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

Name: _______Yihan Wang____________ Program: _______TINNM________

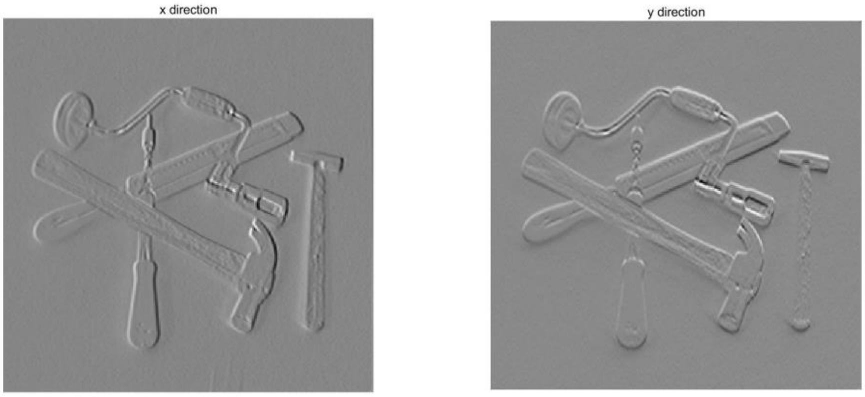
Instructions: Complete the lab according to the instructions in the notes and respond to the questions stated below. Keep the answers short and focus on what is essential. Illustrate with figures only when explicitly requested.

Good luck!

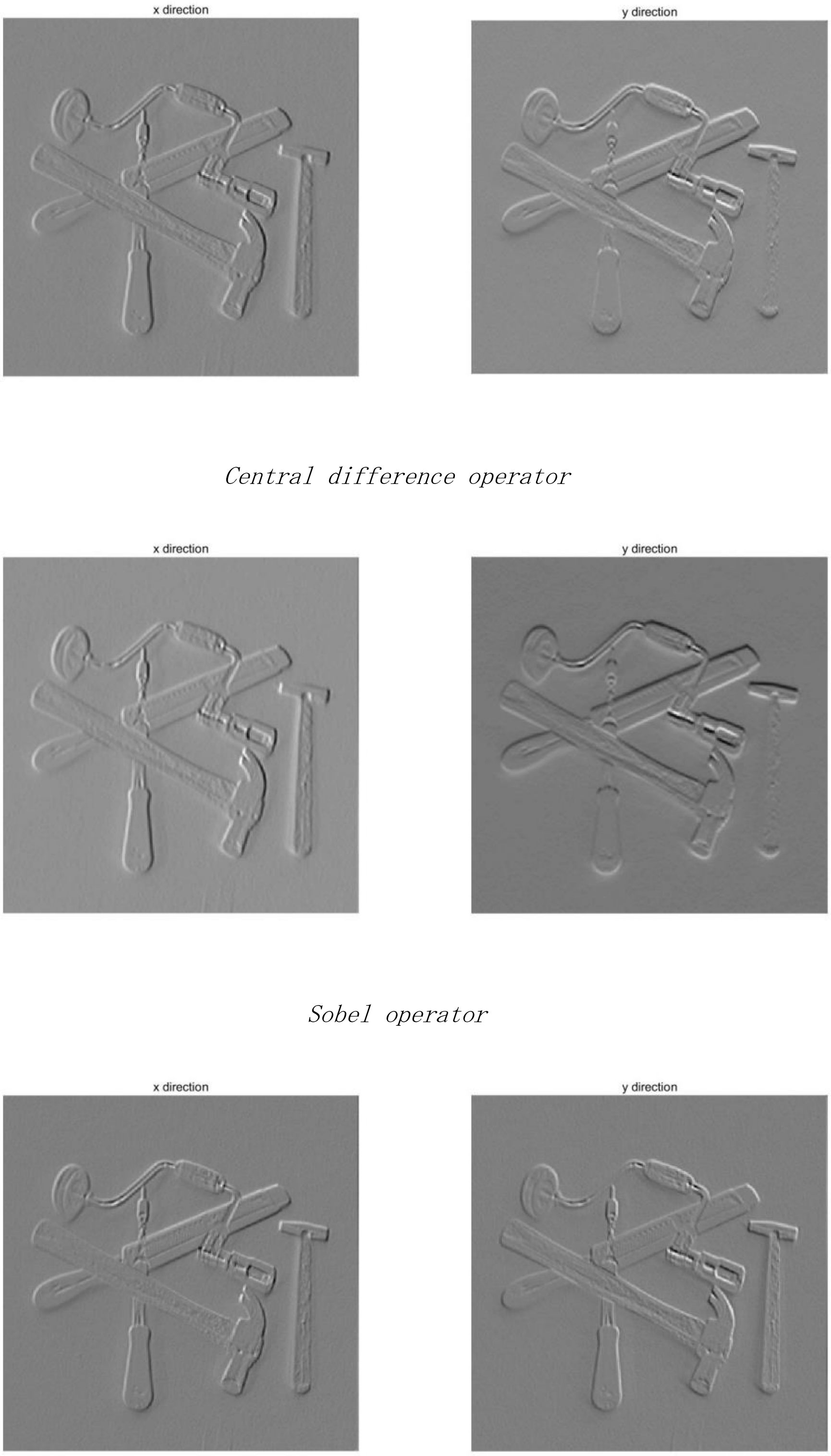
Question 1: What do you expect the results to look like and why? Compare the size of *dxtools* with the size of *tools*. Why are these sizes different?

Answers:
Expected results: dxtools will emphasize fluctuations in derivation on x-axis, while dytools will show more clear difference on y-axis. And at edges, there are positive and negative ridges. Here I tried several different differential operators, the results are shown in Figure 1. From these results, some difference can be observed:

1. results for dxtools are relatively similar for the first three operators, and sobel operator looks better in y direction (edges are thicker).
2. Robert's operator has thinner edges in both x and y direction.



Simple difference operator



Central difference operator

Sobel operator

Robert's diagonal operator

Figure1 : differential results with varies differential operator

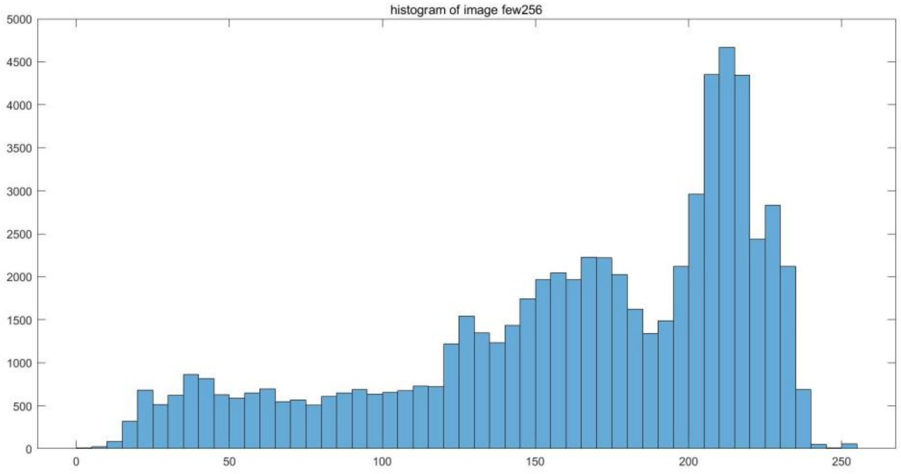
The size of tools and dtools are shown in Table 1, from which we can see the size of dtools are smaller then tools. This is because when conv function convolving the tools with operators, it cannot approximation derivation of edges.

Table1: size of tools and dtools

	x-direction		y-direction	
	row	column	row	column
tools	256	256	256	256
simple	256	254	254	256
central	256	254	254	256
sobel	254	254	254	254
robert	255	255	255	255

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

Answers:
Here we use histogram of input pictures to find a suitable threshold. The histograms are shown in Figure 2. However, there is no specific pattern of histogram for threshold identification and edges are not consistent. If threshold is too low, sharp edges become wide and local maxima can be influences by noise. If threshold is too high, mild edges will be broken or fake. We can find some values have possibility to be a threshold, but the effect is hard to predict. Thus, we choose some possible values and observe whether it is reasonable or not. Results with different threshold are shown in Figure 3, and choose threshold=100 for figure few256 and threshold=185 for figure godthem256 as the best value.



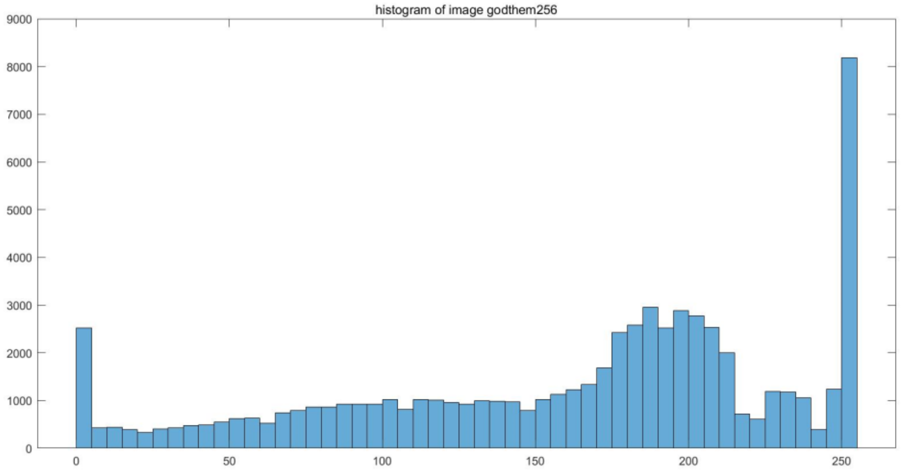


Figure 2: histogram of figures

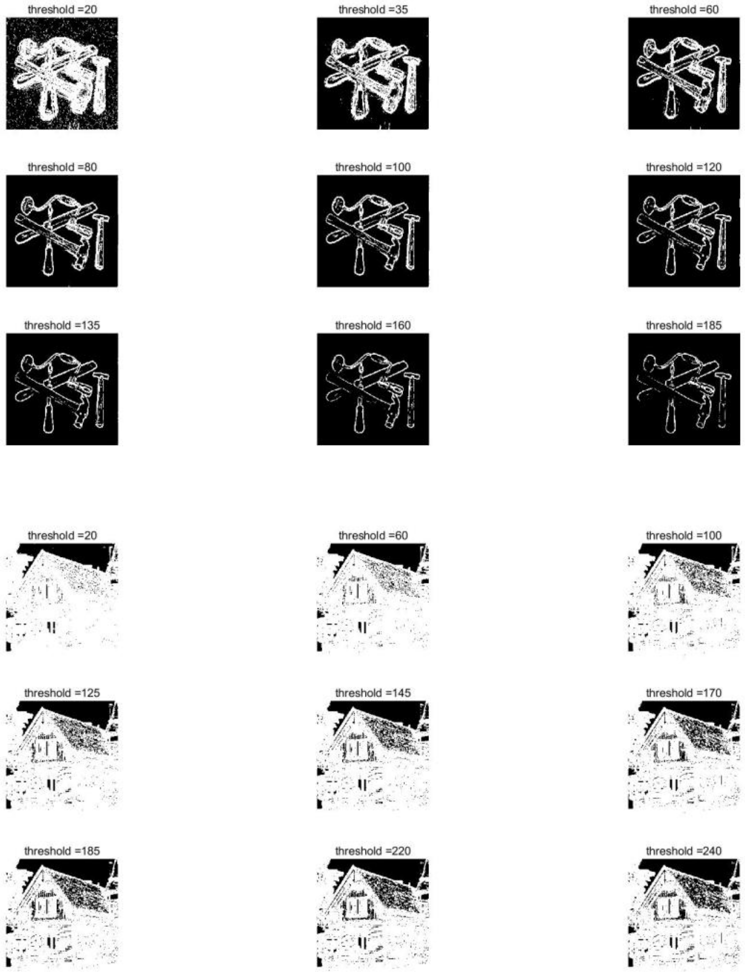
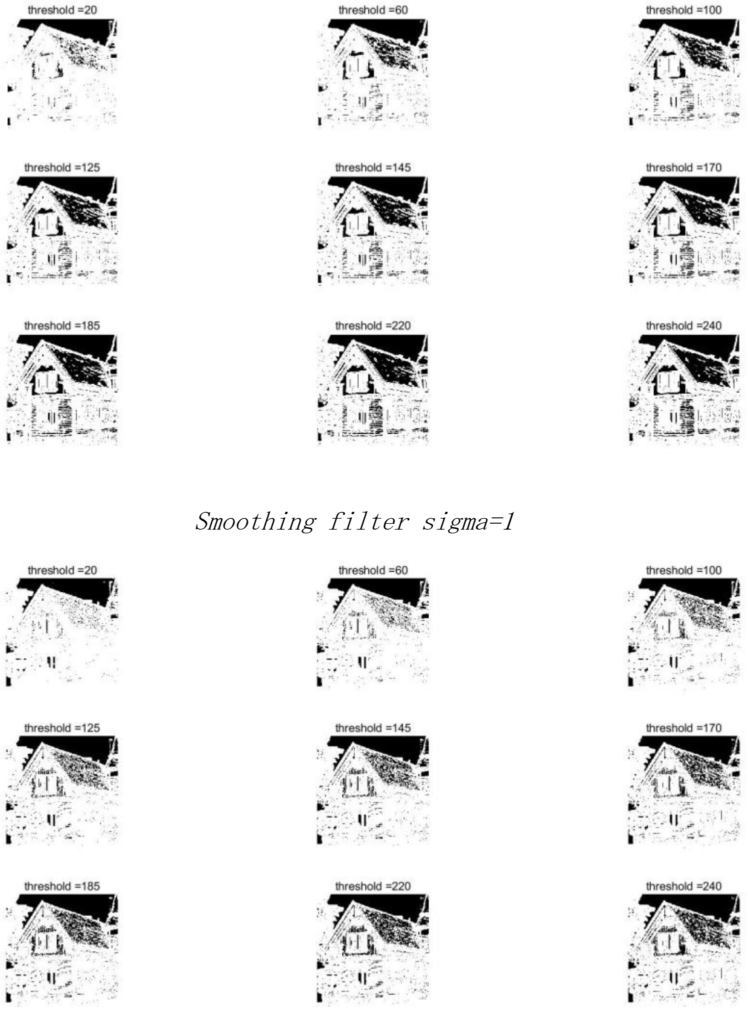


Figure 3: various threshold for figures

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

Answers:
Smoothing help to find edge in some way because it can remove part of high frequency noise which can make edge line more clearly. However, at the same time, some edge information may be discarded and reasonable threshold will be more difficult to find (lower than before). The smoothing results are shown in Figure 4.



Smoothing filter sigma=1

Smoothing filter sigma=5

Figure 4: find edge after smoothing with different sigma of blurring filter

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

Answers:
The results are shown in Figure 5. From the results we can find:

1. Smaller scale will make images have more curves, and one edge may have a lot corresponding curves which makes the picture look more messy. This is because a lot zero-crossing has occurs, which means the second derivation has a lot zero point.
2. Suitable scale will make images more clear and identifiable. This is because smoothing can reduce number of second derivation's zero points.
3. Too large scale will make curves too little and some may be distorted. (smoothing too much)

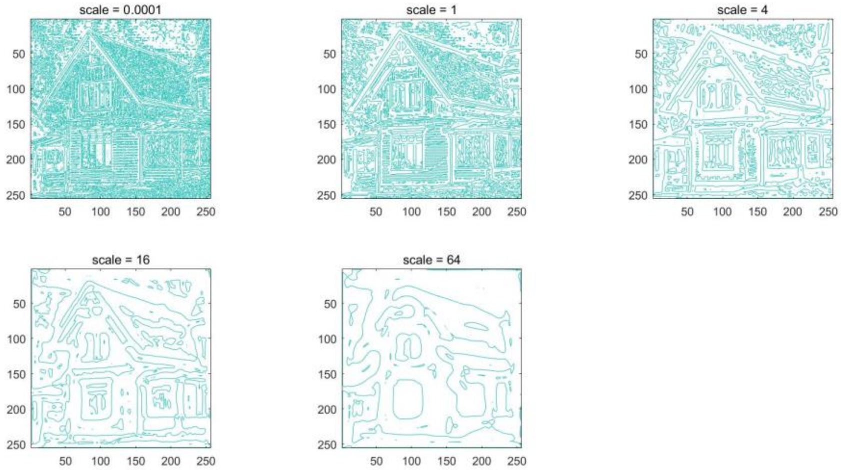


Figure 5: edges with various value of scale

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 results are shown in Figure 6. Observe the results, we have:

1. The contours becomes more recognizable then second derivation, but it seems more messy then before.
2. The white areas are edges and pixels whose Lvvv<0.
3. When scale getting larger, the edges become more and more thick. If the scale is too large, it will be hard to recognize edges.

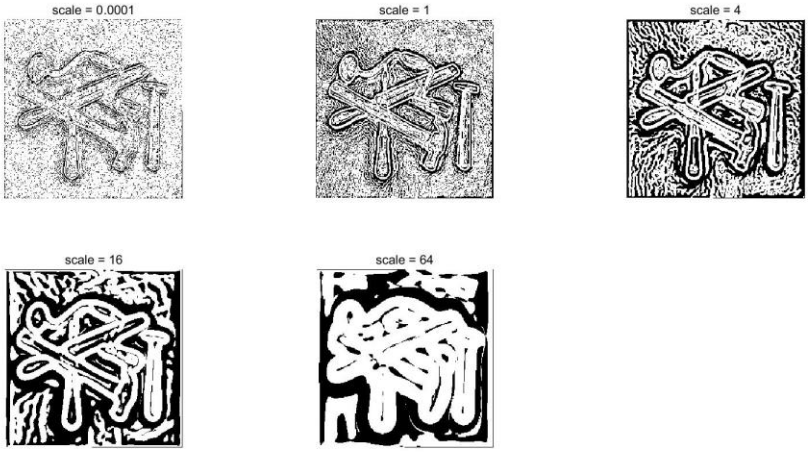
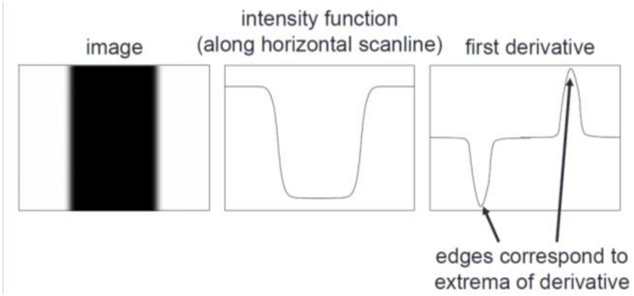


Figure 6: edges of few256 with different scale

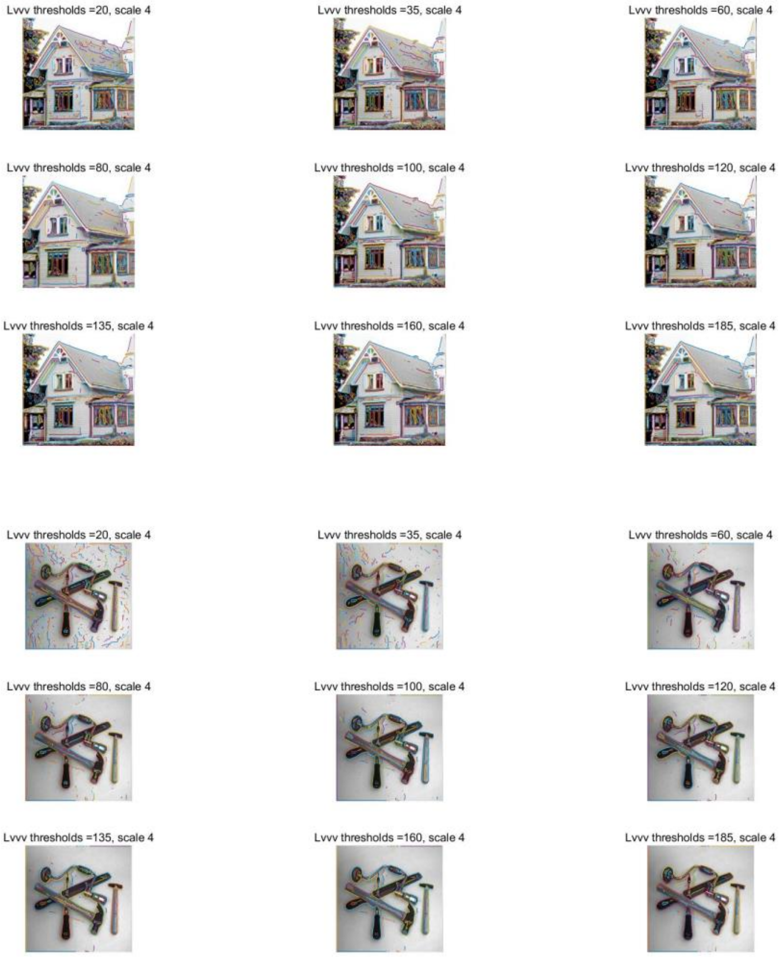
Question 6: How can you use the response from *Lvv* to detect edges, and how can you improve the result by using *Lvvv*?

Answers:
Because these two methods have their own advantages and disadvantages in edge detection, we should use them together, that is to say, we choose the pixel points that meet constraints ①Lvv=0 and ②Lvvv<0 at the same time as the points that make up the edge. However, we also need to pay attention to the moderation of parameter adjustment, otherwise it will backfire.

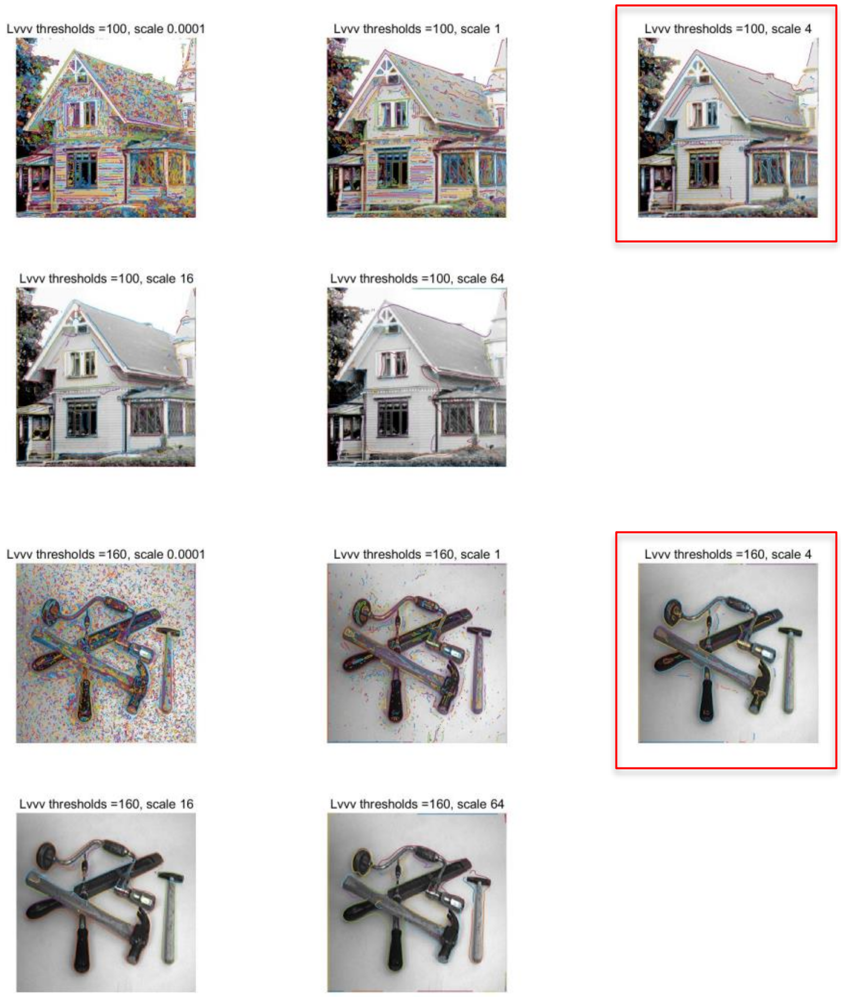


Question 7: Present your best results obtained with *extractedge* for *house* and *tools*.

Answers:
Few256: threshold=160,scale=4
godthem256: threshold=100,scale=4


Lvvv thresholds =20, scale 4
Lvvv thresholds =35, scale 4
Lvvv thresholds =60, scale 4
Lvvv thresholds =80, scale 4
Lvvv thresholds =100, scale 4
Lvvv thresholds =120, scale 4
Lvvv thresholds =135, scale 4
Lvvv thresholds =160, scale 4
Lvvv thresholds =185, scale 4
Lvvv thresholds =20, scale 4
Lvvv thresholds =35, scale 4
Lvvv thresholds =60, scale 4
Lvvv thresholds =80, scale 4
Lvvv thresholds =100, scale 4
Lvvv thresholds =120, scale 4
Lvvv thresholds =135, scale 4
Lvvv thresholds =160, scale 4
Lvvv thresholds =185, scale 4

Different threshold with fixed scale

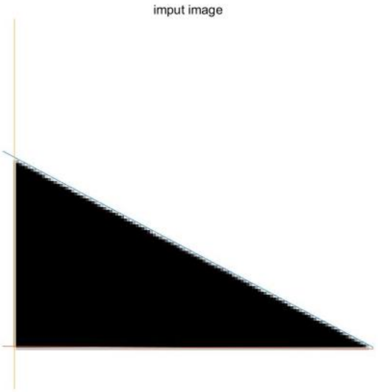

Lvvv thresholds =100, scale 0.0001
Lvvv thresholds =100, scale 1
Lvvv thresholds =100, scale 4
Lvvv thresholds =100, scale 16
Lvvv thresholds =100, scale 64
Lvvv thresholds =160, scale 0.0001
Lvvv thresholds =160, scale 1
Lvvv thresholds =160, scale 4
Lvvv thresholds =160, scale 16
Lvvv thresholds =160, scale 64

Different scale with fixed threshold

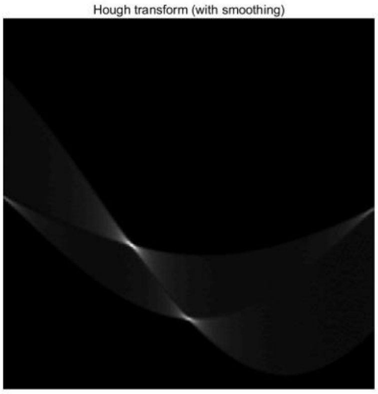
Figure 7: results for different thresholds and scales

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.

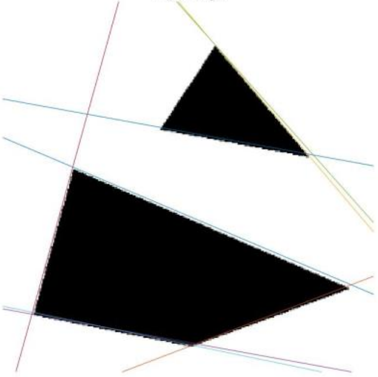
Answers:
The results are shown in Figure 8. As we can see, the implementation is correct.



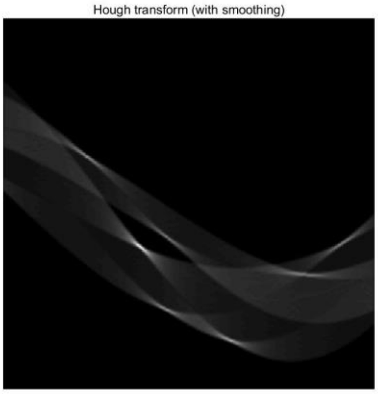
imput image



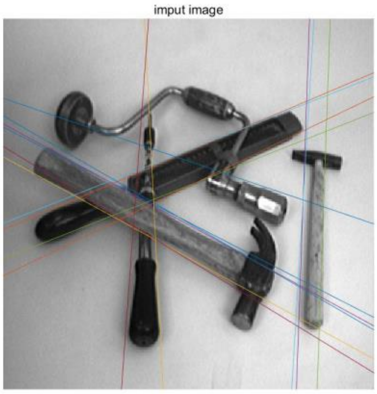
Hough transform (with smoothing)



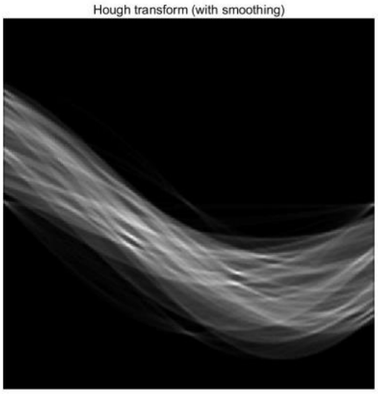
imput image



Hough transform (with smoothing)



imput image



Hough transform (with smoothing)

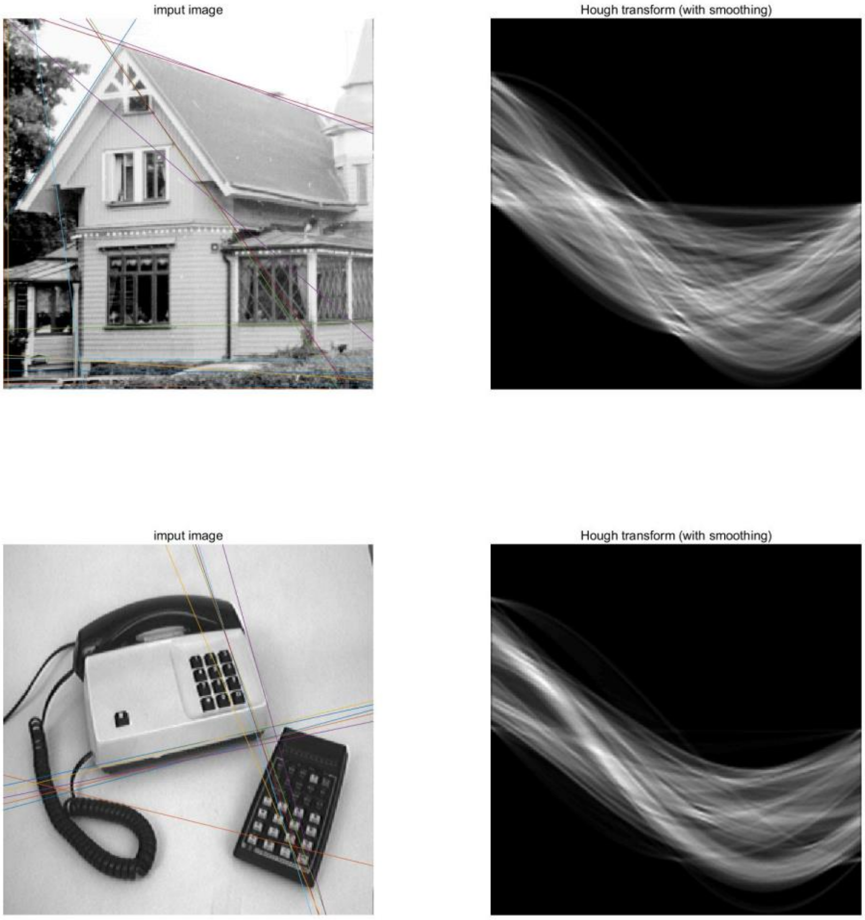


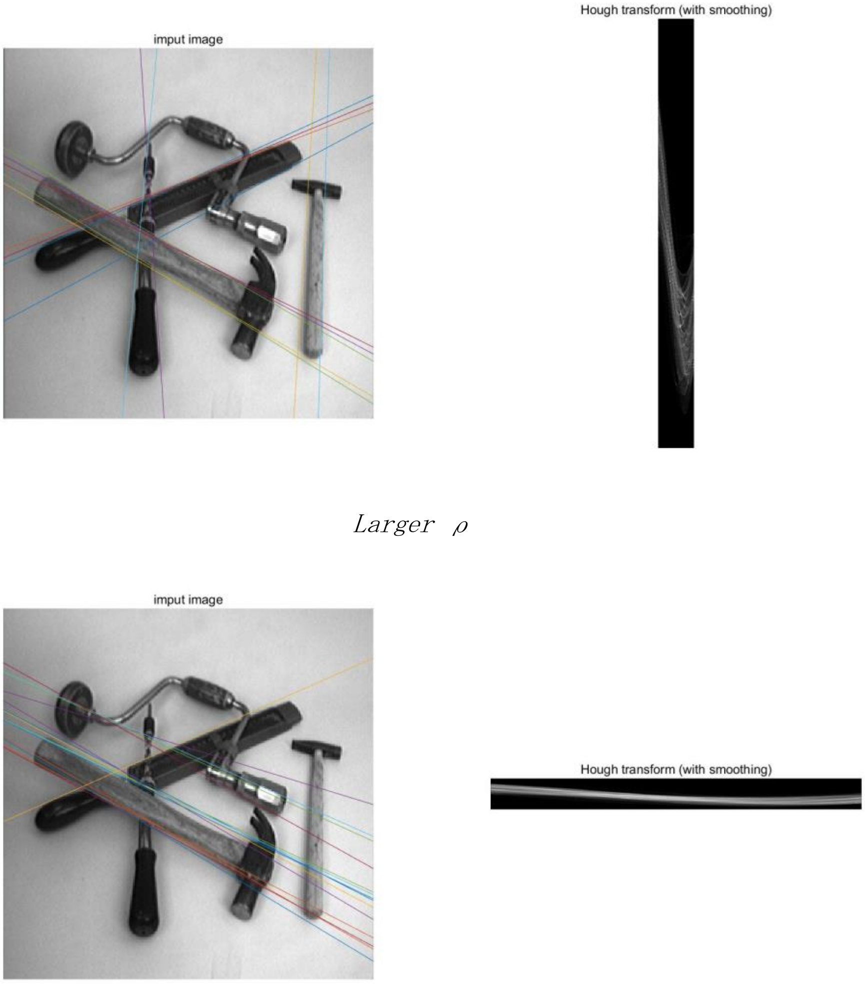
Figure 8: images with lines (left) and Hough transform after smoothing (right)

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

Answers:

The figure results are shown in Figure 9 and time results are shown in Table 2. We find:

1. The computational time increases along with increasement of number of ρ and θ because more computation need to be done.
2. Increase of θ has bigger affect than ρ, this is because for each point we need to compute lines for each different θ.
3. Increase of θ makes more lines gathered at sharper edges and local maxima is more near edge lines.


imput image
Hough transform (with smoothing)
imput image
Hough transform (with smoothing)

Larger ρ

Larger θ

Figure 9: results after changing parameters

Table 2: computational time

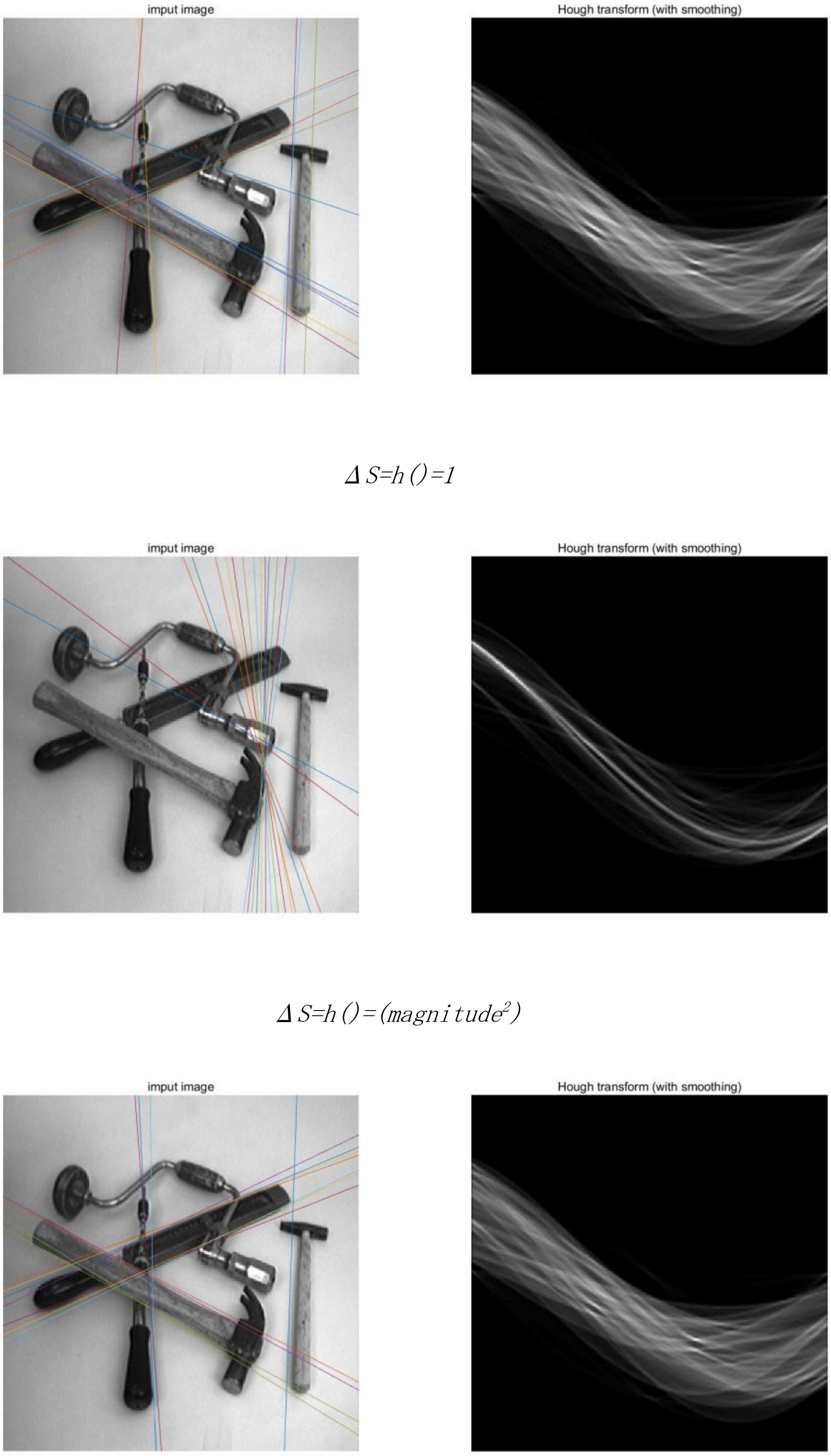
ρ	θ	computational time
100	100	4304.7ms
1200	100	4304.9ms
100	1200	4305.9ms

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:
Here use three different h function.

1. ΔS=h()=1
2. ΔS=h()=(magnitude^2)
3. ΔS=h()=log(magnitude)

The core idea for this problem is to change voting weight. Since log function will reduce the difference between high and low values, this will make edges with lower magnitude have larger possibility to be detected. However, as for the squared h function, shaper edges will have too much weight, which may cause too many lines gathered around sharper edges. The results are shown in Figure 10.



$\Delta S=h()=1$

$\Delta S=h()=(magnitude^2)$

$\Delta S=h()=log(magnitude)$

Figure 10: different accumulator incrementation function