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

Name: Kepan Gao   Program: Transport and Geoinformation Technology

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 case, the Sobel Operators are used:

$$Kernel_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad Kernel_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

The difference operators are shown as below (Figure 1).

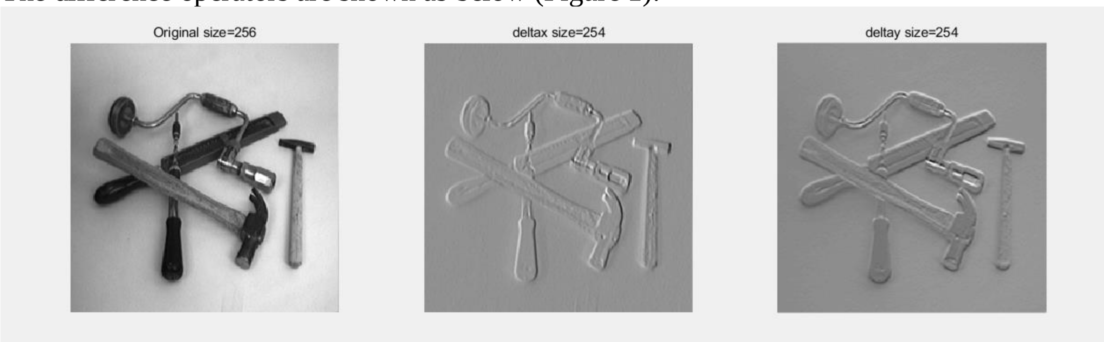


Figure 1. Difference Operators of Tools

When Δx is used, the vertical edges will be highlighted. When Δy is used, the horizontal edges will be highlighted. Explanations are as follows:
When Δx is used, vertical edges can be well detected:

$$Origin = \begin{bmatrix} 0 & \cdots & 1 & \cdots & 0 \\ 0 & \cdots & 1 & \cdots & 0 \\ 0 & \cdots & 1 & \cdots & 0 \\ 0 & \cdots & 1 & \cdots & 0 \\ 0 & \cdots & 1 & \cdots & 0 \end{bmatrix} \quad Edge = \begin{bmatrix} 0 & \cdots & 3 & 0 & -3 & \cdots & 0 \\ 0 & \cdots & 4 & 0 & -4 & \cdots & 0 \\ 0 & \cdots & 4 & 0 & -4 & \cdots & 0 \\ 0 & \cdots & 4 & 0 & -4 & \cdots & 0 \\ 0 & \cdots & 3 & 0 & -3 & \cdots & 0 \end{bmatrix}$$

but horizontal edges cannot be well detected:

$$Origin = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 1 & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad Edge = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 0 & 0 & 0 & -1 \\ 2 & 0 & 0 & 0 & -2 \\ 1 & 0 & 0 & 0 & -1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

When Δy is used, vertical edges cannot be well detected:

$$Origin = \begin{bmatrix} 0 & \cdots & 1 & \cdots & 0 \\ 0 & \cdots & 1 & \cdots & 0 \\ 0 & \cdots & 1 & \cdots & 0 \\ 0 & \cdots & 1 & \cdots & 0 \\ 0 & \cdots & 1 & \cdots & 0 \end{bmatrix} \quad Edge = \begin{bmatrix} 0 & \cdots & -1 & -2 & -1 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 0 & 0 & 0 & \cdots & 0 \\ 0 & \cdots & 1 & 2 & 1 & \cdots & 0 \end{bmatrix}$$

but the horizontal edges will be highlighted:

$$Origin = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & 1 & 1 & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad Edge = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ -3 & -4 & -4 & -4 & -3 \\ 0 & 0 & 0 & 0 & 0 \\ 3 & 4 & 4 & 4 & 3 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The size of images after edge detection is 254×254 because the size of convolution kernel is 3×3. Only 254×254 pixels in the middle will have a convolution value.
If we turn the third parameter of function filter2 or conv2 from 'valid' into 'same', 0s will be added around the edges and the size of convoluted matrix will be the same.

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

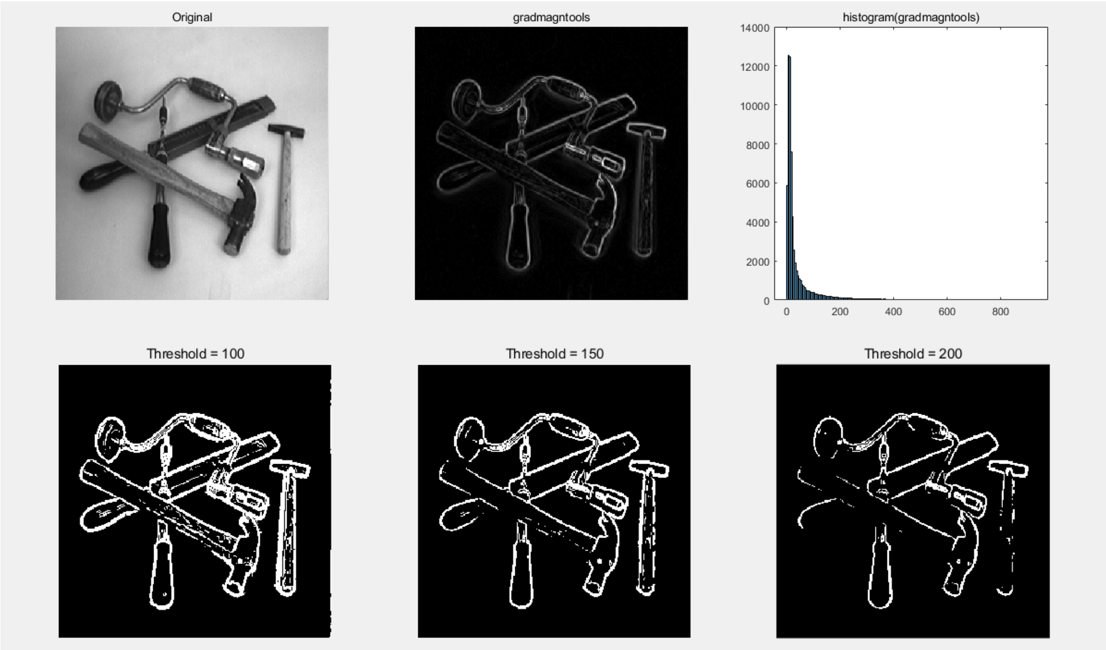


Figure 2. Point-wise Threshold

It is easy to find a threshold that results in thin edges, but it's not easy to get thin and connected edges.
An edge can have different magnitude and thickness in different parts. When threshold is too small, there are some noises and the edges are thick. So, when threshold is increased, those parts with a small magnitude will be removed thoroughly. That is why edges are not connected when threshold is high.

Question 3: Does smoothing the image help to find edges?
Answers:
In this case, $discgaussfft(pic, sigma)$ was used to get smooth figures. This function convolves an image by the separable discrete analogue of the Gaussian kernel by performing the convolution in the Fourier domain. The parameter $sigma$ is the variance of the kernel.
The results are as follows (Figure 3).

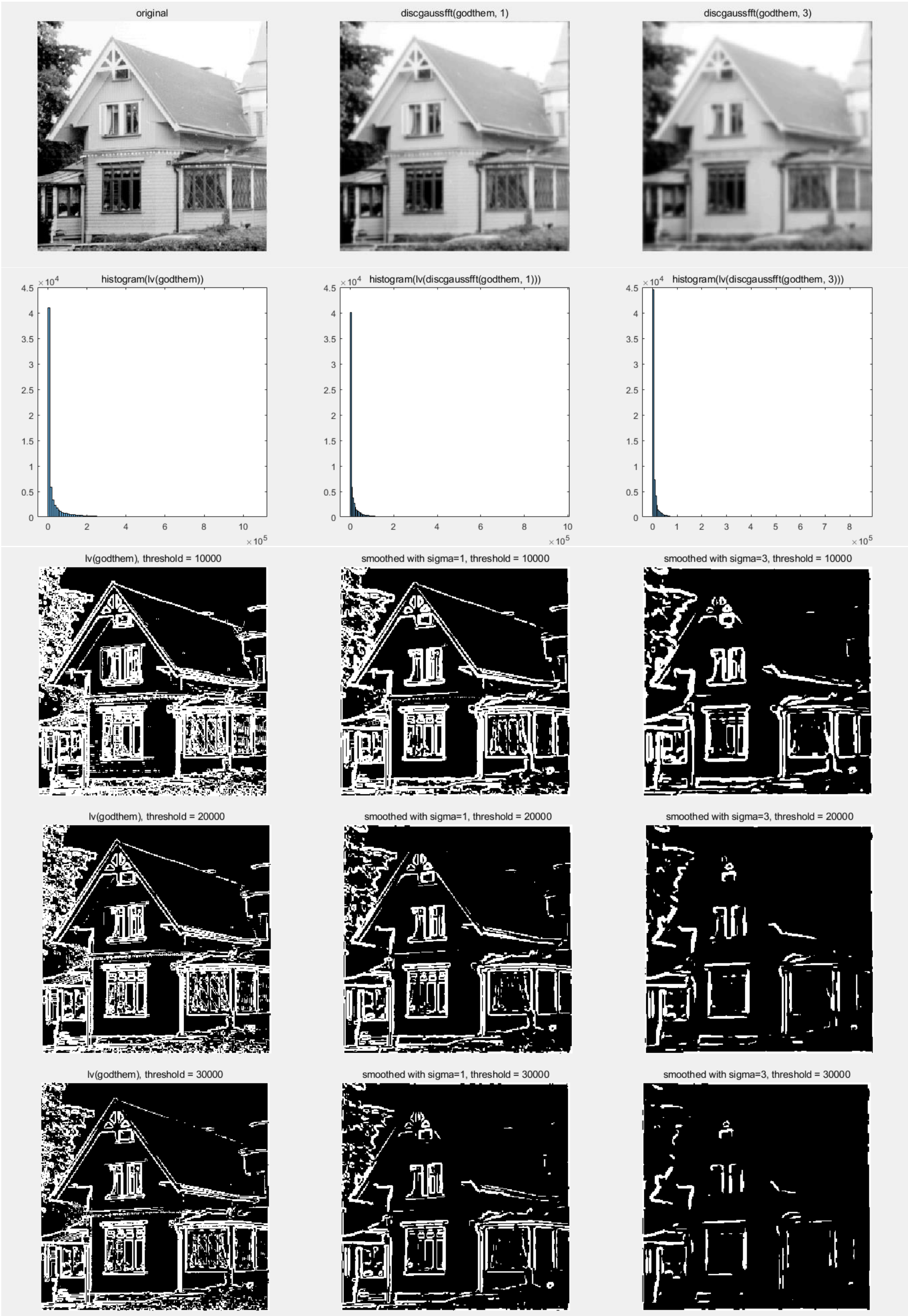


Figure 3. Edge Detection of Smoothed Figure

From the histograms we can see that values of pixels are more concentrated after smoothed. Smoothing does help to get better edges efficiently. Gaussian filter is a low pass filter and it can reduce noises and details with a high frequency. Relatively, the edges are well preserved. When the variance of Gaussian kernel gets larger, the figure will be smoother. The noise can be well suppressed, the gradient magnitude of edges will be suppressed as well. As a result, some edges with a smaller magnitude will be removed with noises. We need to pick a compromised variance to keep as much information as possible while removing noises.

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

Answers:

The edges computed by differential geometry are shown as below (Figure 4).

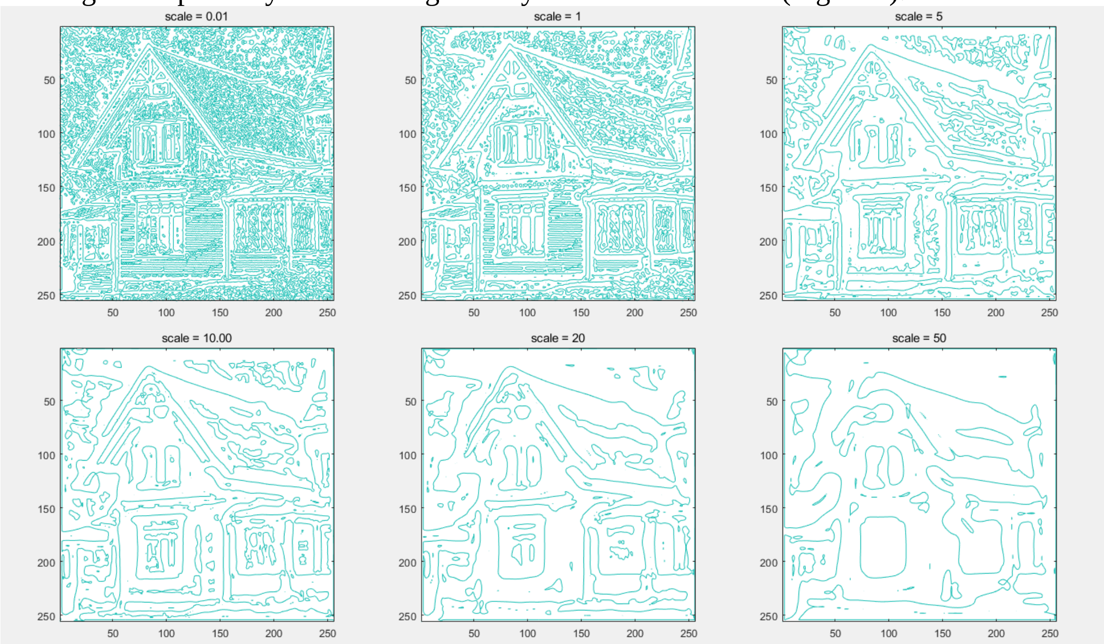


Figure 4. Contour Images with Different Scales of Smoothing

The images are convolved by Gaussian kernels with different scales. Then, the contour image where $\tilde{L}_{vv} = 0$ is drawn.

I observed that when the scale (the variance of Gaussian kernel) is larger, the image is smoother and there will be less noises. When the scale is too large, the image will be badly distorted, and some significant edges will be distorted or gone with noises.

The reason why edges disappear with noises is when the scale (variance of Gaussian kernel) gets larger, the gradient magnitude of whole image will be decreased. The reason of the distortion is Gaussian convolution is a low pass filter, and the details (high frequency components) will be lost.

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 edges computed by differential geometry are shown as below (Figure 5).

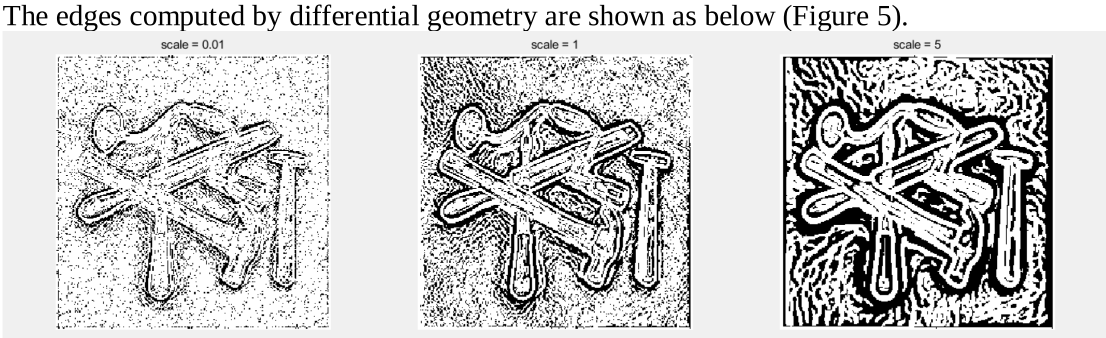


Figure 5. Images with Different Scales of Smoothing

The images are convolved by Gaussian kernels with different scales. Then, the image with a white color for each pixel where $\tilde{L}_{vvv} < 0$ is drawn.
For the pixels in white area,

$$\tilde{L}_{vvv} < 0$$

∴

$$\tilde{L}_{vv} \text{ is decreasing}$$

∴

$$\tilde{L}_{v} \text{ has maximum in white areas}$$

When the scale (the variance of Gaussian kernel) is larger, the connected white areas get larger. Intuitively, the edges get wider.

Question 6: How can you use the response from $\tilde{L}_{vv}$ to detect edges, and how can you improve the result by using $\tilde{L}_{vvv}$?
Answers:
Here I tried several ways to combine two plots together, but none of them works well:

$$\text{and}\big(\text{lvvtilde}(\text{pic}, \text{scale}) == 0, \text{lvvvtilde}(\text{pic}, \text{scale} > 0)\big)$$

$$\text{lvvtilde}(\text{pic}, \text{scale}) == 0 \,.*\, \text{lvvvtilde}(\text{pic}, \text{scale} > 0)$$

After reading the document of function contour, the returned matrix is defined as:

The value of level

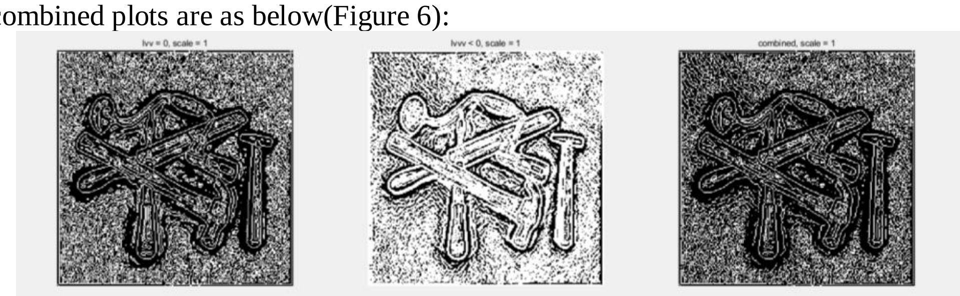
The coordinates in this level

The value of level of another contour line

0	127.1525	127	126	...	0	149.0770	149	148
128	1	1.9998	1.9986	...	6	1	2.0000	1.9996

The number of pixels in this level

The edges with $\tilde{L}_{vv}$ must be estimated due to the returned matrix of function contour consists of coordinates with a double value.
The combined plots are as below(Figure 6):



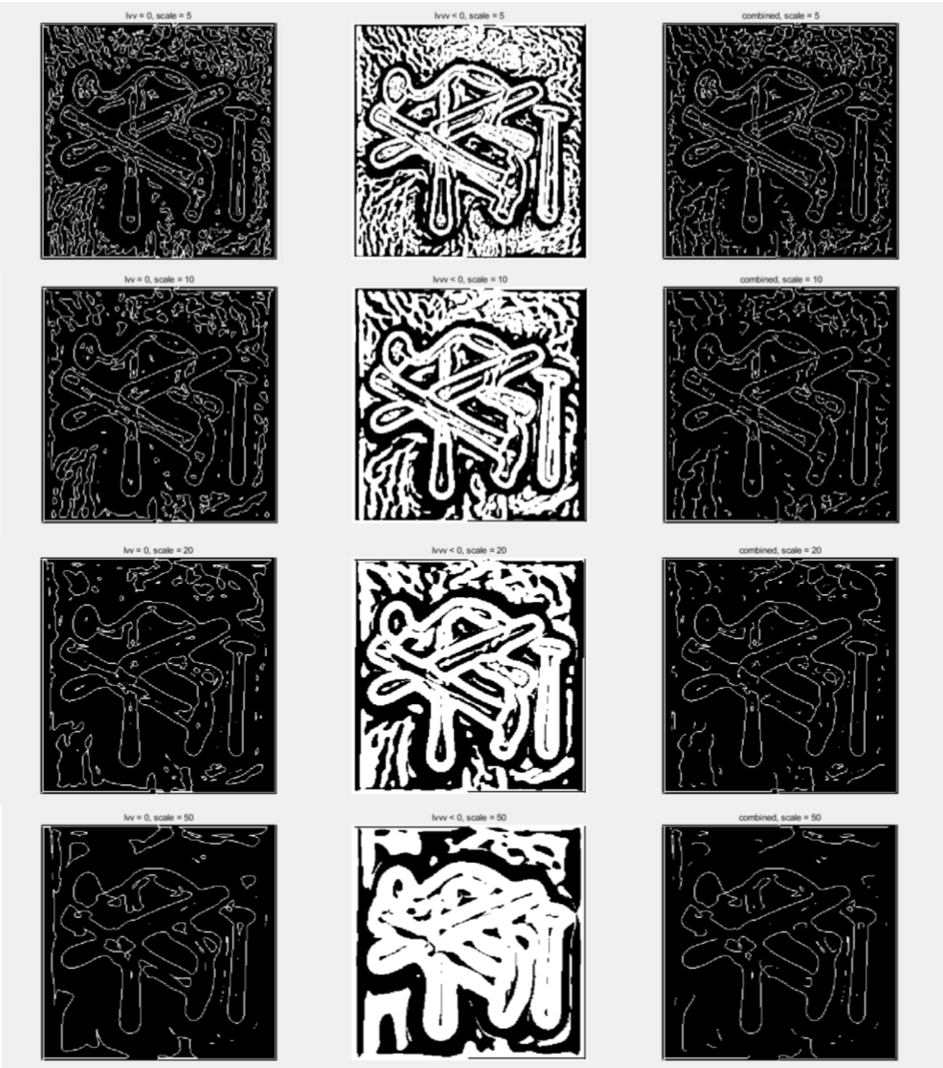


Figure 6. Images with Different Scales of Smoothing

When $\tilde{L}_{vv} = 0$, $\tilde{L}_v$ has an extremum but it could be a maximum or minimum. When combined with the plot where $\tilde{L}_{vvv} < 0$, $\tilde{L}_v$will have only maximum. And that is what we want.

Question 7: Present your best results obtained with *extractedge* for *house* and *tools*.
Answers:
The best results of edges of house and tools are shown as below (Figure 7-2 and Figure 7-2).

Figure 7-1. Best Edges of House

Figure 7-2. Best Edges of Tools

As shown in Figure 7-1 and Figure 7-2. When scale is increased, the gradient magnitude of image will be lower, noises will be suppressed, and the image will be distorted. To get clear edges, the threshold should be decreased despondingly.

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 result of procedure houghedgeline to the simple test image triangle128 is shown as below.

Figure 8. Test to Triangle128

First, calculate edges from the former functions. Then, for each point on the edge, draw a responding curve in Hough space according to:

$$\rho = \sqrt{x^2 + y^2}\cos(\theta - \varphi), \quad where\ \varphi = \arctan\left(\frac{y}{x}\right)$$

Finally, draw lines with parameters (ρ, θ) with a highest accumulative value in the Hough space (brightest spots in the middle of Figure 8).

Application.

The result of houghedgeline applied to the image houghtest256 is as below.

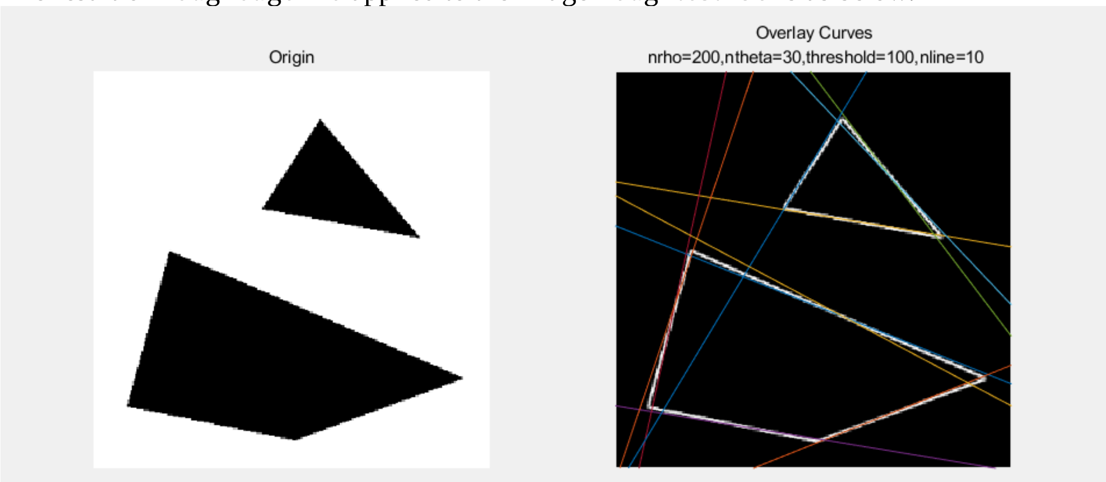
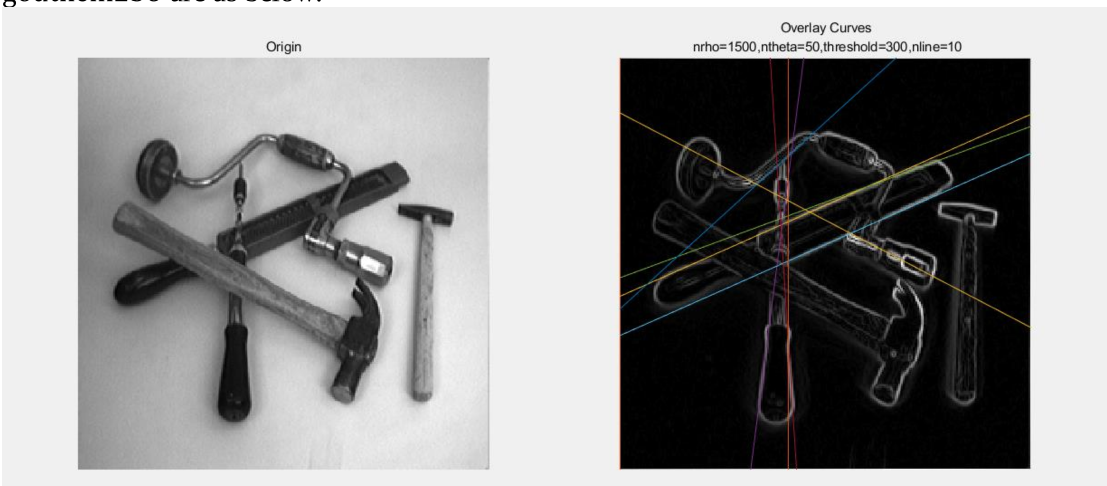


Figure 9-1. Test with houghtest256

The results of houghedgeline applied to the image few256 , phonecalc256 and godthem256 are as below.



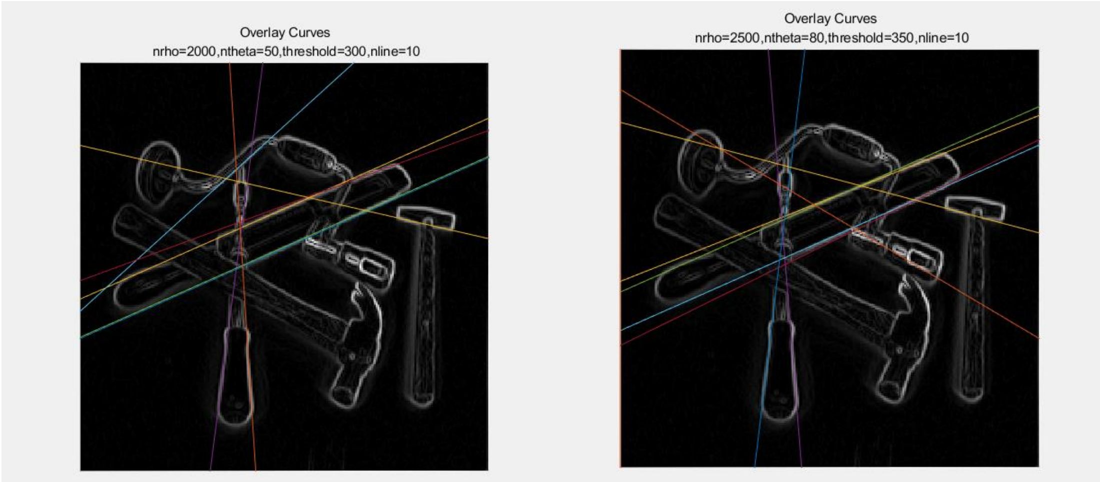


Figure 9-2. Best Results of Few256

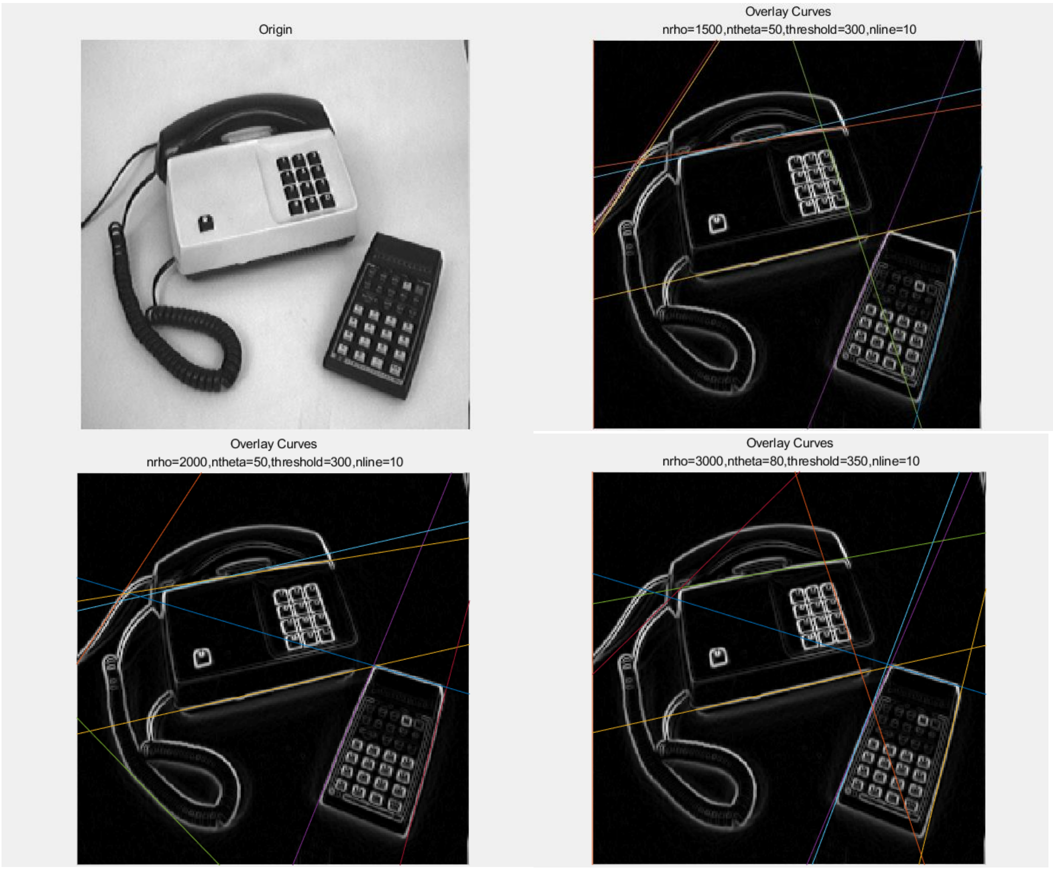
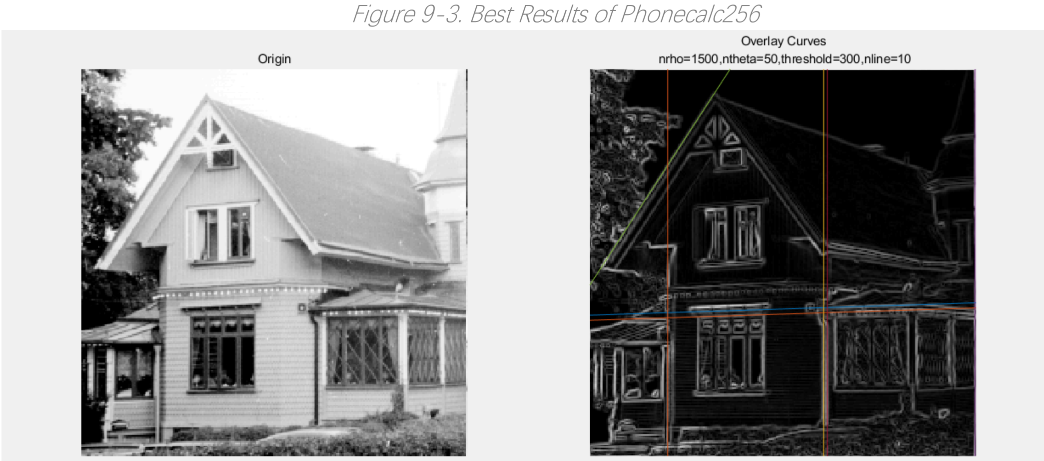


Figure 9-3. Best Results of Phonecalc256



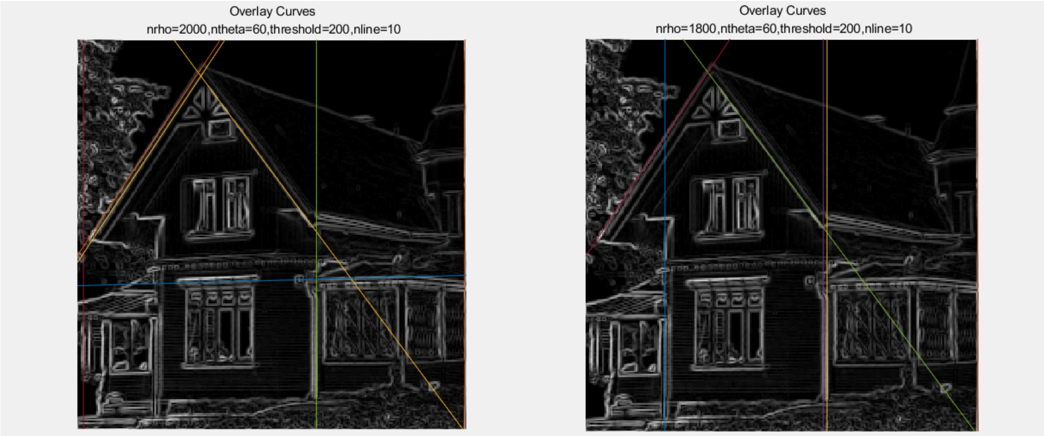


Figure 9-4. Best Results of Godthem256

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

Answers:

From the experiment above, I concluded that:

When n_ρ or n_θ is too small (n_ρ lower than 100 and n_θ lower than 10), the accuracy will be extremely low. The output edges will be some random lines (Shown below).

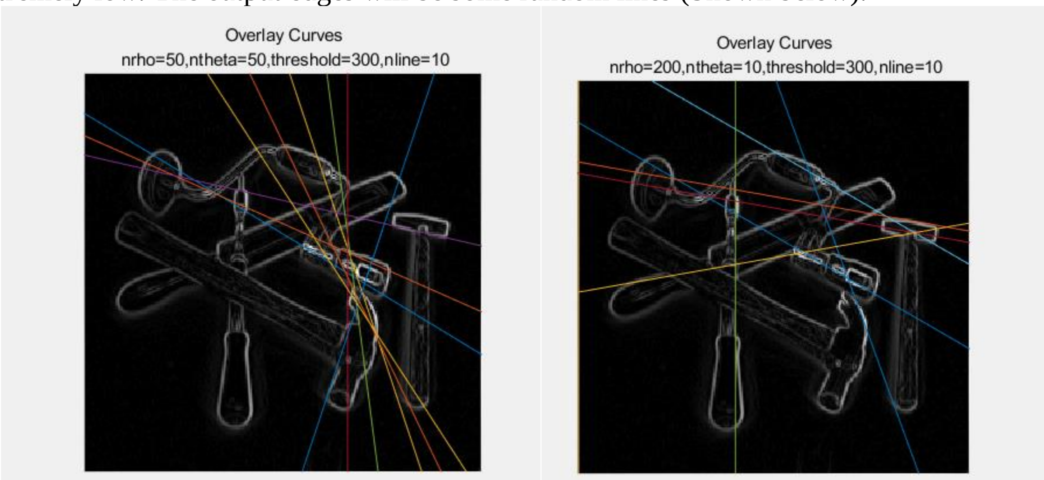


Figure 10-1. Tests of Parameters

When n_ρ or n_θ get too larger (n_ρ larger than 2000 and n_θ larger than 200), the accuracy will be extremely low. The output edges will be on same edges.

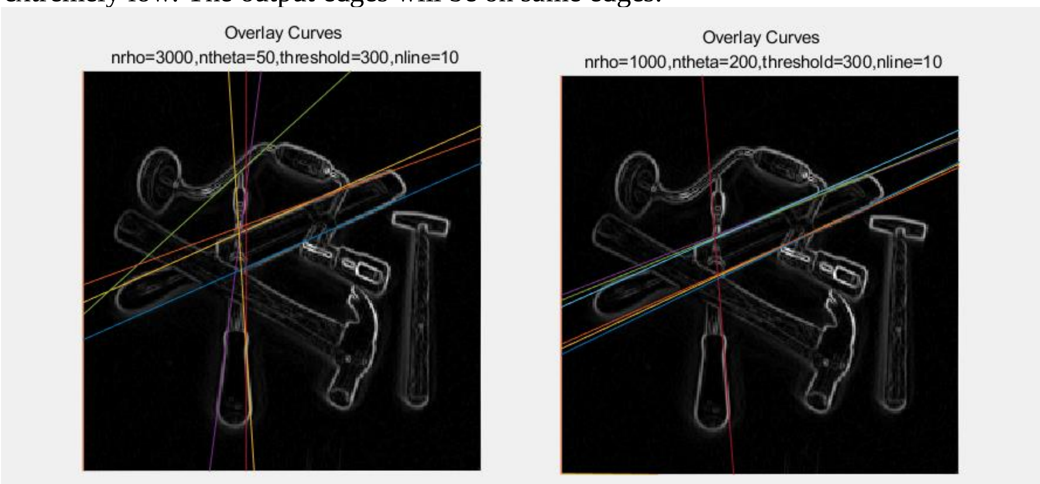


Figure 10-2. Tests of Parameters

The time is measured by MATLAB function tic and toc.

```
tic;
[linepar, acc] = houghedgeline(pic, scale, gradmagnthreshold, nrho, ...
                               ntheta, nlines, verbose);
toc;
```

Several groups of parameters are tested.

Image	n_ρ	n_θ	time (second)
tools (few256)	1000	1000	4.025447
tools (few256)	2000	1000	5.861931
tools (few256)	1000	2000	7.218010
tools (few256)	2000	2000	11.149137

Chart 1. Computational Time with Different Size of Accumulator

There will more loops when either of n_ρ or n_θ increases, From the chart we can see that the computational time will be increased when either of n_ρ or n_θ is increased, and n_θ influents the computational time more than n_ρ.

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:

In some cases (when n_ρ and n_θ are relatively large), the edges with a large value in accumulator should not be distinguished evidently, on the contrary, those edges with a small value should be separated with the former ones. For example, 1 should be distinguished with 101, but 1000 doesn't need to be distinguished with 1100.

The function I used is

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

```
acc(rhoIndex, thetaIndex) = acc(rhoIndex, thetaIndex) + 1;
```

```
acc(rhoIndex, thetaIndex) = acc(rhoIndex, thetaIndex) + ...
    log(magnitude(round(x), round(y)));
```

Some tests are shown below:

Normal Accumulator | Updated Accumulator

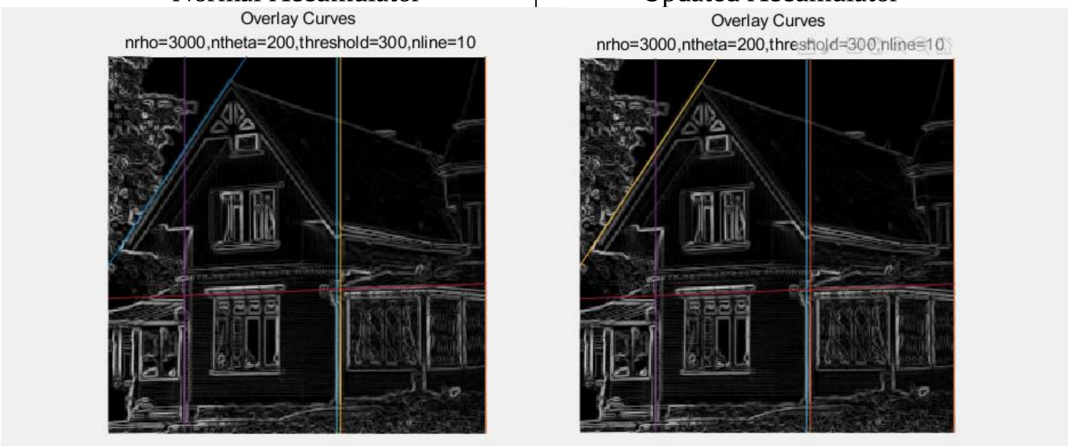


Figure 11-1. Tests of Accumulator's Methods

On the contrary, in those cases (when n_ρ and n_θ are relatively small), the edges with a large value in accumulator need to be distinguished evidently, but the difference between those edges with a small value can be ignored. For example, 1 doesn't need to be distinguished with 101, but 1000 must be distinguished with 1100.

The function I used is

$$\Delta S = e^{magnitude}$$

```
acc(rhoIndex, thetaIndex) = acc(rhoIndex, thetaIndex) + 1;
```

```
acc(rhoIndex, thetaIndex) = acc(rhoIndex, thetaIndex) + ...
    exp(magnitude(round(x), round(y)));
```

Some tests are shown below:

Normal Accumulator | Updated Accumulator

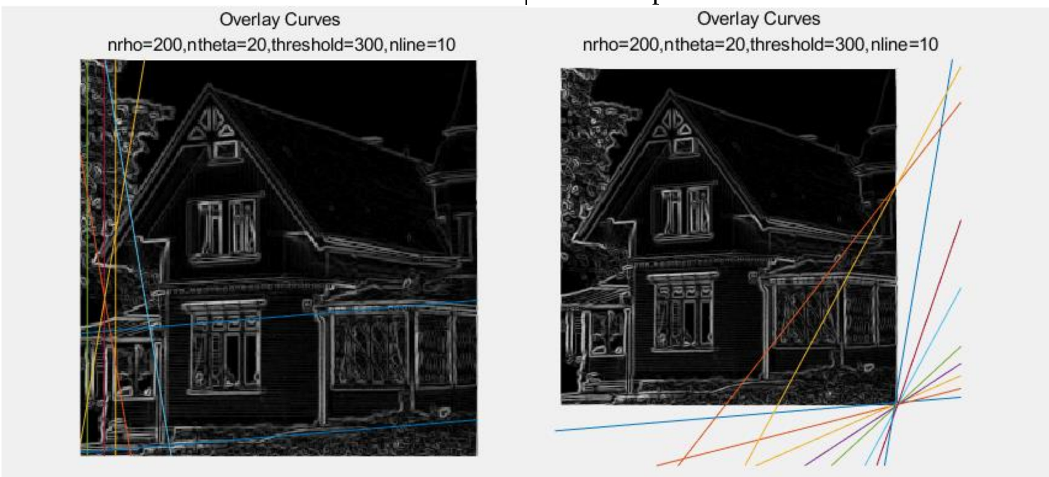


Figure 11-2. Tests of Accumulator's Methods

After using the functions above, the threshold is not that important as before with a finite number of lines. The values of accumulator of each (ρ, θ) can be sorted more efficiently. And it does improve the results.