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

Name: Serkan Arda Yılal
Program: Msc. Information and Network Engineering (TINNM),

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.

Good luck!

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:

In this step, we have used Sobel operators for partial derivatives which are the following matrixes:

$$\text{deltax} = \begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}, \ \text{deltay} = \begin{matrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{matrix}$$

After we create deltax and deltay which are x-wise and y-wise partial derivatives, we should expect x-wise edges of tool image from dxtools and y-wise edges of tool image from dytools. Because the values in these matrixes approximates the horizontal and vertical changes. And we can see it as in the following:

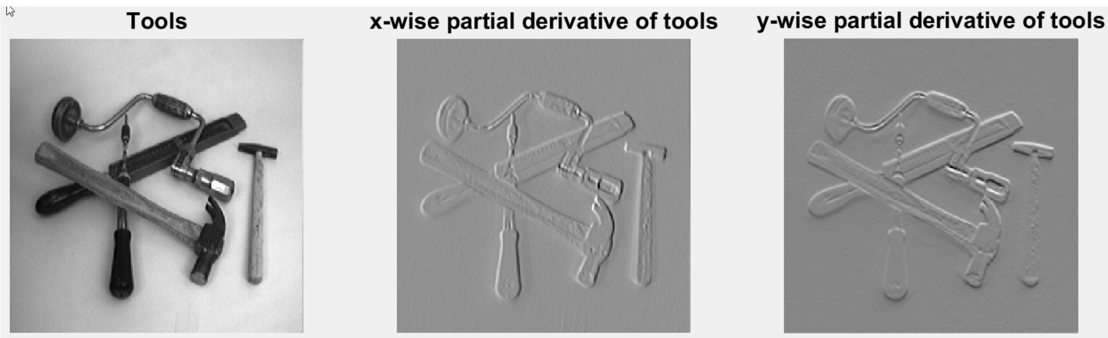

Figure 1.1-X-wise and Y-wise derivative of Tools with Sobel Operators

If we compare the sizes, we will see that dxtools and dytools has 2 missing rows and columns compared to tools:

Figure 1.2-Sizes of dxtools, dytools, and tools for comparison

The reason is that deltax and deltay requires 8 neighbors of a center pixel point. Since the border and corner pixels do not satisfy this requirement due to some of their neighbors will be missing, partial derivatives of the first and the last rows and columns of the image cannot be calculated properly so they are not included in dxtools and dytools.This approach is set because we have chosen "valid" argument as input which has the following meaning:

'valid' — Return only parts of the convolution that are computed without zero-padded edges.

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

Answers:

In this exercise, we have calculated gradient magnitudes of the images and checked their histogram for a correct threshold value selection. But since there was no minima that could have been used for a good threshold selection, we have instead applied different thresholds and tried to see which one is a proper value.

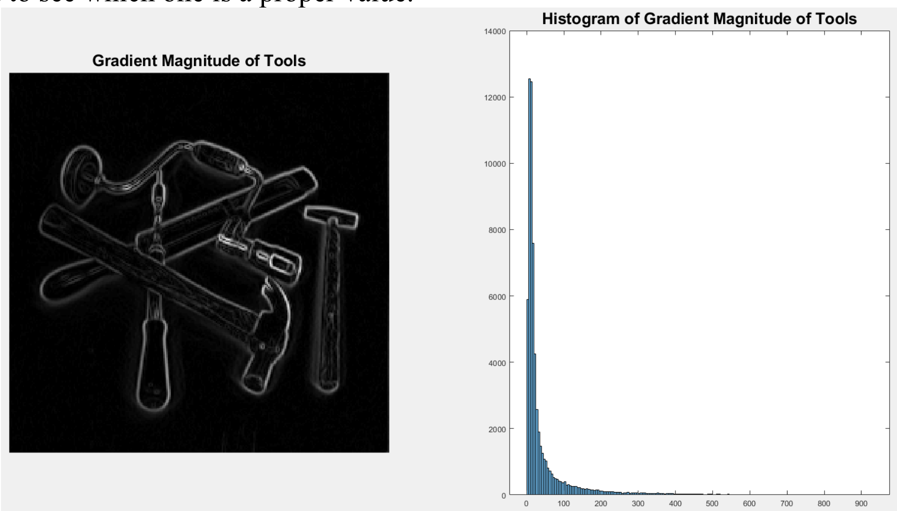

Figure 2.1-Gradient Magnitude and its Histogram for Tools image

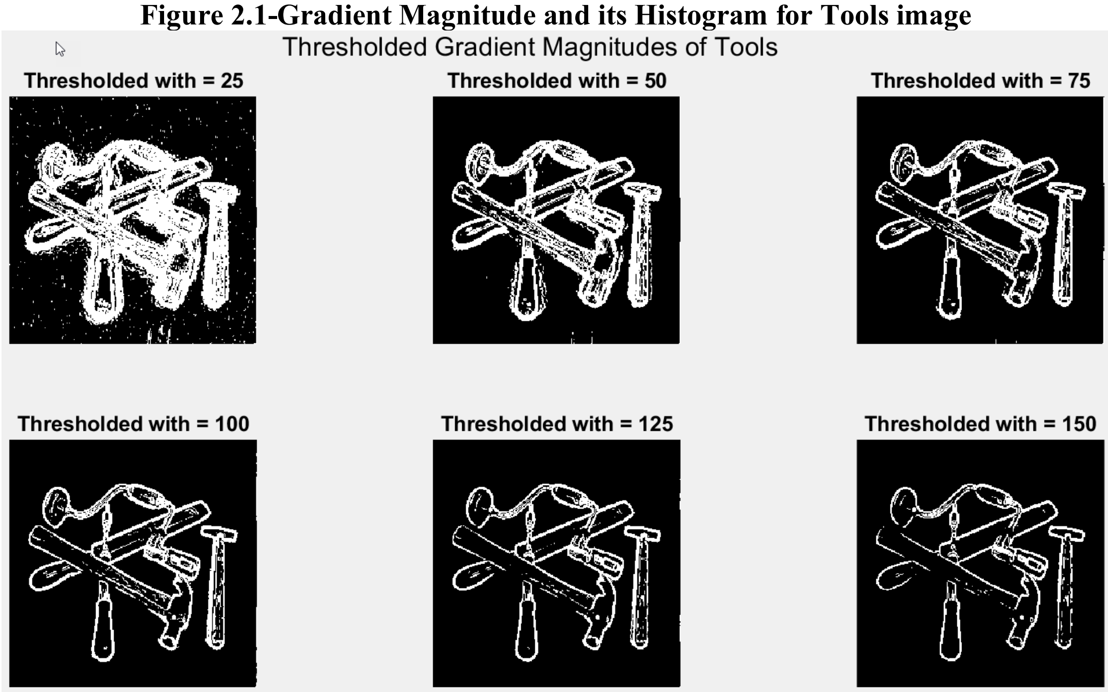

Figure 2.2-Gradient Magnitude and its Histogram for Tools image

As it can be seen, as we increase the threshold value, the edges start to become thinner. But the problem is that even though we get rid of false positives due to noise, we also start to lose weak edges due to that threshold. So, we can consider as a trade-off and select the threshold as 100 for that image since it has thinner lines but still preserves the weak edges as much as it can.

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

Answers:

We can definitely see that smoothing helps us to remove the noise and make the true edges remain on the image since we are removing higher frequencies from the image. On the other hand, if we have a high smoothing then we may also start to distort true structure of the image which will lead to lose information about true edges. Because we will start to remove frequencies at lower values which may contain core content of the image.

In following example, gaussian filter is used for smoothing with the variance value of 2. The threshold values are decreased in this example for comparison since smoothing is applied.

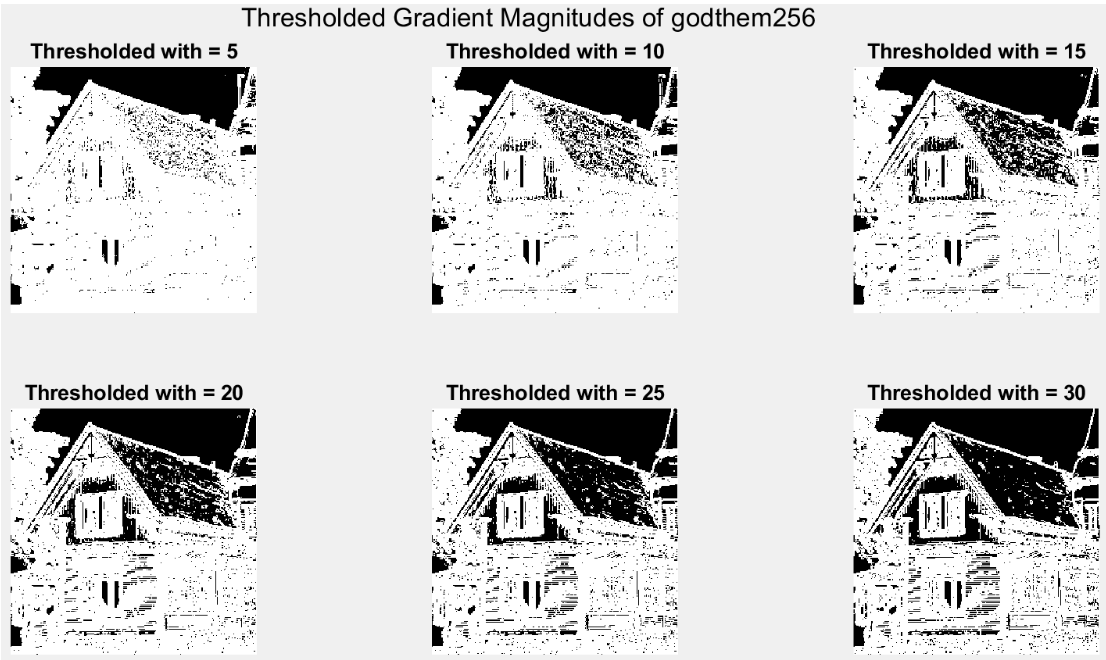

Figure 3.1- Gradient Magnitudes of image godthem256 with different thresholds

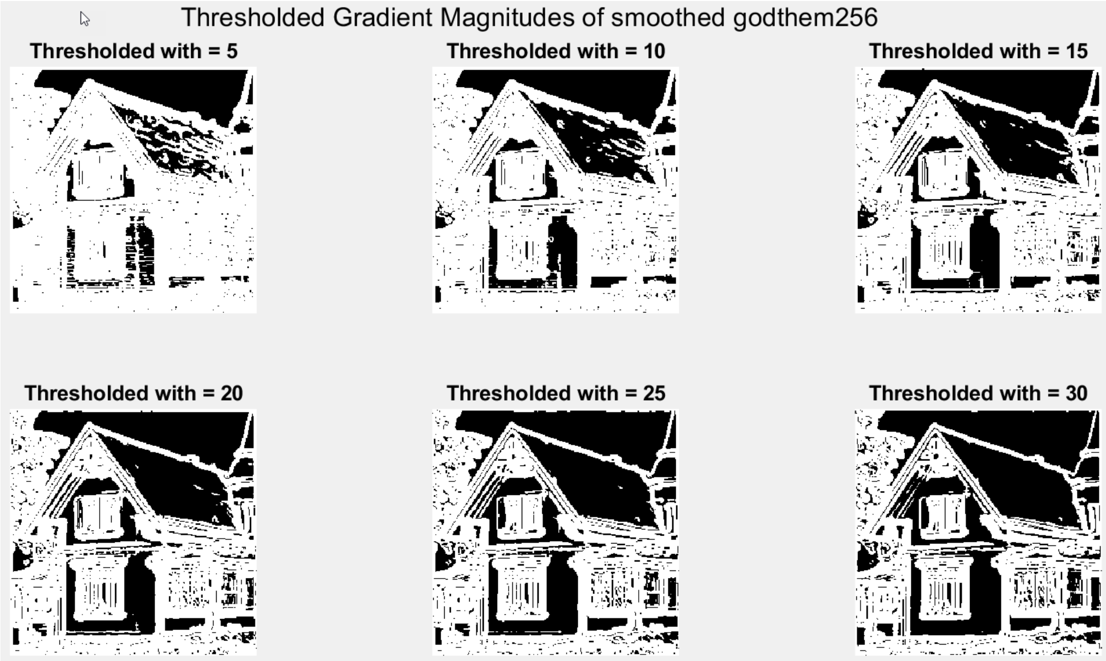

Figure 3.2- Smoothed Gradient Magnitudes of image godthem256 with different thresholds

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

Answers:

In this exercise, we have tried to obtain thinner edges by simply using smoothed intensity function L and the conditional definition of edge detection with consideration of second and third derivative of L as in the following:

$$\begin{cases} L_{vv} = 0, \\ L_{vvv} < 0, \end{cases}$$

By defining Lvvtilde and Lvvvtilde functions in matlab environment, we became able to calculate these derivatives under following simplified definitions:

$$\begin{cases} \tilde{L}_{vv} = L_v^2 L_{vv} & = L_x^2 L_{xx} + 2 L_x L_y L_{xy} + L_y^2 L_{yy} = 0, \\ \tilde{L}_{vvv} = L_v^3 L_{vvv} & = L_x^3 L_{xxx} + 3 L_x^2 L_y L_{xxy} + 3 L_x L_y^2 L_{xyy} + L_y^3 L_{yyy} < 0. \end{cases}$$

Just before applying these functions, we applied a gaussian filter with different variance values with the purpose of giving a scaling effect to the image we are experimenting on.

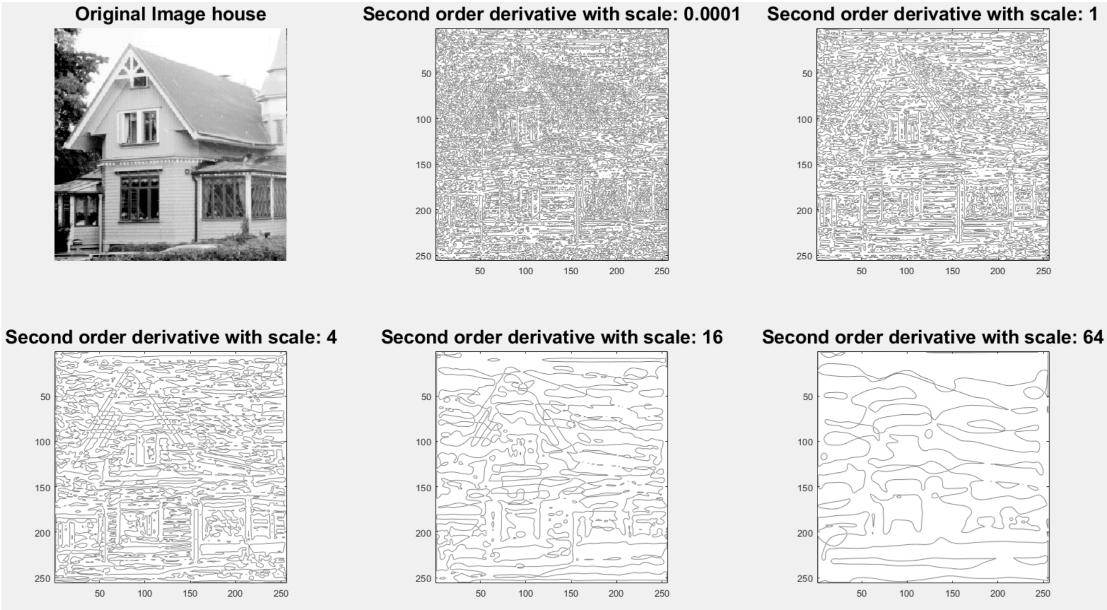

Figure 4- Image house and its second order derivative observations with different scales

From our experiment with application of second derivative on different scales, we see that as we have a coarser observation that results with less details because of less zero-crossing. This actually in a way helps us the recognize the main object in the image because it removes noise as we filter out the higher frequencies.

But after a level of scaling, we also start to lose the main structures because zero-crossings belonging to main object start to disappear or become distorted in their directions. Also we can notice than the shape changes dynamically as we change scaling. This results with removal of core structures of the image which makes it harder to recognize the object in the image.

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

Answers:

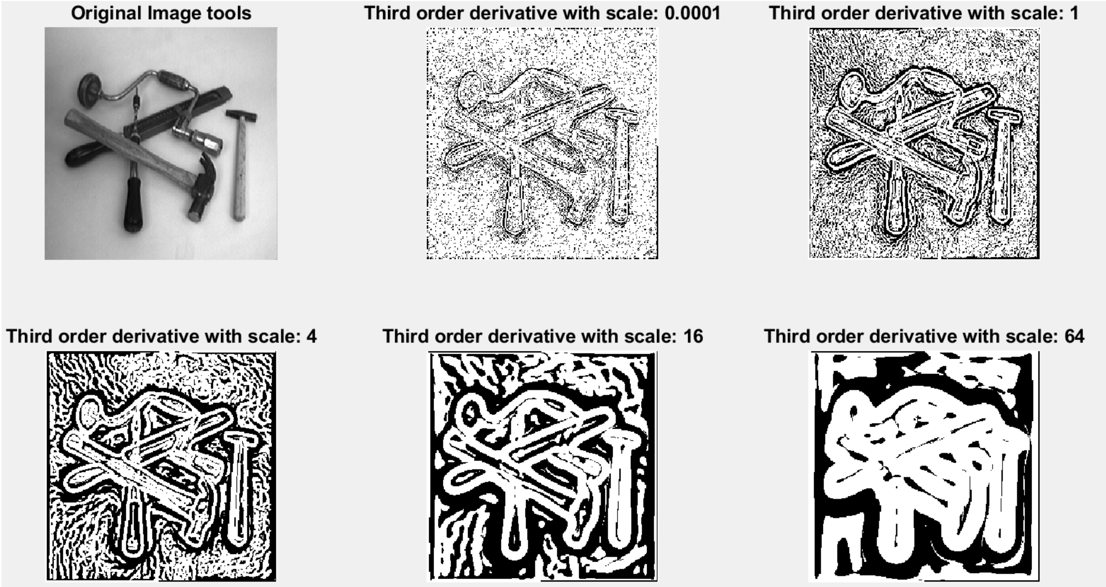

Figure 4- Image tools and its third order derivative observations with different scales

When we observe the third order derivative of tools image with different scales, we see that in that case white areas correspond to the edges and pixels which satisfies Lvvv < 0. For lower values of scaling, the density of these edges is not high. As we increase our scale, we see that the edges become denser and thicker and the detailed edgy pixels in the background start to unite and produce white areas.

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

Answers:

When both derivatives are checked individually, they both contain information related to edges. Lvv provides us the possible thin edges that we are aiming for. But the quality of them individually is not satisfying for a proper edge detection.

In order to achieve the best possible edge detection, we need to get the local maxima for gradient magnitude. And for this, we need to combine the cases where Lvv = 0 and Lvvv < 0 which will provide us both zero-crossings and sign variations.

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

Answers:

After experimenting with different scales and thresholds, the best combinations we have considered for these images are as in the following:

Image	**Threshold**	**Scale**
godthem256	**4**	**4**
few256	**10**	**4**

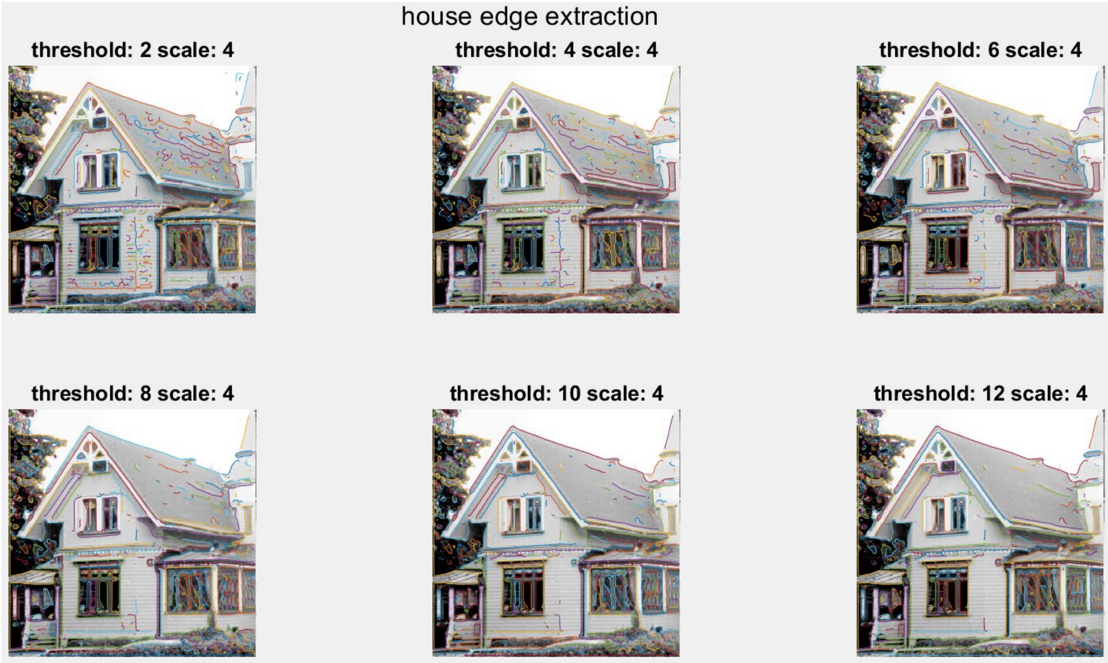

Figure 7.1- Edge extraction of house image for scale = 4 and different thresholds

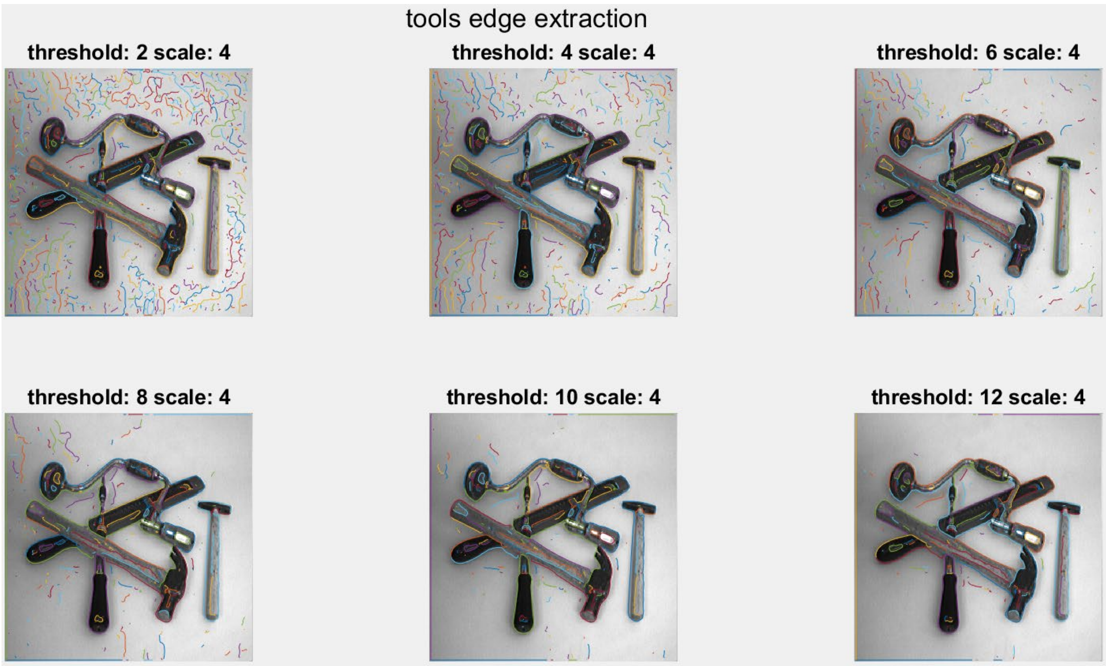

Figure 7.2- Edge extraction of tools image for scale = 4 and different thresholds

We can see that the reference given in the lab manual is achieved as in the following zoom in:

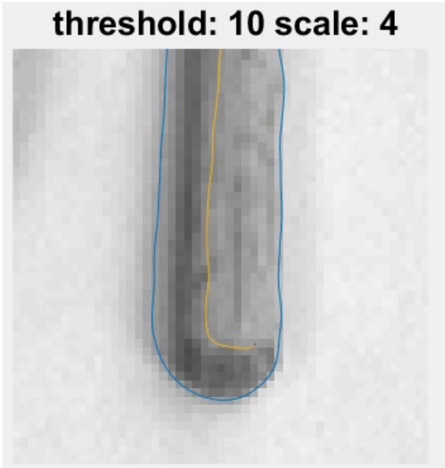

Figure 7.3- Closer observation of tools edge extraction for reference similarity check

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:

In this exercise, hough transforms and line segments are calculated with following parameters:

Threshold	10
Scale	4
Nrho	512
Ntheta	512
Nlines	10

The peak and line segment relationship can be observed by following examples:

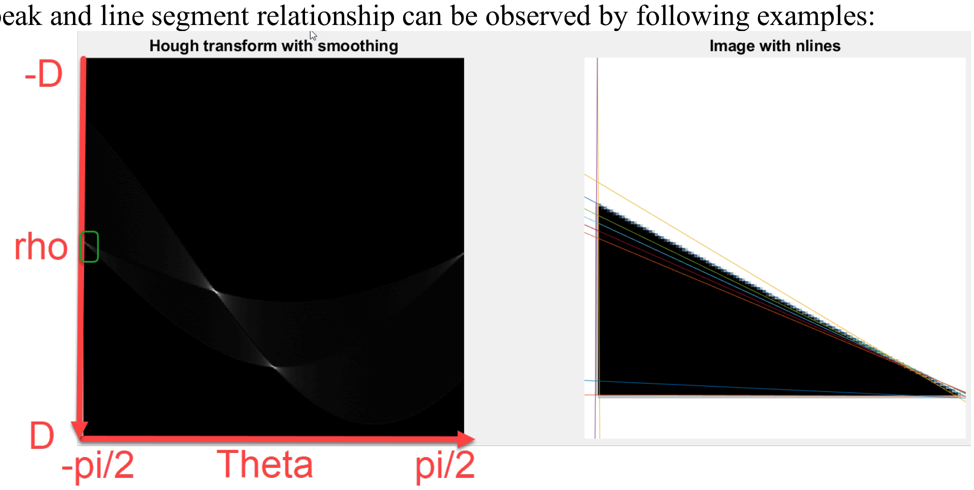

Figure 8.1-Hough transform and hough line added triangle128

As we can see, corresponding edge angle(theta) values has a dense white color region in the Hough transform since we have a horizontal line in image which corresponds to 0 degree at the middle. Also, we have a vertical and sloped line with 135 degree (-45 degree) which are visualized as on the middle and left middle region. Other examples are as follows:

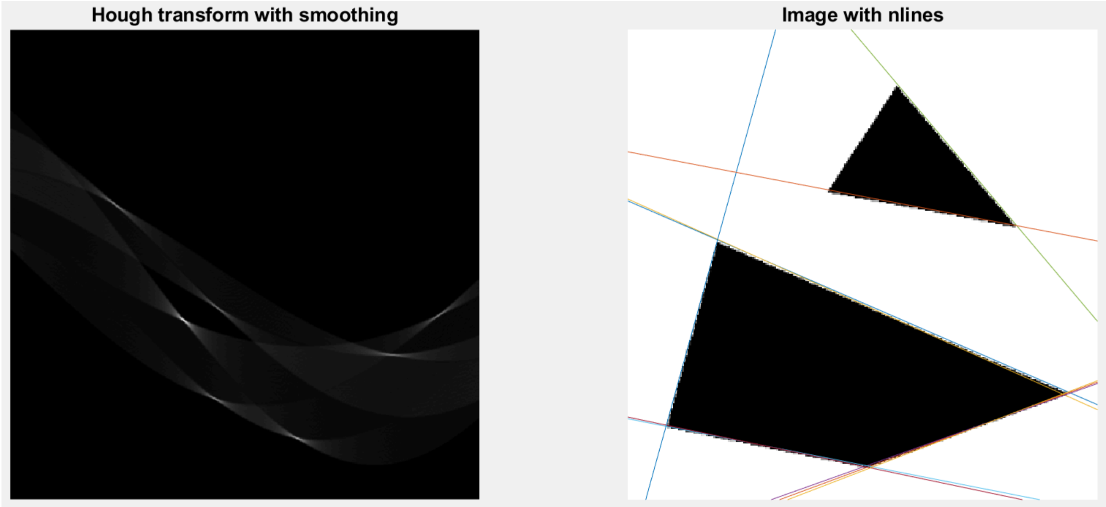

Figure 8.2-Hough transform and hough line added houghtest256

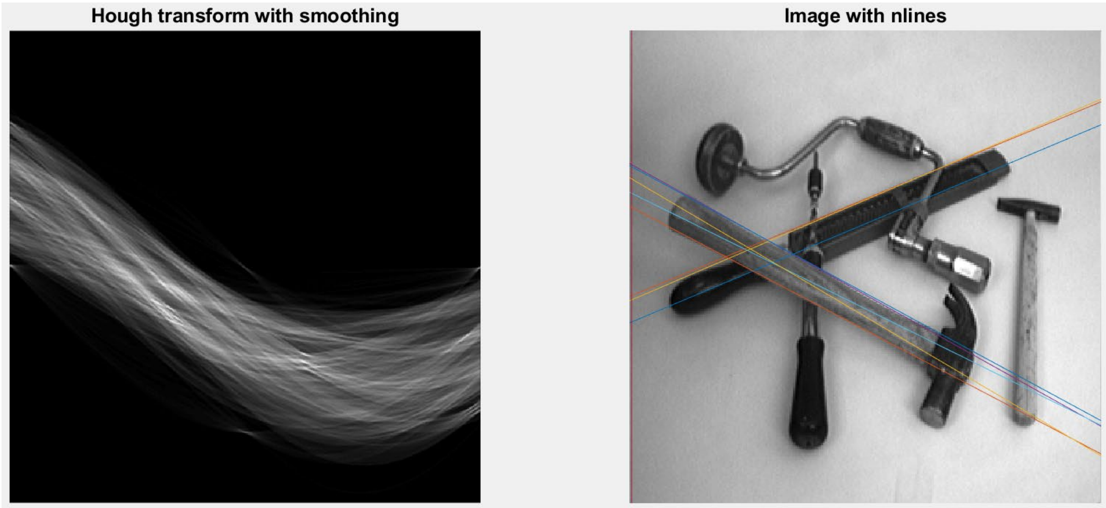

Figure 8.3-Hough transform and hough line added few256

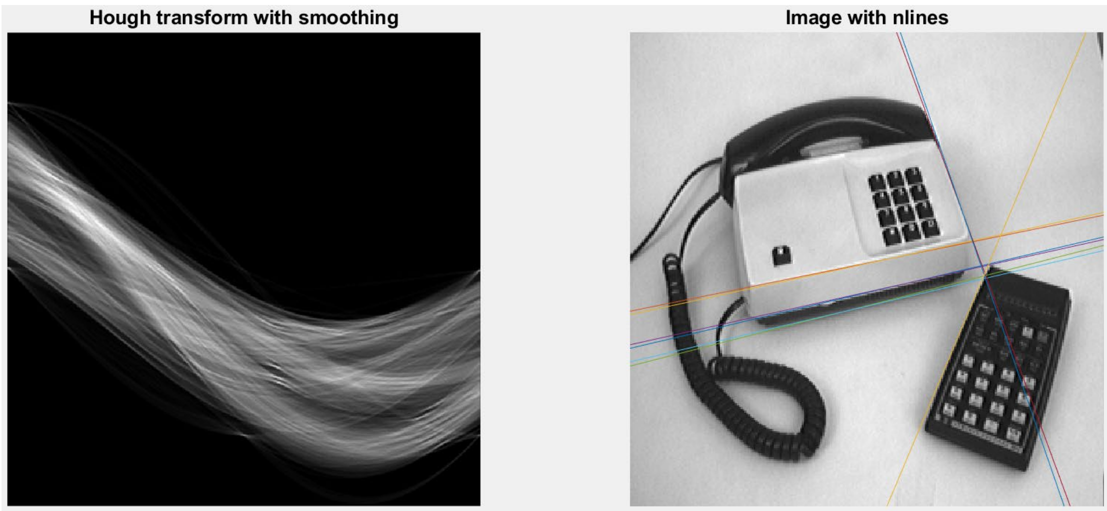

Figure 8.4-Hough transform and hough line added phonecalc256

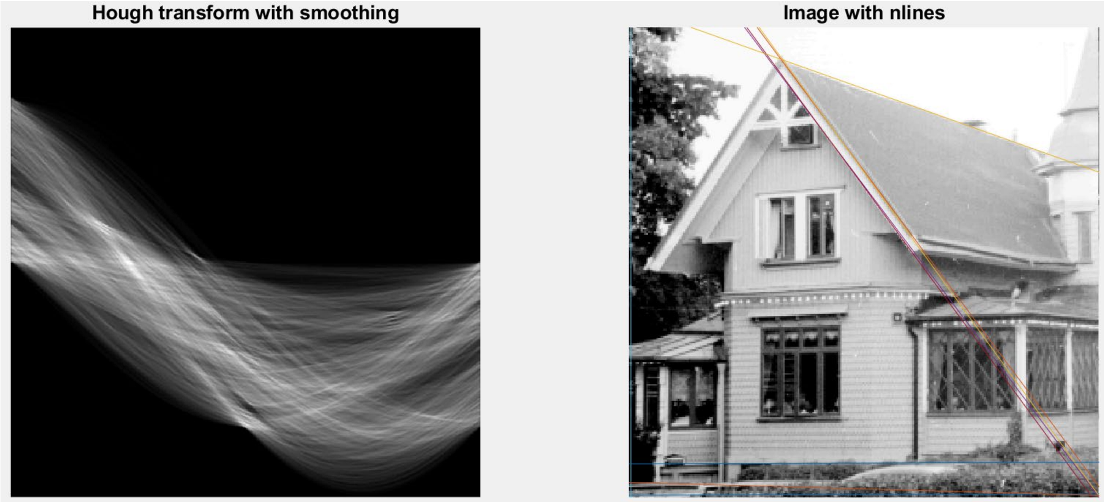

Figure 8.5-Hough transform and hough line added godthem256

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

Answers:
When we are checking the computational time, we have checked it according to the number of rho and theta. During this exercise, 4 cases have been calculated in terms of time and their duration can be seen as in the following:

```
Duration for small nrho and small ntheta
Warning: The error argument from CONTOURS
Elapsed time is 0.715389 seconds.
Duration for large nrho and small ntheta
Warning: The error argument from CONTOURS
Elapsed time is 1.386003 seconds.
Duration for small nrho and large ntheta
Warning: The error argument from CONTOURS
Elapsed time is 6.235860 seconds.
Duration for large nrho and large ntheta
Warning: The error argument from CONTOURS
Elapsed time is 13.337721 seconds.
```

Figure 9-Durations of cases with different values of nrho and ntheta

Case Number	**nrho**	**ntheta**
Case 1	100	100
Case 2	1000	100
Case 3	100	1000
Case 4	1000	1000

That can be easily seen that as we increase the number of cells with either nrho or ntheta, the duration of the process increases. But the impact of the increase of ntheta is much higher on the total duration of the process. The reason is that since we are checking all the angles on the edges, the increase in the angle resolution will increase the loop duration and it will affect much more.

On the other hand, we cannot say that nrho has no importance because as we have lowe nrho, we have less resolution for distance which will not be able to provide the varying length edges.

In terms of results, lower number of cells really affect the accuracy of the edge lines and there should be a sufficient number of balanced cells in order to achieve the accuracy for these lines.

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:

We have tried to implement different functions of gradient magnitude for corresponding point and different attempts have been implemented with different functions as in the following:

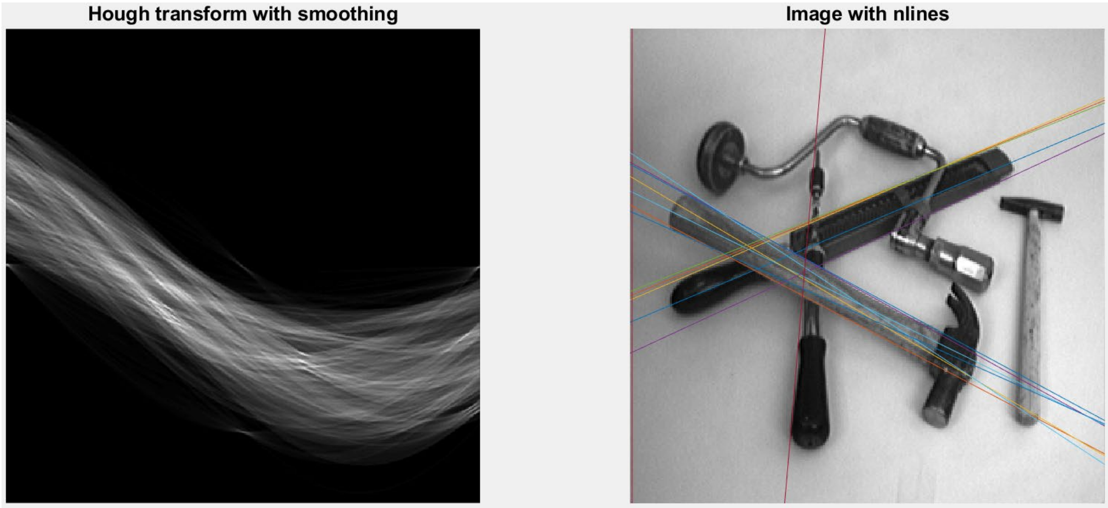

Figure 10.1-Accumulator incremented by 1

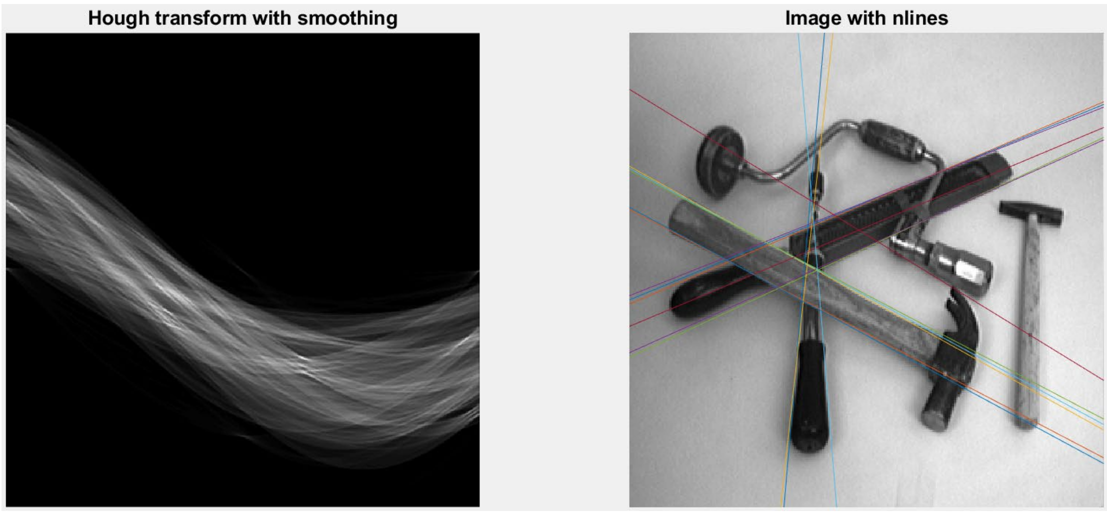

Figure 10.2-Accumulator incremented by log(gradient magnitude)

We see that the logarithmic dependence on gradient magnitude is similar to the incrementation by 1 since logarithmic function decreases the weight effect of the gradient magnitude.

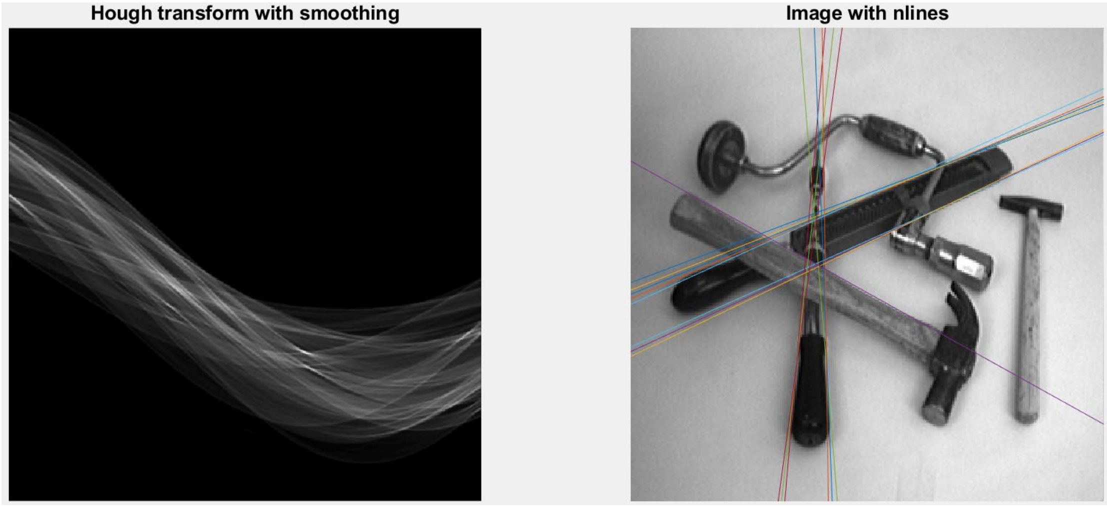

Figure 10.3-Accumulator incremented by gradient magnitude

From that figure, we can realize that incrementation of vote by gradient magnitude started to weight the dominant edges which also started some repetition of edge lines on particular edges. That also affects the thickness of the hough transform on the left which is thinner compared to previous implementations.

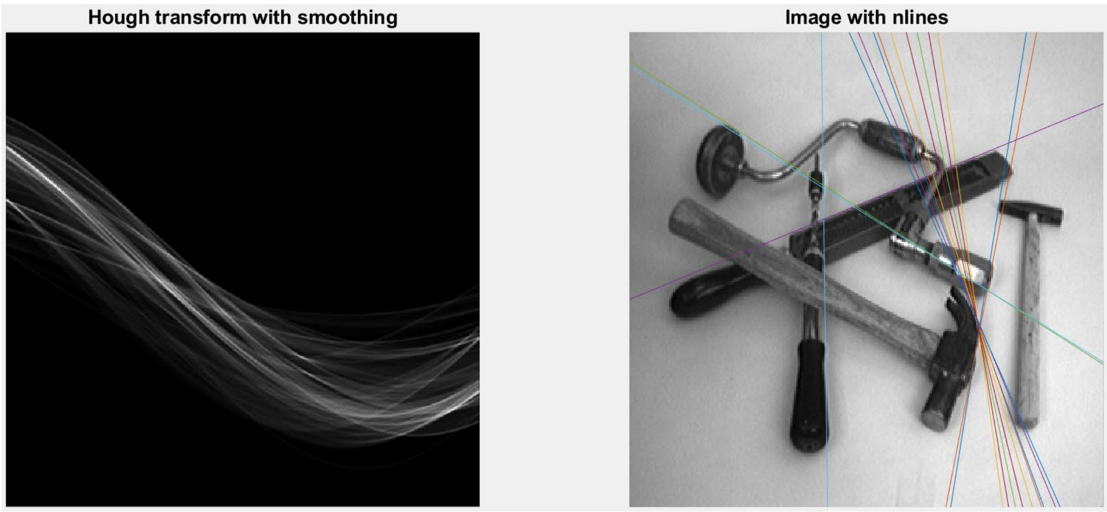

Figure 10.4-Accumulator incremented by cube of gradient magnitude

In our last figure, we analyze the incrementation with the cube of gradient magnitude and it is easy to see that it started to stuck at a point and repeat edge lines over that point which leads hough transform to become even thinner.

From these different approaches of functional incrementations based on gradient magnitude, we may like to prefer a logarithmic or linear dependence on gradient magnitude if we want to weight stronger edges more. Otherwise, cubic incrementation results with so much dominancy which may not be an improving implementation.