Report of MP5

1. Canny Edge Detector

It designed to detect a wide range of edges in images. It consists of following steps, noise reduction, gradient the intensity of image, non-maximum suppression, thresholding, and edge tracking by hysteresis, this detector is effective in delineating detailed edges structure that would be missed by simpler edges detection methods.

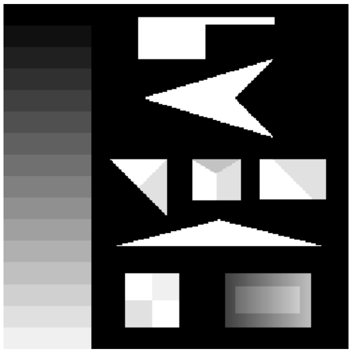
2. Algorithm Implementation

The whole program includes, GaussSmoothing, ImageGradient,FindThreshold, NonmaximaSupress, and Edge Linking 5functions.

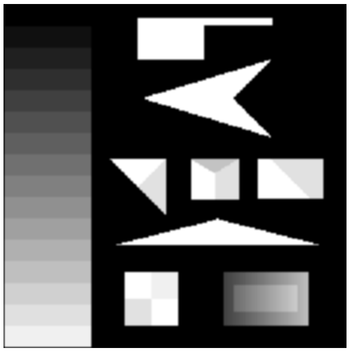
- **GaussSmoothing(image, N, Sigma):**
  ‘N’ for the [height, width] of gaussian kernel, ‘Sigma’ the standard deviation of gaussian function. In this function, we first create the gaussian kernel, normalize it. Then use ‘convolve2d’ method to convolute image.
- **ImageGradient(image, mode):**
  Mode for [Robert cross, Sobel], use the smoothed image to gradient and get magnitude and direction of edge map.
- **FindThreshould(mag_image, percentangeOfNonEdge):**
  ‘mag_image’ come from ImageGradient, parameter percentangeOfNonEdge is what we define, then we use histogram to get [high, low] threshold. In here low threshould equals to half of high threshould .
- **NonmaximaSupress(mag_image, direction):**
  ‘mag_image’ , ‘direction’come from ImageGradient. Use quantization method to implementation. And LUT method to get four major directions.
- **EdgeLinking(supr_img, T_high, T_low):**
  ‘supr_image’ from NonmaximaSupress, [T_high, T_low] from FindThreshoulding,

3. Parameter and detector Comparison

3.1 (N, Sigma)



N = [1,1]



N = [3,3]

N = [5,5]

Fix sigma =1 , change the size of kernel, the more of the size, the smoother/ grounder of the edge of image

Sigma = 1

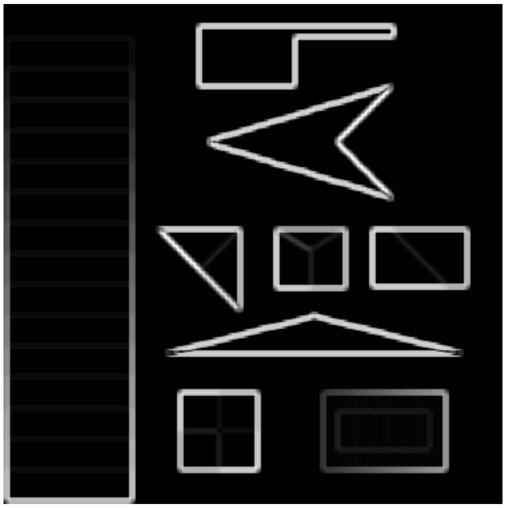
Sigma = 3

Sigma = 5

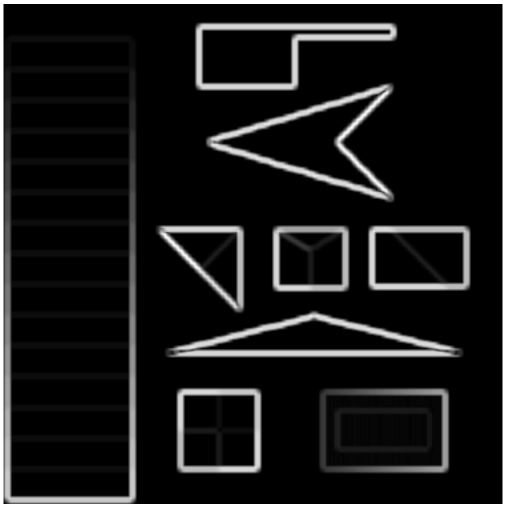
Fix N= [5,5], the more of value sigma, the unclear of each image, due to it more focus on the center and use smaller data.

So, we use N=[3,3], sigma=3 to implementation next steps.

3.2 edge detector [Robert, Sobel]



Robert



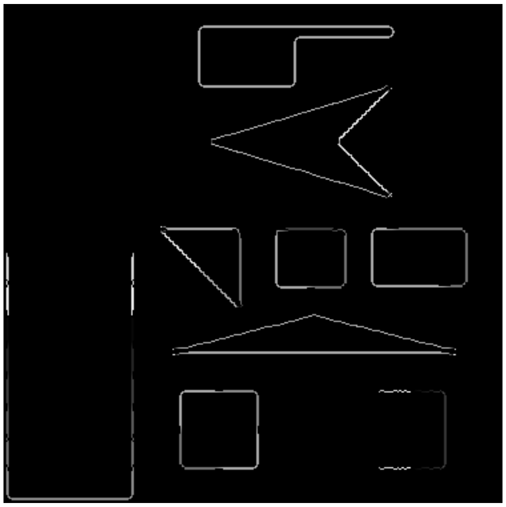
Sobel

Sobel operator has a more large kernel size, 3*3, it is robust than rebort cross. It can incorporate nore neighboring pixel when compute gradient, so it is less sensitive to noise, then we can get a more smoother gradient

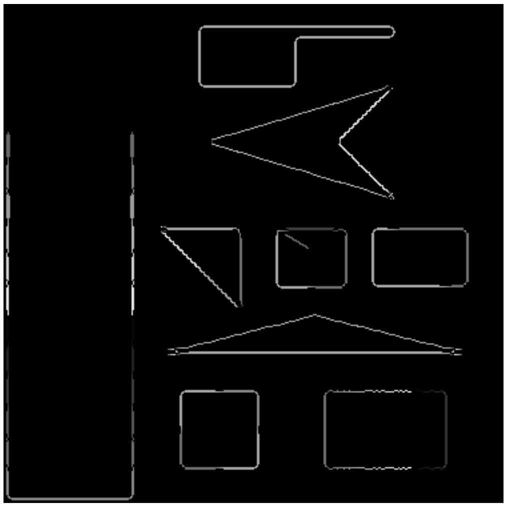
3.3 percentageOfNonEdge

percentageOfNonEdge = 0.99

High Threshold

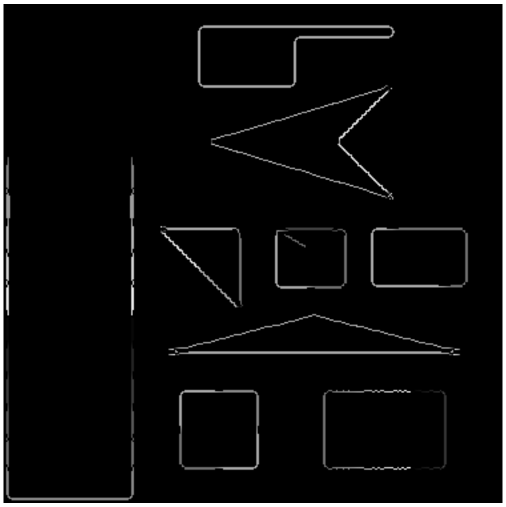


Low Threshold

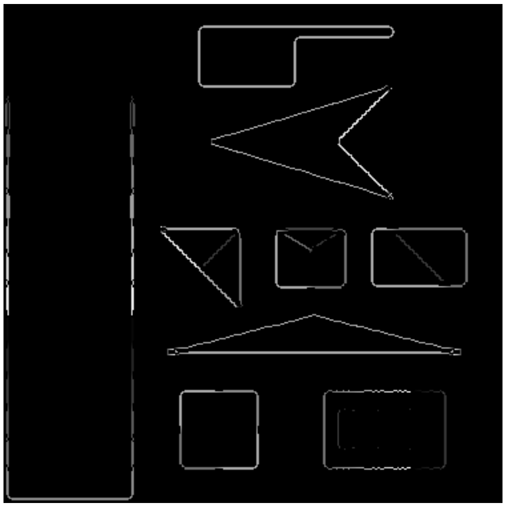


percentageOfNonEdge = 0.9

High Threshold



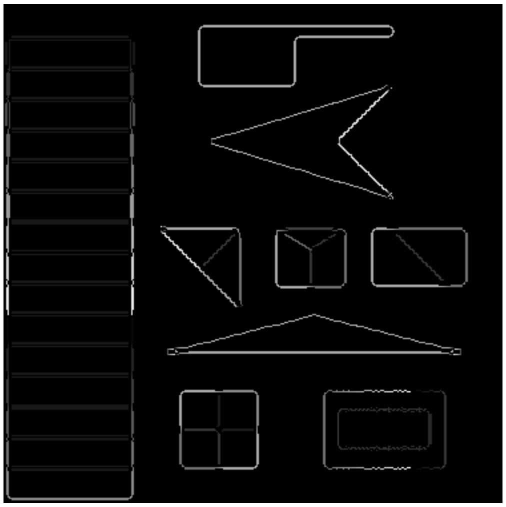
Low Threshold

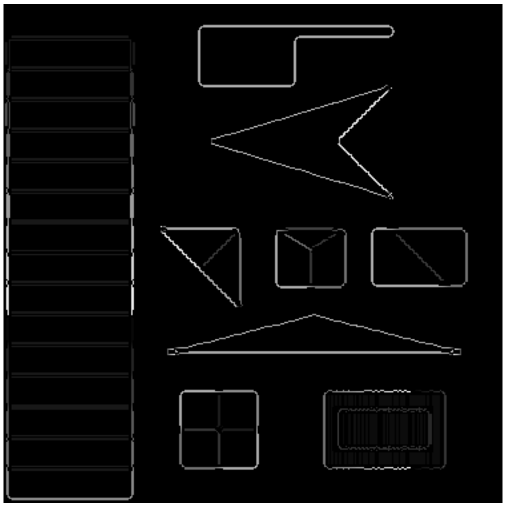


percentageOfNonEdge = 0.8

High Threshold

Low Threshold





From the result, the more value of **percentageOfNonEdge**, the fewer edge will get, due to most of them are considered as noise, but the value to low, the edges will be too detailed, too more, include more noise, so we fix the **percentageOfNonEdge = 0.9**

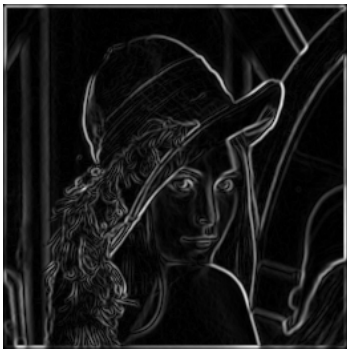
4. Result analysis

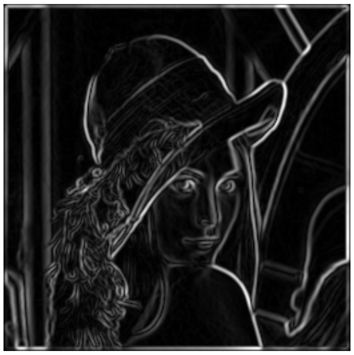
In this analysis part, I will show the whole 4 test images results, but focus on 'lena.bmp' to analysis performance, and compare the different model, how to determine the parametes value

4.1 Lena:

Gaussian Smoothing: in the part one, we define the N=[height, width] = [3,3], Sigma=3 to smooth image. From the result, we can see the borders are rounder.

Calculating Image Gradient,[compare the Robert and sobel model]: in part two, we can use 'robert cross' or 'sobel' model to calculate, but from the result we can see, model 'sobel' are more robust, it shows more detailed edge then 'robert'. So we select 'sobel' model for following steps


(Robert model)

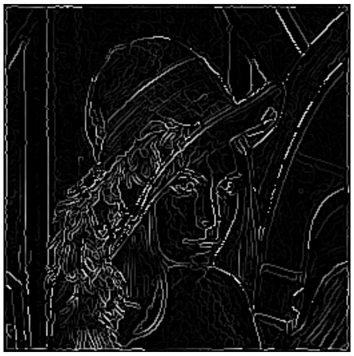

(sobel model)

<u>Selecting High and Low Thresholds</u>: in part three, we use histogram to determine [high, low] thresholds, in here, the important is the value of parameter 'percentageOfNonEdge', we make some test and get these results:

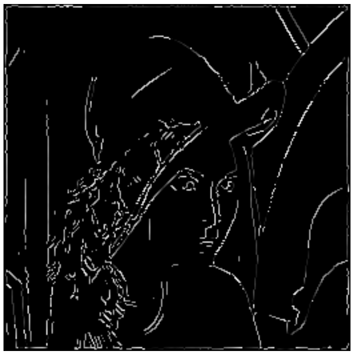
percentageOfNonEdge value	[high, low] thresholds
0.99	[235, 235/2]
0.9	[143, 143/2]
0.8	[96, 96/2]

this parameter represents the percentage of pixels in iamge that are consider non-edges, we want to use it to determine the major edge in image, so we can drop weaker one and noise. We can combine the result in 5th function to determine the value. Now, we know the more the value, the high the high threshold, and we will have fewer edges, and fewer nosie. So we can not select it to high or to low, we now select to 0.9.

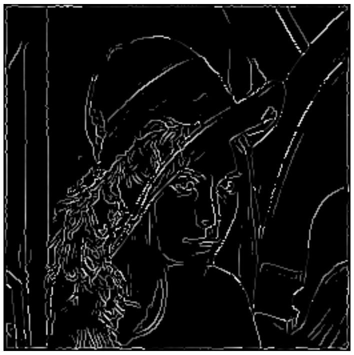
<u>Supressing nonmaximal</u>: get it based on imageGradient, 'sobel' model, but nonmaximal image include more clear edges, just 'thin' edge. That some of them are not major edges, some of them are noise should be remove, due to now use local maxima.



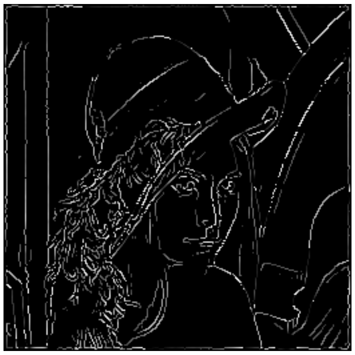
Edge Linking: after determine high and low threshold, we use percentageOfNonEdge = 0.9 get these two edge map, it is clear, the T_high image has low noise, but the edge is blur, the T_low image has more thin edge than T_high. Comibne the T_high and T_low images, we get the edge link image. To comparsion, this image has more clear and detailed edge then T_high image, and fewer noise then T_low image.



(T_high)

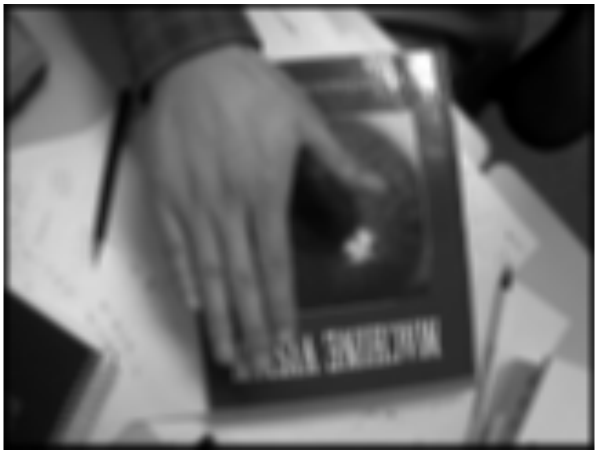


(T_low)

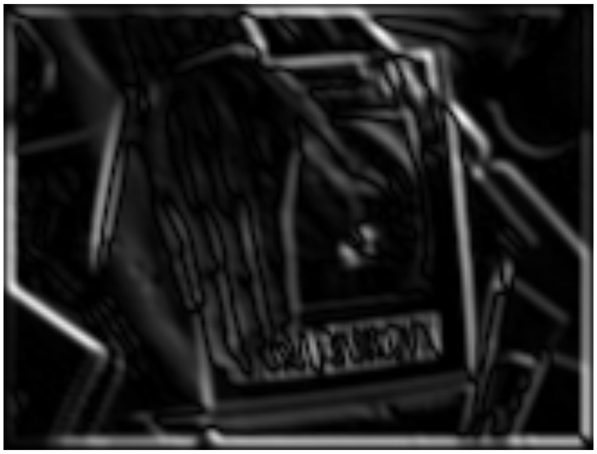


(edge link)

4.2 joy1.bmp:

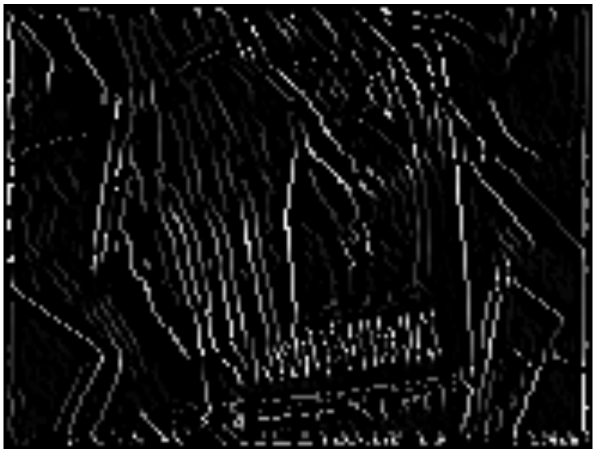


Gaussian Smoothing

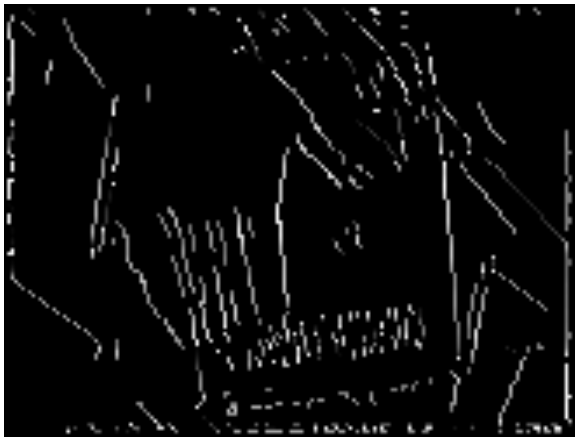


Gradient

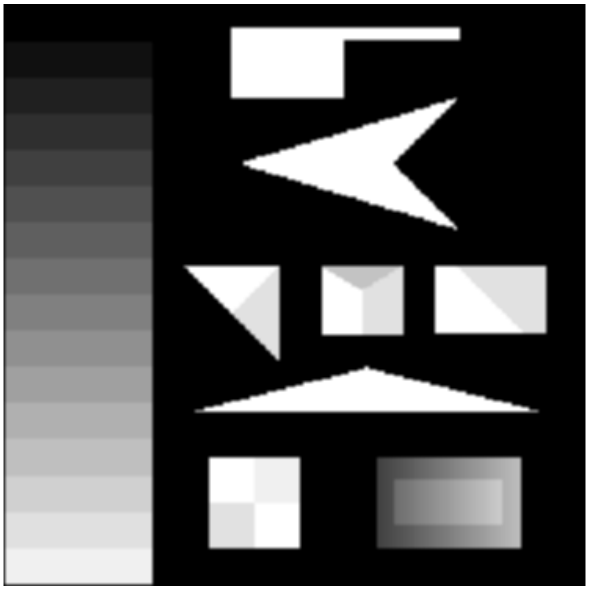
NonMaxima Supress



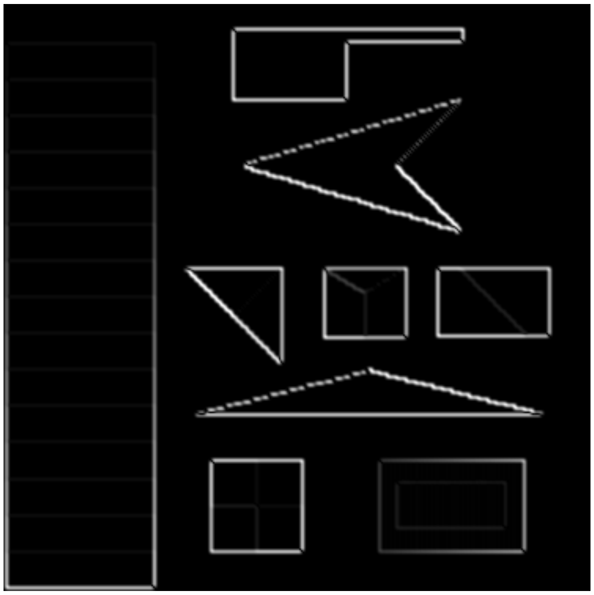
Edge Linking



4.3 test1.bmp:

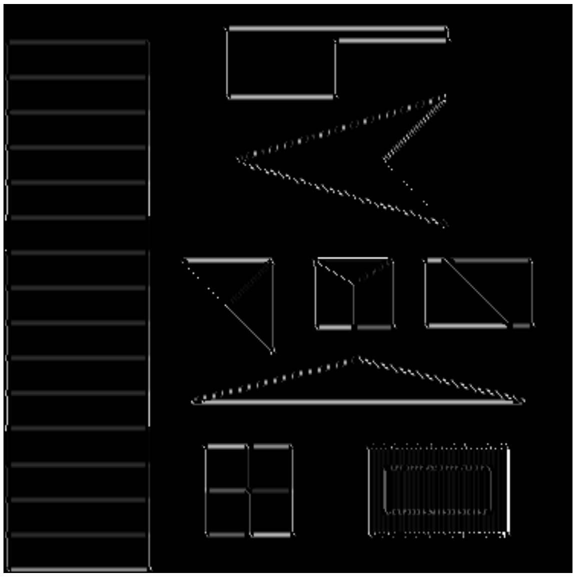


Gaussian Smoothing

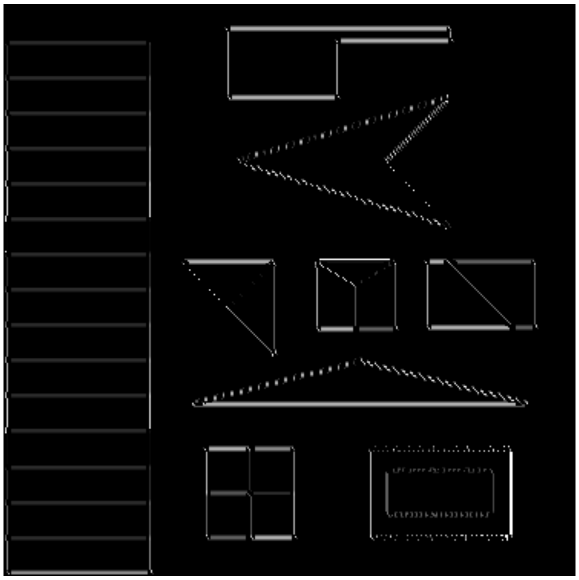


Gradient

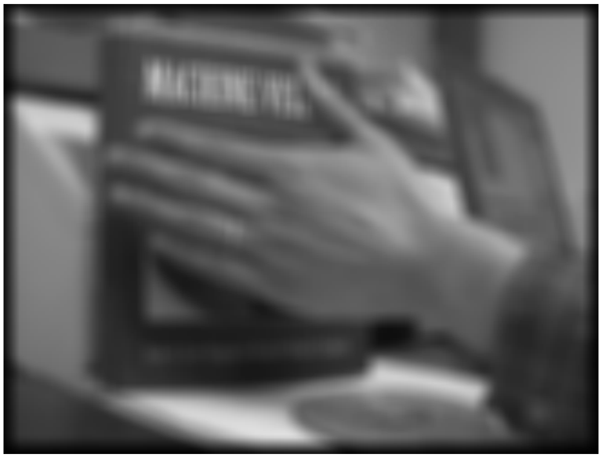
NonMaxima Supress



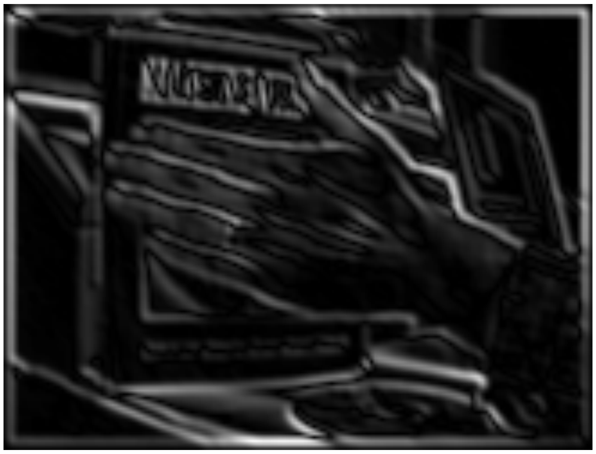
Edge Linking



4.4 pointer1.bmp

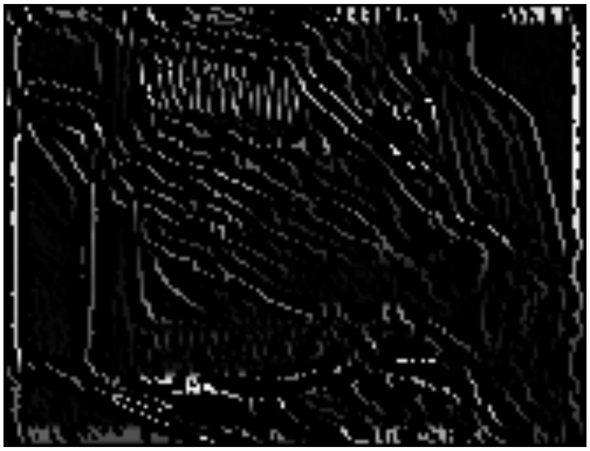


Gaussian Smoothing



Gradient

NonMaxima Supress



Edge Linking