CMPEN 555: Digital Image Processing II

Computer Project # 1:

Hit-or-Miss Transform

Naichao Yin, Wenrui Wang, Zekai Liu

Date:1/31/2020

A. Objectives

1. Threshold image to binary-value
2. Remove salt and pepper noise
3. Create structuring elements
4. Do hit-or-miss transform
5. Reconstruct target image

B. Methods

First, we set the optimal threshold (1.1) to turn the original image into a true binary-valued image in order to segment the object(disks) from the image that we want to operate. If src(x,y) is larger than thresh value, the new pixel value dst(x,y) is MaxVal(255), otherwise is 0.

$$dst(x,y) = \begin{cases} maxVal, & src(x,y) > thresh \\ 0, & otherwise \end{cases} \tag{1.1}$$

The reason we set the binary threshold at 127 is that by presenting the histogram of the original image (Figure1). If the histogram of the gray level is bimodal, the corresponding gray level of the valley bottom between the two peaks is selected as the threshold. In other words, most of the foreground pixels (dark areas) are less than 127, the background pixels are the opposite.

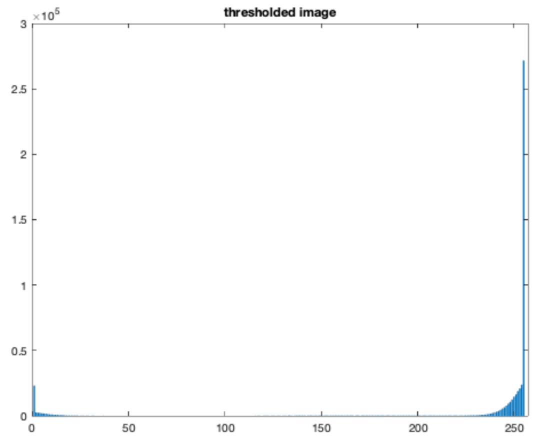

Figure 1. Histogram

Q: How to reduce the salt-and-pepper noise:

We denoise the binary image with salt and pepper noise by using the self-defined close operation and then the self-defined open operation:

Open operation: the process of erosion followed by dilation. Used to eliminate small objects (pepper noise).

Matlab (close function): DilationOwn (ErosionOwn (src, kernel), kernel) → closed image

Closed operation: the process of dilation followed by erosion. It is used to fill small voids (salt noise) in the disks without changing their area significantly.

Matlab (open function): ErosionOwn (DilationOwn (src, kernel), kernel) → opened image

We use a 3 by 3 kernel because this is the optimal size to remove salt and pepper noise without affecting any foregrounds(disks). The structure is symmetric about the origin. The dilation operation uses structure element to scan every pixel of image matric and to do or operation with the binary image it covers. If one element is 255, the pixel of the result image is 255. Otherwise it is 255. The structure element is used for erosion operation. Each pixel of image is scanned. The and operation is performed with the structure element and the binary image it covers. If one pixel is 0, the pixel of the result image is 0. Otherwise it is 255. The dilation operation and erosion operation are the basis of opening by reconstruction.

Q: How to select structuring elements:

We use self-defined function “disk_lma” and “hold_lma” to build the structuring elements.

Processing flow:

1. Determine the diameter of the disk
2. Create a matrix of zeros
3. Scan the image and find the radial distance from the center. If the distance is less or equal to the specified radius then make pixels 1
4. Invert the image to create disk window (for disk)

Matlab: disk_lma (radius) → the structuring element A

Matlab: hole_lma (radius) → the structuring element B(W-A)

We create a black disk A_b with a radius of 36 and a square window B_b ($W-A_b$) that contains a white disk with a radius of 31 to be a structuring element of detecting the biggest disks.

Figure 2. A_b(left) and B_b(right)

We create a black disk A_s with a radius of 10 and a square window B_s ($W\text{-}A_s$) that contains a white disk with a radius of 8 to be a structuring element of detecting the smallest disks.

Figure 3. A_s(left) and B_s(right)

We gradually increasing the disk radius from 1 until the smallest or the biggest disks disappear, and set the corresponding radius to A_s and A_b. To better fit the target object, we set the radius of the windows (B_s, B_b) to be slightly smaller than the disks (A_s, A_b).

Then we start to design a hit-or-miss transform (1.2):

$$(X \ominus A^s) - (X \oplus B^s) \tag{1.2}$$

We start to reverse the cleaned main image (X), then use structuring element A for erosion and structuring element B for dilatation, next we create the self-defined function "compare_lma" to compute the operation of subtracting the dilated image from the eroded image.

Processing flow:

1. Create an M * N temp image with gray level 1(white)
2. Check pixel by pixel grey value of two images
3. If both images own same gray level then put zero in the particular pixel of image temps

Matlab: compare_lma (scr1, scr2) → Output image

After these operations, we get the location of the smallest disks and the biggest disks respectively and merge them into an image.

Finally, we restore the biggest and smallest disks using by repeating the self-defined function “DilationOwn” 50 times, which dilate the hit-or-miss image with a 3x3 mask, intersect the result with the binary-valued image and reconstruct the operated image again, which gives us the reconstructed image of the disk with only the biggest and the smallest ones. The formula of reconstruction is shown as (1.3), in which R represents the original image, B^S represents 3x3 mask and X represents the location image. After 50 times operation, reconstruct result will be obtained.

$$O_R^{(n)}(F) = \left[\left(X \oplus B^s\right) \cap R\right]^{50} \tag{1.3}$$

Q: If do not apply the small close/open operation.

The salt and pepper noise will have a great impact on hit-or-miss transform. Because hit-or-miss transform is designed for special size disks and noise will change their size when erosion and dilation. Some disks won’t be recognized.

C. Results

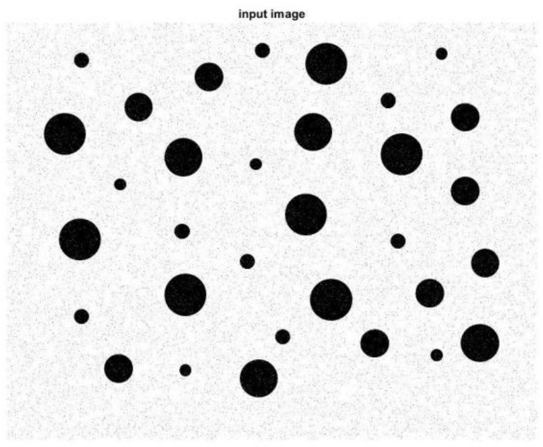

Figure 4. Original Image

Figure 4 is the original image with gray value between 0 and 255, which is disturbed by salt and pepper noise.

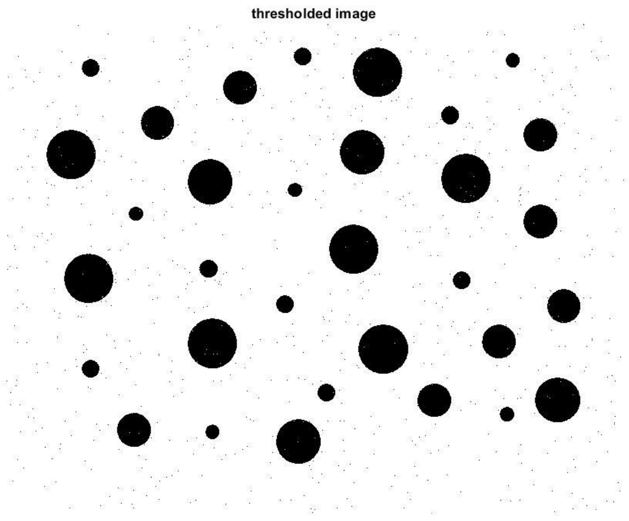

Figure 5. Binary-valued Image

Figure 5 is the binary-valued image with gray value either 0(disks and pepper noise) and 255(background and salt noise)

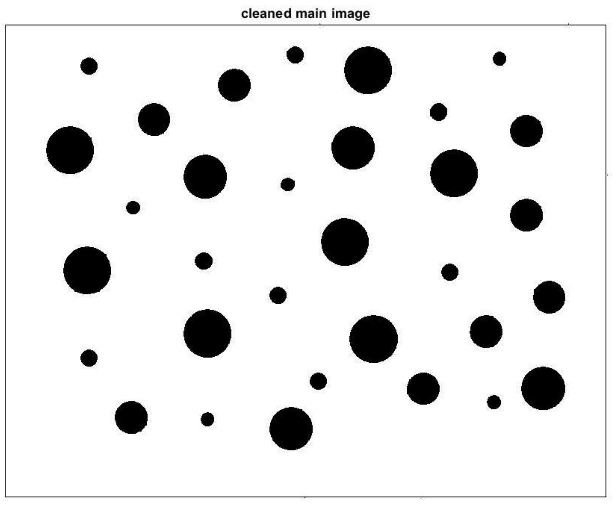

Figure 6. Denoised Image

The image shown in Figure 6 removes the whole salt and pepper noise but present four sides of the black border, because the out border cannot be operated.

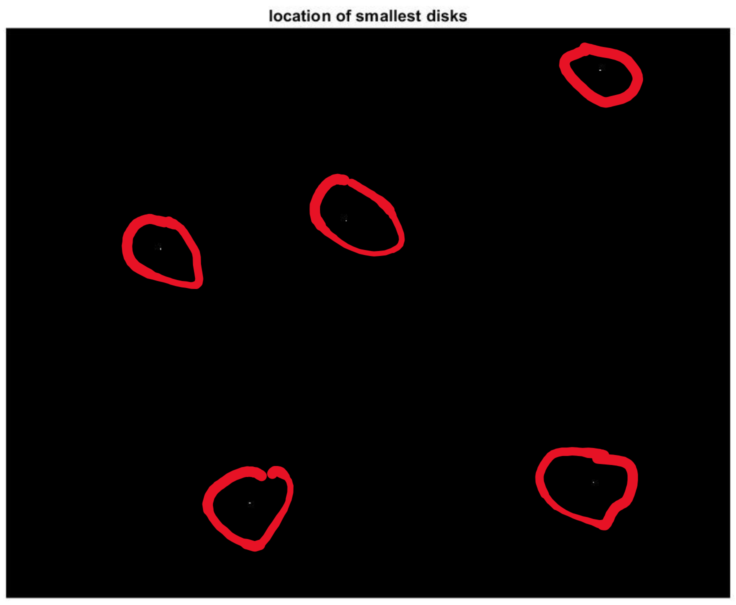

Figure 7. Location of smallest disks (without noise)

The image shown in Figure 7 is the result of location of smallest disks when pepper and salt noise are filtered. Because our codes are designed the situation where white points are valid, we inverse the original image. The white points are the locations of black smallest disks in original image.

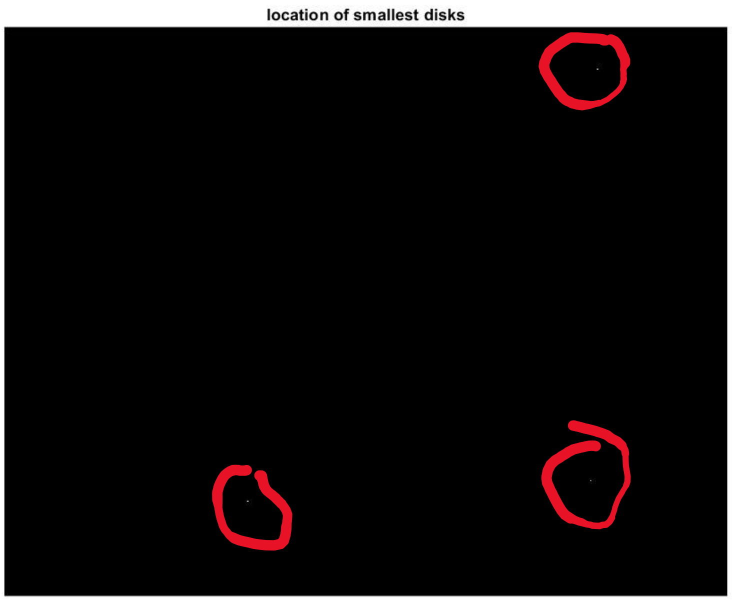

Figure 8. Location of smallest disks (with noise)

The image shown in Figure 8 is the result of location of smallest disks when pepper and salt noise are not filtered. It is shown that some of smallest disks were not be located if the noise existed.

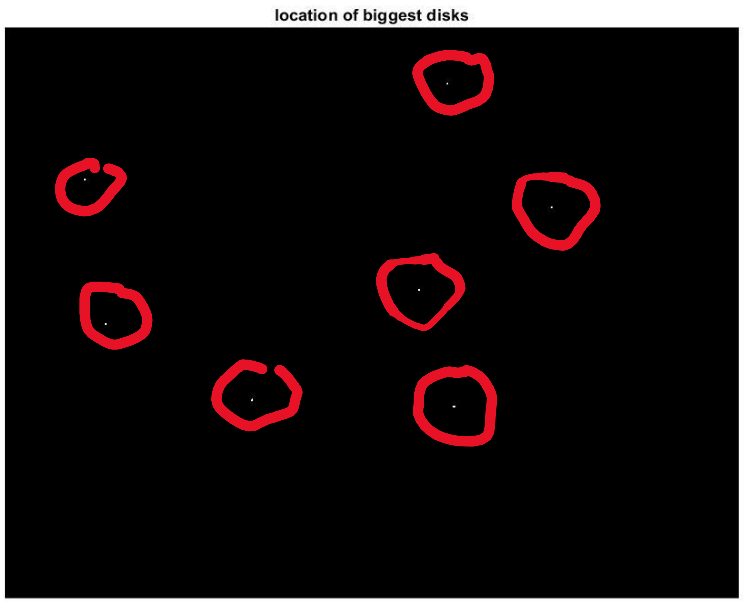

Figure 9. Location of biggest disks (without noise)

The image shown in Figure 9 is the result of location of biggest disks when pepper and salt noise are filtered.

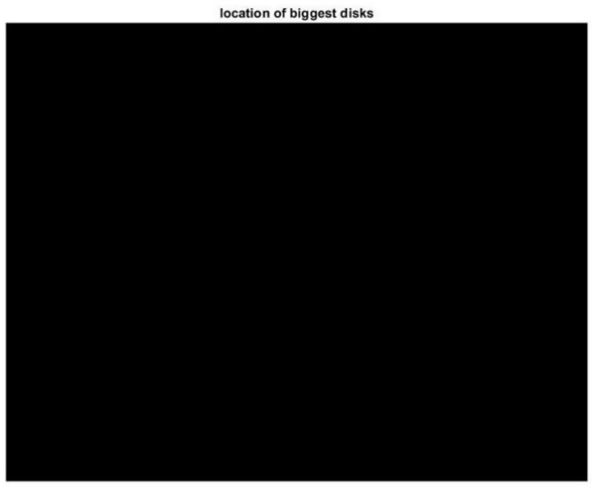

Figure 10. Location of biggest disks (with noise)

The image shown in Figure 10 is the result of location of biggest disks when pepper and salt noise are not filtered. It is shown that all of smallest disks were not be located if the noise existed.

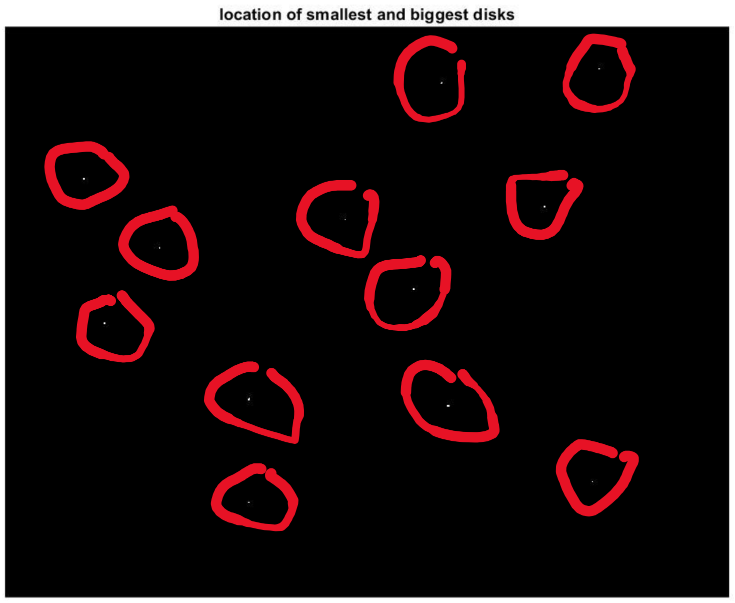

Figure 11. Location of smallest and biggest disks (without noise)

The image shown in Figure 11 is the result of location of smallest and biggest disks when pepper and salt noise are filtered.

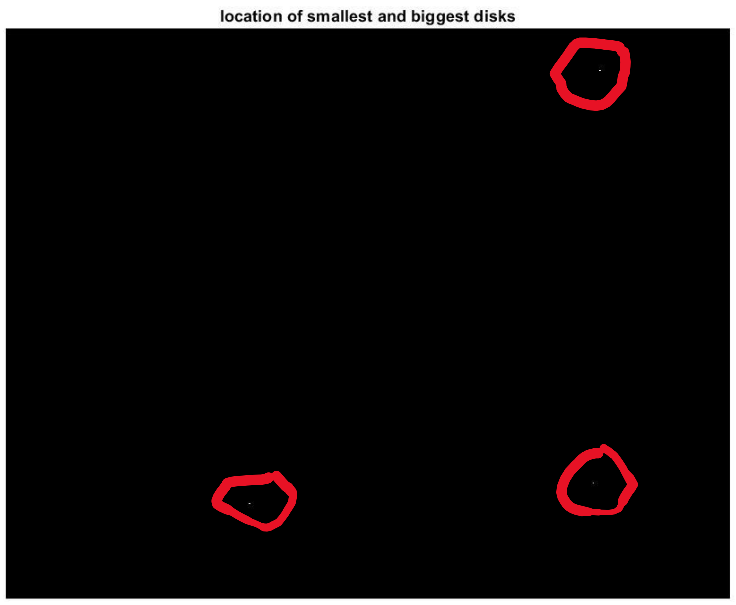

Figure 12. Location of smallest and biggest disks (with noise)

The image shown in Figure 12 is the result of location of smallest and biggest disks when pepper and salt noise are filtered.

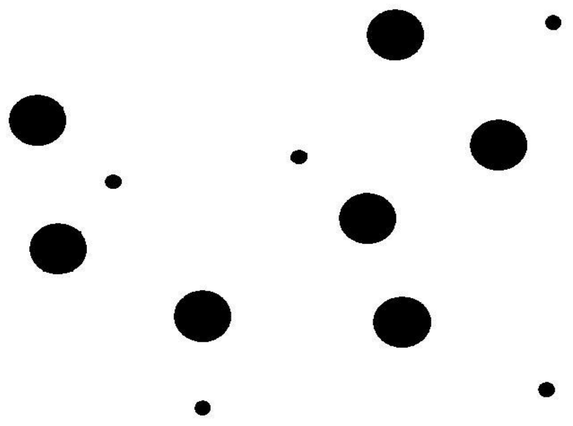

Figure 13. reconstruct Result (without noise)

The image shown in Figure 13 is the reconstruct result of location of smallest and biggest disks when pepper and salt noise are filtered.

Figure 14. Final Result (with noise)

The image shown in Figure 14 is the reconstruct result of location of smallest and biggest disks when pepper and salt noise are not filtered.

D. Conclusion

In this project, we do the following operations: threshold image to binary-value, using close/open filter to fill holes and reduce salt and pepper noise, create structuring elements, detecting the smallest and largest disks by applying erosion and dilation, reconstruct target image. Different parts of the image could be picked up by applying hit or miss transform. However, before applying the transform, the image must be processed to reduce the noise. From the result it is shown that with the noise, some big disks and small disks disappear after reconstructing target image.

In the future, if images need to be processed or operated, reducing noise must be done at the beginning. We can apply different filter to reduce noise based on different types of noise, such as mean filter, opening & closing. Besides, to do the Hit or Miss Transform, appropriate structuring elements are important.