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

Name: Timotheos Souroulla Program: Systems, Control and Robotics

Instructions: Complete the lab according to the instructions in the notes and respond to the questions stated below. Keep the answers short and focus on what is essential. Illustrate with figures only when explicitly requested.

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:

When we convolve the original image with the Delta_x, then in the result, all the image discontinuities in the x direction will be highlighted. Same result is expected when convolving with Delta_y, but it will highlight the discontinuities in the y direction. This is because of the shape of deltax and delta y signal (which the matrix deltax and deltay signify).

Delta_x and Delta_y were created using the Sobel filter as shown below:

Delta_x = $\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$ Delta_y = $\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$

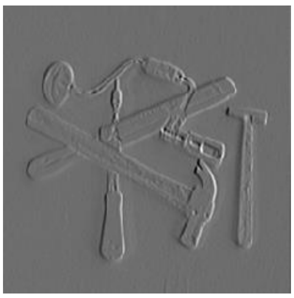
Here we can clearly see that convolving with deltax will ignore any sharp changes in pixel values in the y direction (because the -1 and 1 will negate each other) whereas the changes in the x direction would be highlighted.

If size of 'tools' was m * n then the size of 'dxtools' becomes (m-1)*(n-1).This is because when 'tools' gets convolved with deltax (which is a 3*3 matrix) then the border pixels from all the 4 borders are used up to calculate the updated pixel value for the inner row or column on each side.

Size(tools) = 256 256 Size(dxtools) 254 254

Original Image

Dx Operation



Dy operation

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

Answers:

It is easy to find a threshold with thin edges manually (trial and error method). This is because when we obtain multiple images after applying a range of thresholds, we can easily see that there is a threshold beyond which the edges start disappearing and before which the edges are too thick.

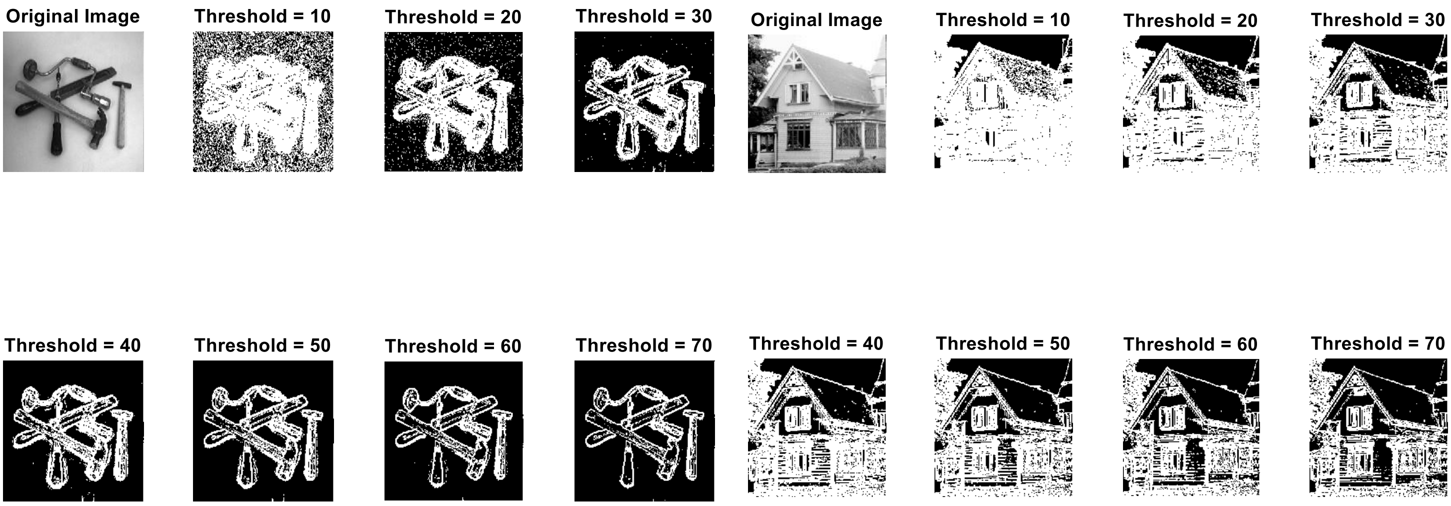
But when we try to set a threshold without plotting the resulted images, this task is proved to be hard. One could use some methods proposed, but this will be time consuming.

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

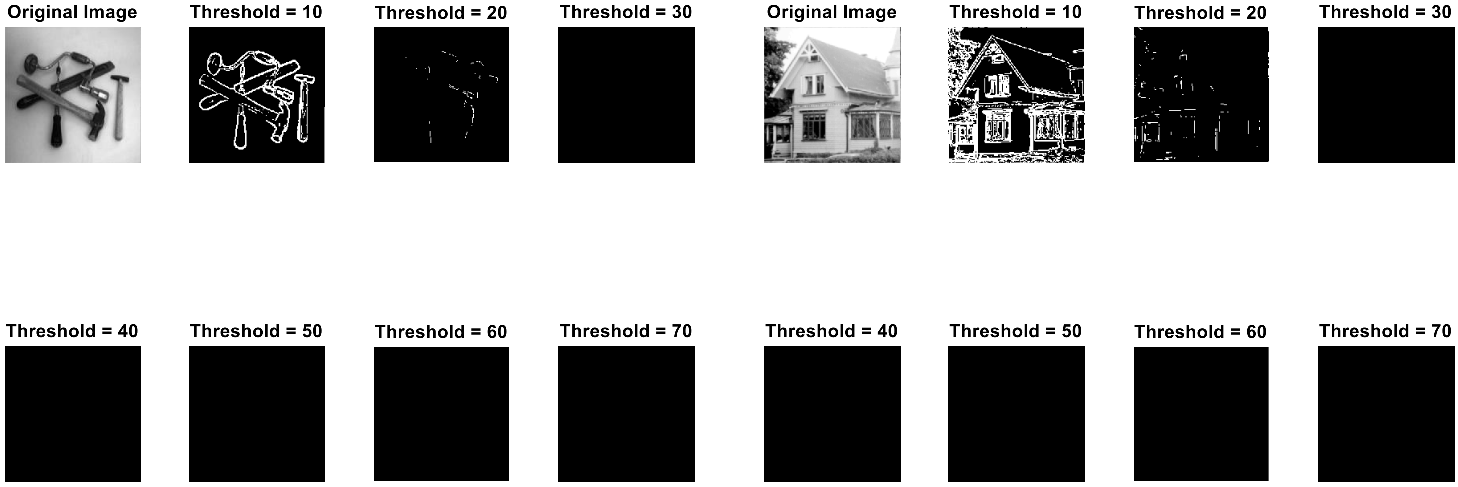
Answers:

Yes, smoothing does help to find the edges better. This is because smoothing helps to eliminate the variations inside an object e.g. the hammer in the below image gives a better edge when it is smoothened first, because the pixel variations on its body are eliminated because of smoothing. Using smoothing also helps to eliminate noise and discontinuities due to shading.

<u>Without Smoothing:</u>

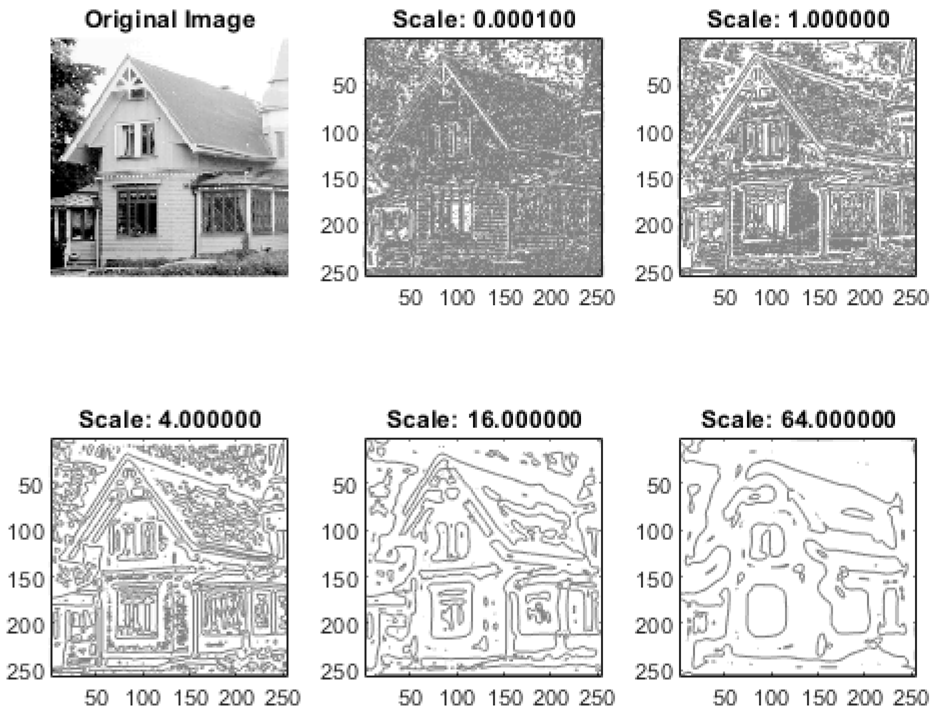


After Smoothing:



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

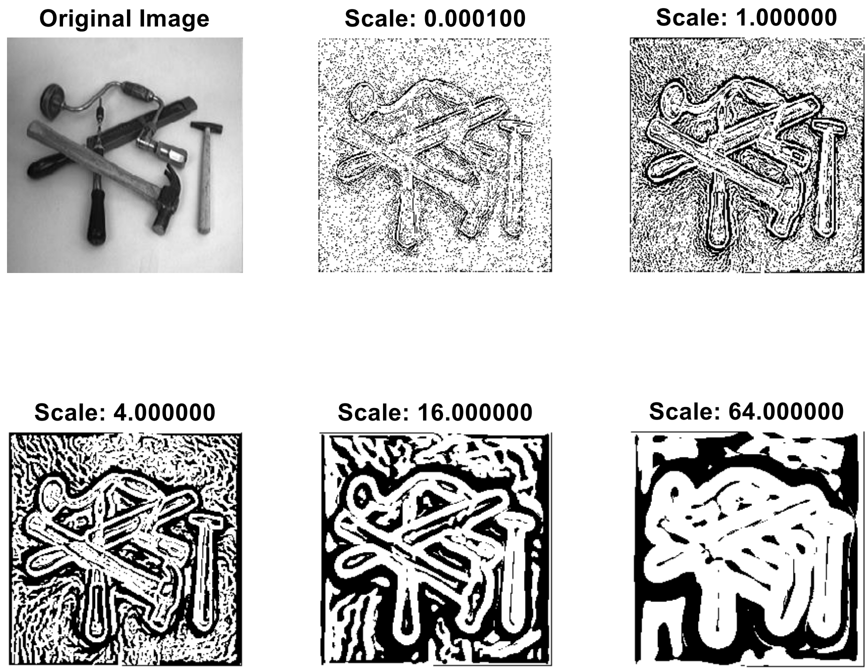
Answers:



We observe that second order derivative highlights the places where there is an edge in the image which would correspond to the basic principle of the second order derivative of being low for a smooth signal and high for a spike signal. The scale of the gaussian influences the results such that more the scaling, more the smoothening and consequently we start losing out details of the edges.

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

Answers:



The third order derivative helps us highlight only the local maxima in the image. This is because Lvvv < 0 corresponds to those points in the signal which are having a negative slope after a spike.
The result is images where we can see the contours well, however they are not as clear as for the second derivative.
Lower value of scale results in a detailed image but with many black and white dots.
Higher scale value results in a smoothened image.
We can find a scale value (not too high or too low) because of which the Lvvv has detailed borders along with lower noise.

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

Answers:

The second order derivative Lvv highlights the points at which there is a transition from one-pixel value to another which represents the edge in an image. These points could either be local maxima, or local minima.
Whereas the third order derivative highlights points that have a negative slope after a spike. Consequently, local minima are eliminated, and the result is only local maxima after using both second and third derivative.

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

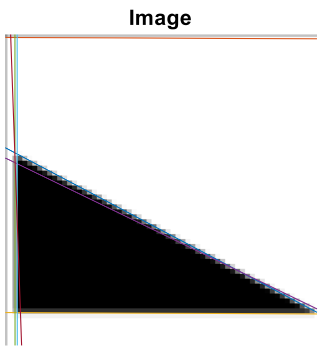
Answers:

Threshold = 30
Scale = 4

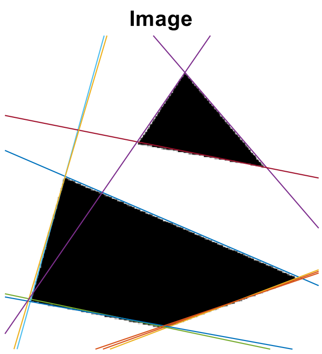
Threshold = 20
Scale = 4

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.

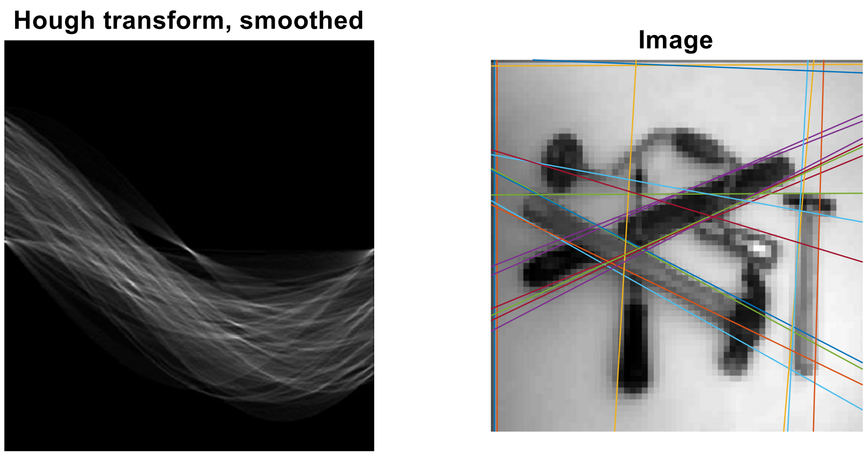
Answers:
`triangle128:`



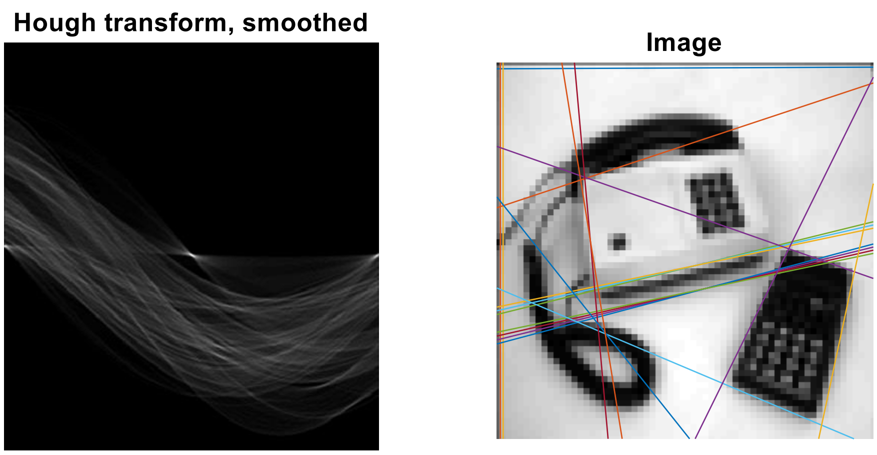
`houghtest256:`



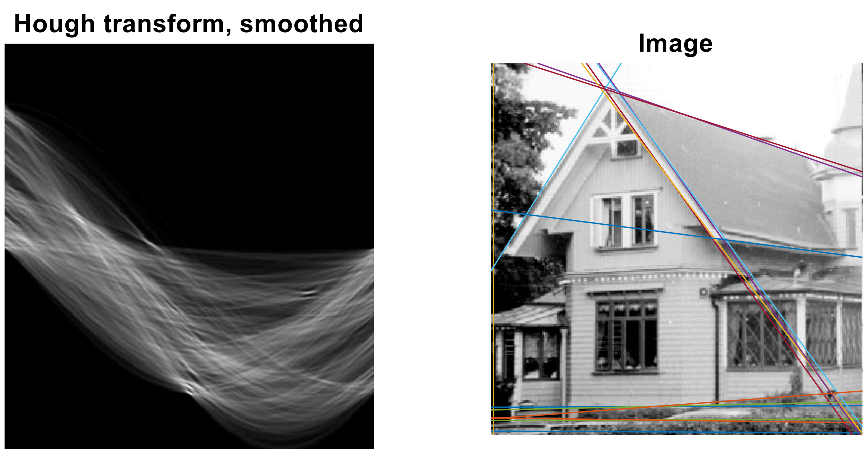
Few256:



PhoneCalc256:



Godthem256:

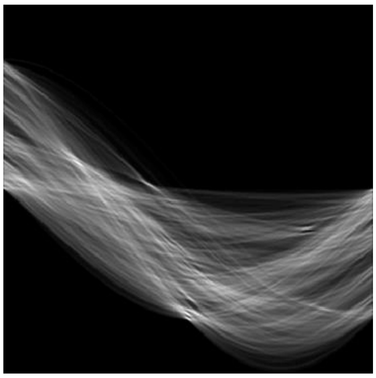


Question 9: How do the results and computational time depend on the number of cells in the accumulator?

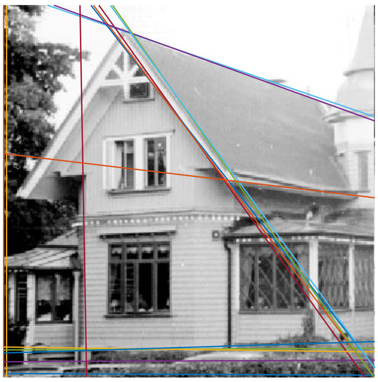
Answers:

```
Same Theta and Distance
Elapsed time is 4.202901 seconds.
```

Hough transform, smoothed

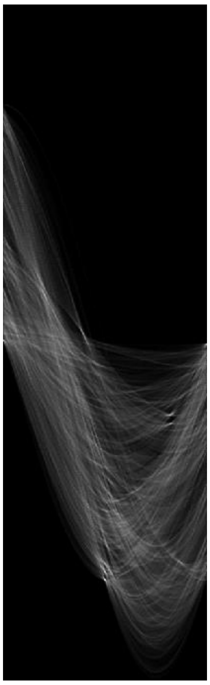


Image

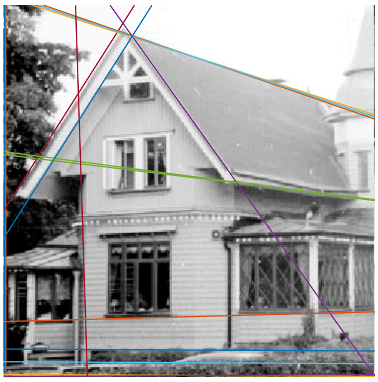


```
Low Theta and High Distance
Elapsed time is 5.598104 seconds.
```

Hough transform, smoothed

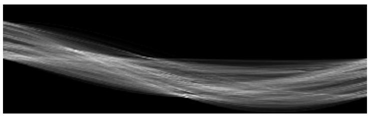


Image

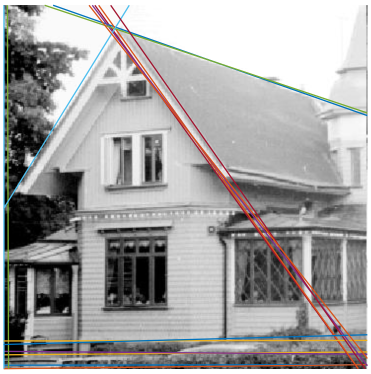


```
High Theta and Low Distance
Elapsed time is 9.870187 seconds.
```

Hough transform, smoothed



Image



We observed that the number of thetas affect the computational time higher than the number of rhos.
This might be because it must look for given amount of rhos for large number of thetas which is more time consuming than looking for more number rhos for given number of thetas.

More thetas result in good detection of edges close to each other whereas more rhos results in better detection of continuous 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?

Answers:

Log (Gradient Magnitude):

Image

Exp (Gradient Magnitude):

Image

For this question we used two monotically increasing functions, log(x) and exp(x). According to theory, the accumulator space must be increased by one at each time step. So, we first used log(x), which is approximately one for most of input numbers, and the results were close enough to the results obtained when the accumulator space was updated by one.
Afterwards, we used exp(x) function, which has greater values than 1, and the results obtained were not as good as when the accumulator space was updated by 1.