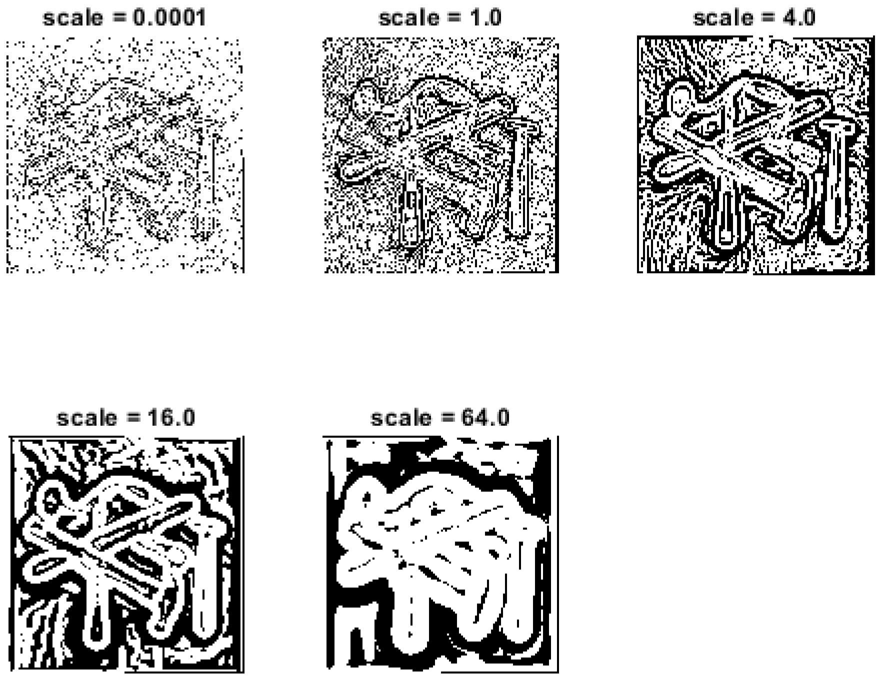


Observation :

When the scale is small the noise is huge and many curves show intricate details where as in case of large scale edge curves are more cleaner however when the scale is very large there is loss of edge information. When the scale is small we have several zero crossings due to meniscal changes in intensity and thus noise is also the part of the figure, where as when the scale is large there is the smoothening effect due to which edges are clear and noise is rejected but when the standard deviation or scale is very large it tends to over smoothening which tends to blurring of images and thus distorting edges.

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

Answer:

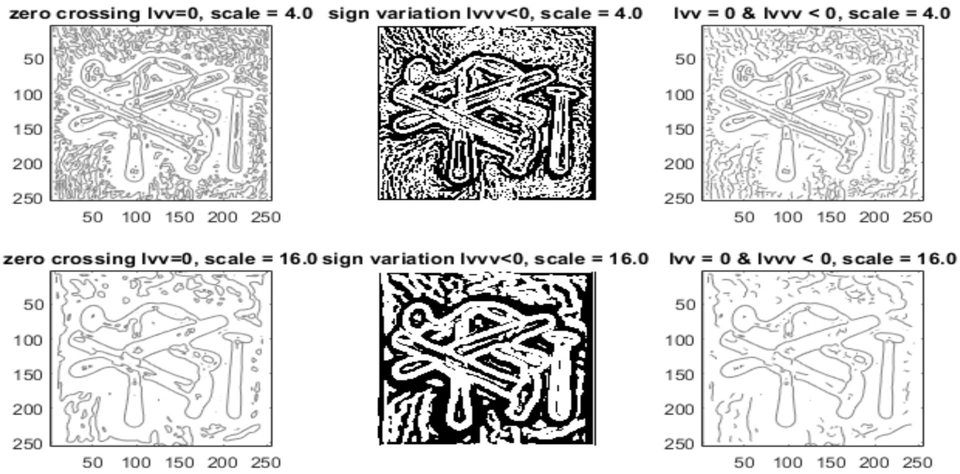


Observation:

As the Gaussian smoothening increases, we see that the edges are blurred. White area in the image indicates pixels with negative third order derivatives. Thus, we see that smoothing in a limit will help in getting the edges however over smoothing leads to loss of information of the image with regards to the third order derivatives.

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

Answer:



We can look at the zero crossings in lvv =0 to detect edges and also simultaneously check if lvvv is negative, only then true edges will be detected. Also that selection of right scale of the gaussian and thresholding for the 2nd derivative is essential in improvement of edge detection procedure.

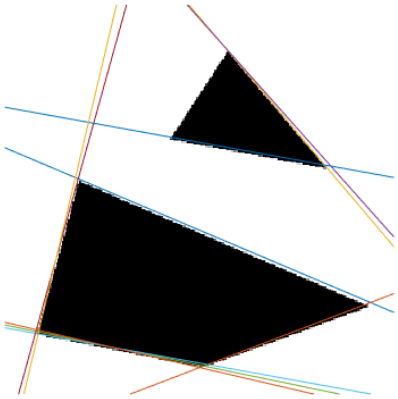
Question 7: Present your best results obtained with *extractedge* for *house* and *tools*.

Answer

In the above figures the house parameters are scale=3 and threshold =6 and for the tools the scale =3 and the threshold=5.

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.

Answer:

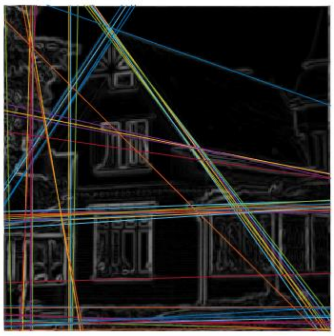


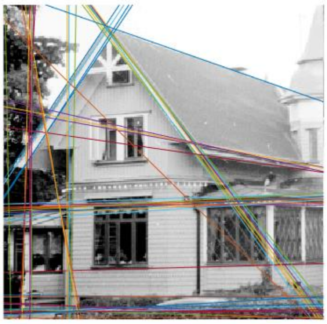
In the above hough line transform we see 7 peaks which correspond to the 7 highest votes in the accumulator matrix that gives us the edges required in the figure. For obtaining this hough transform the scale used was 1, threshold was 50, while the number of divisions for rho and theta were 256 and number of lines 10. We can find the corresponding rho and theta for the lightented points in hough transform to get equations of line in image domain.

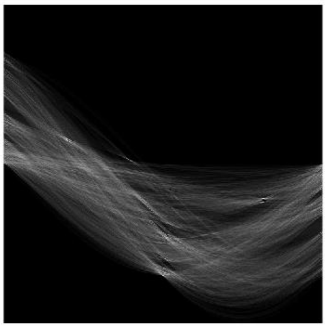
Question 9: How do the results and computational time depend on the number of cells in the accumulator?

Answer:

When we increase the accumulator size we get much accurate edge detection but the computational power increases. Also smoothening although blurs edges in a limit can enhance the image in Hough transform domain thus letting us find the corresponding lines easily.







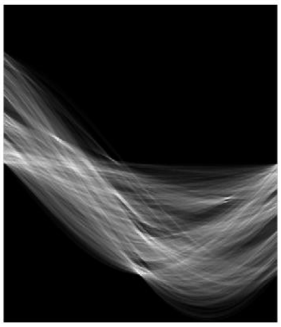


Figure: In the above Images the upper left is lines draw as a result of hough transform and in the lower left and right we see the hough transform without and with smoothening operation for detecting lines in the image.

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:

We can have several monotonously increasing functions such as logarithmic, algebraic (square of a fucnction) or square root under positive limits and so on, The function that we must choose depends on what we want to achieve and our input image .For instance, if we consider logarithmic function, the initial rate of rise is slow but for higher values it is high. Thus We must first see the histogram of the image and then decide how should we go about choosing the monotonuous function for voting for a line or figure for generalised hough transform. It is just another way of adding a weight to the accumulator matrix which otherwise in the previous questions was considered as the same.

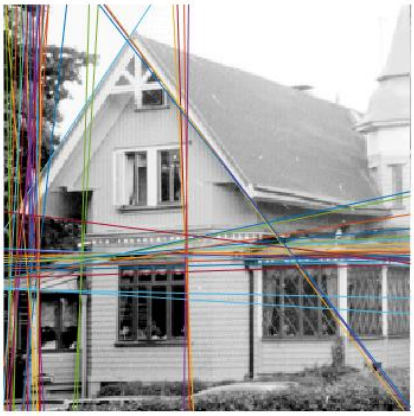


Figure: For increment in accumulator array as the square of pixel magnitude

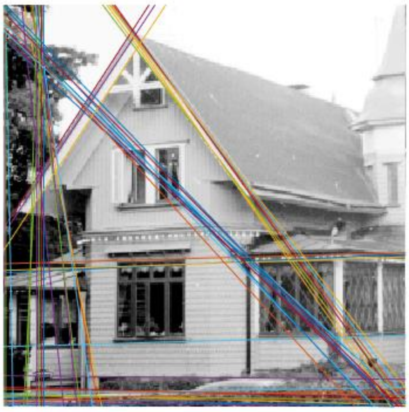


Figure: For increment in accumulator array as the logarithm of the pixel magnitude

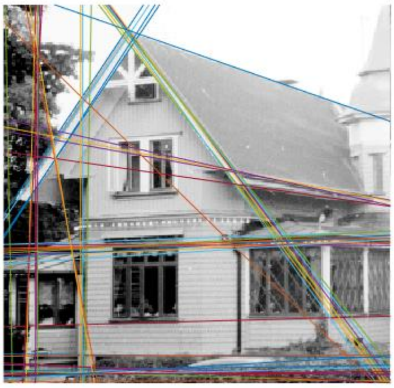


Figure : For increment as a constant =1 for every addition in accumulator array

Thus we observe that there should be a trade off between threshold and the choice of the monotonously increasing function, as the function will affect the threshold. Also noise plays an important role in the choice of the incremental function. For if logarithm function doesn't work well if there is a lot of noise in the lower pixel intensity values which is not good for edge detection as it will accept the noise as the point in image.