[[1]](#footnote-1)

Lab 9

Chengpin Luo, 11510880, SUSTech

*Abstract*—In this lab, we are going to implement some morphological image processing techniques, such as erosion, dilation and connected components. We will see how the shapes of images would be changed and how some of those techniques are combined with practice, such as how to extract a specified component from an image.

*Index Terms*— Morphology, erosion, dilation, connected components

# INTRODUCTION

M

ORPHOLOGY, mathematically, is a tool for extracting image components that are useful in the representation and description of region shape, such as boundaries, skeletons, and the convex hull. In this lab, we will implement some basic morphological techniques, such as erosion, dilation and component extraction. The details of this lab is listed as:

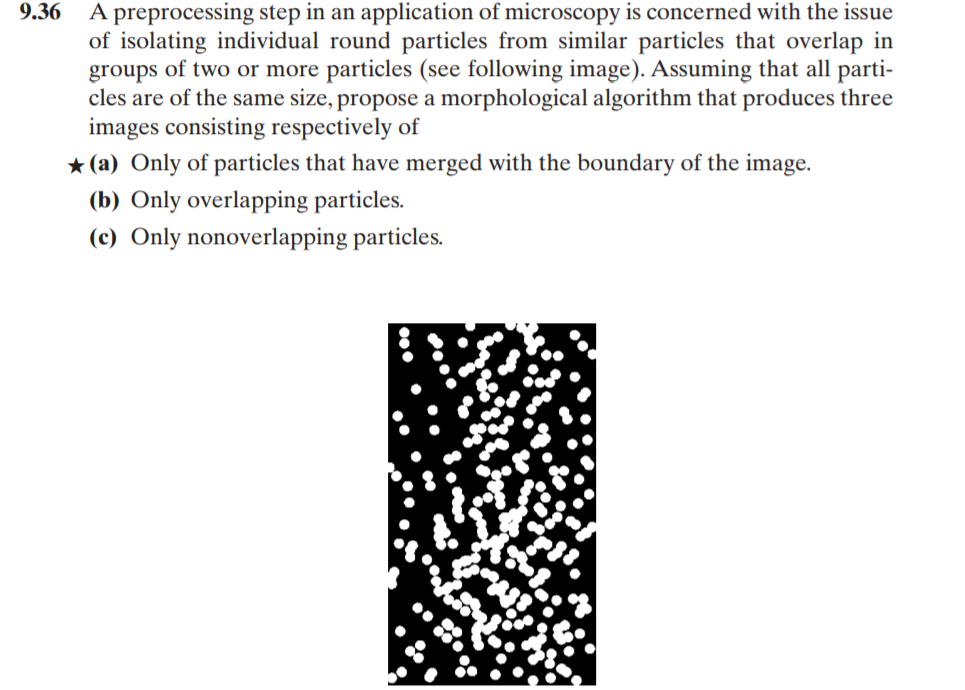
(1) Define two structural elements, perform binary dilation, erosion, opening, and closing operations on noisy\_fingerprint.pgm and noise\_rectangle.pgm.

(2) Extract the boundaries of licoln.pgm and U.pgm.

(3) Extract the connected components from connected.pgm.

(4) Problem 9.36, write a program to separate the three required sets from bubbles\_on\_black\_background.pgm.

The problem 9.36 is:



**Fig. 1.** Problem 9.36

The following sections would be constructed as: (II)Fundamentals. (III)Implementation and Analysis (V) Conclusion.

# Fundamentals

In this part, details of morphology fundamentals will be given, including: erosion, dilation, opening, closing and connected components extraction. All operations in this lab will be based on those basic techniques.

First comes erosion. With two sets A and B in Z2, the erosion of A by B is given by:

 (1)

This equation indicates that the erosion of A by B is the set of all points z such that B, translated by z, is contained in A. Equivalently, the erosion could be described as:

 (2)

On the contrast, dilation is given by:

 (3)

which indicates that the dilation of A by B is the set of all points z such that B shifted by z has at least one element overlapped with A. Equivalently, we have

 (4)

The opening of A by B is given by:

 (5)

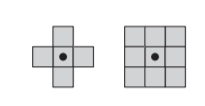
That is, A is eroded by B and then dilated by B.

The closing of A by B is given by:

 (6)

That is, A is dilated by and then eroded by B.

Basically, B is structuring element, which is small sets or subimages used to probe an image under study for properties of interest. In this lab, we will use two kinds of SEs shown in Fig. 2, which is based on 4-connectivity and 8-connectivity respectively from left to right.



**Fig. 2.** Structuring Elements

For any pixel p in S, the set of pixels that are connected to it in S is called a connected component of S. Connected component extraction is given by:

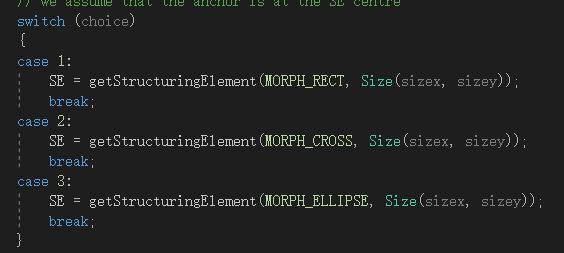
 (7)

Let A be a set containing one or more connected components, and form an array X0(of the same size as the array containing A) whose elements are 0s (background values), except at each location known to correspond to a point in each connected component in which we set to 1 (foreground value). The objective is to start with X0 and find all the connected components. The procedure stops when Xk=Xk-1. The critical thing here is that we need to find a start point at each connected component.

# Implementation and analysis

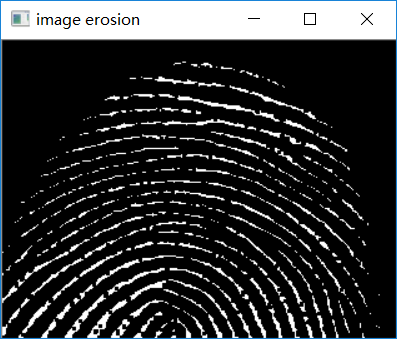
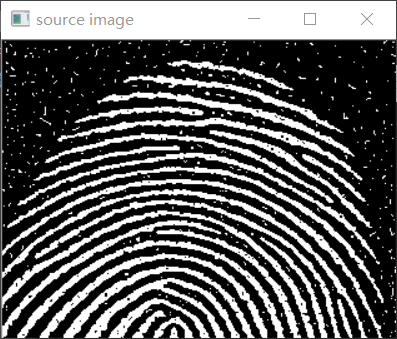
In this part, we are going to implement the morphological techniques discussed in the previous sections.

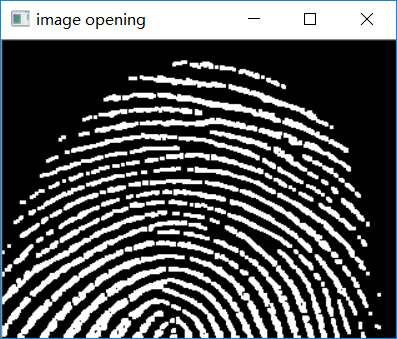
Firstly, we used the two structuring elements in Fig. 3 to erode the given images. To construct a SE, there is a function call getStructuringElement in OpenCV, where we can define three types of SE, rectangle, cross and ellipse and their size.

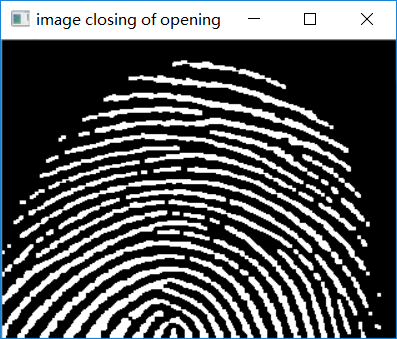


**Fig. 3**. Getting Structural Elements

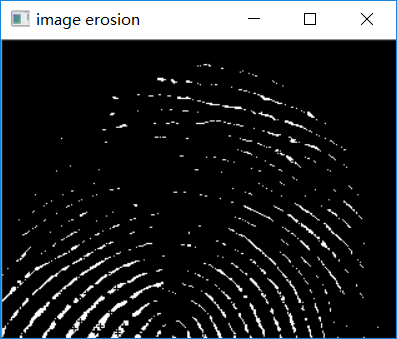
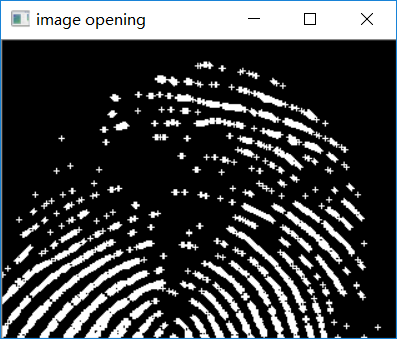
Using OpenCV function “erode” and “dilate”, we can simply get the results shown in Fig. 4, where the size of SE is set to 3x3 and the type of SE is rectangle. From left to right, from top to bottom, the image presents source image, erosion, opening, dilation of the opening image and closing of the opening image, respectively. We can see that using erosion, all of the noise are eliminate because the size of SE is larger than all of the noise. And using opening, we made the eroded image looks thicker, however, with some gaps between fingerprint ridges exists. To eliminate this, we implement the dilation of opening image. However, this thickened the ridges. To eliminate this, we implemented the erosion again, that is, the closing of the opening of source image, shown in the last picture in Fig. 4. Having said that, there are still some gaps exist.





**Fig. 4.** Erosion, Opening, Dilation and Closing (3x3 rectangular SE)

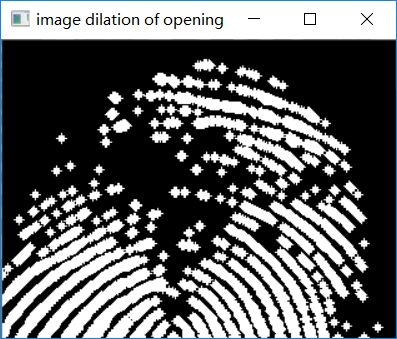
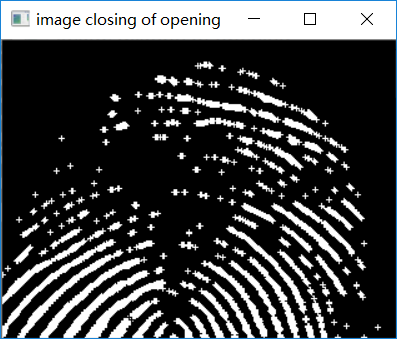
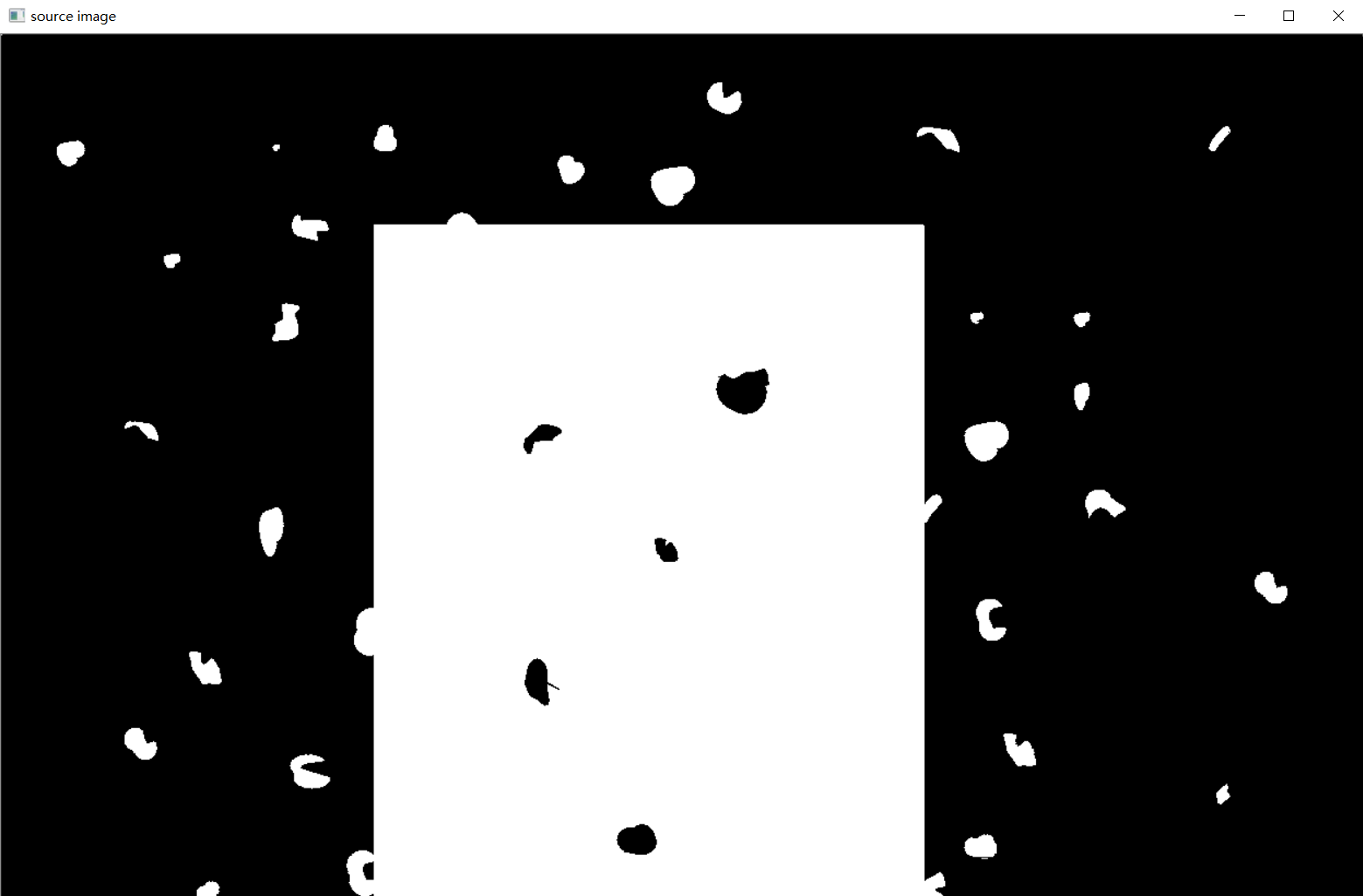
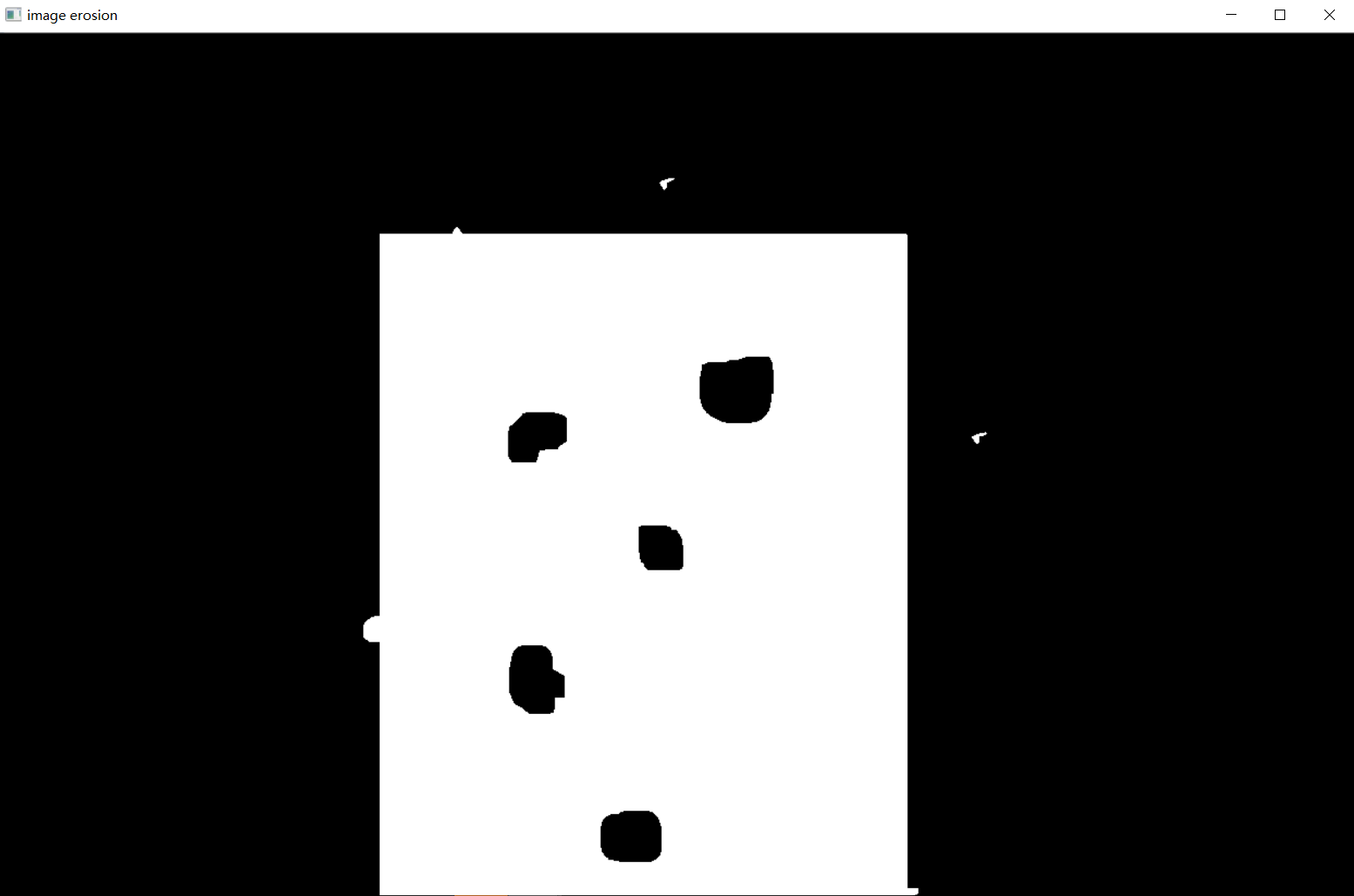
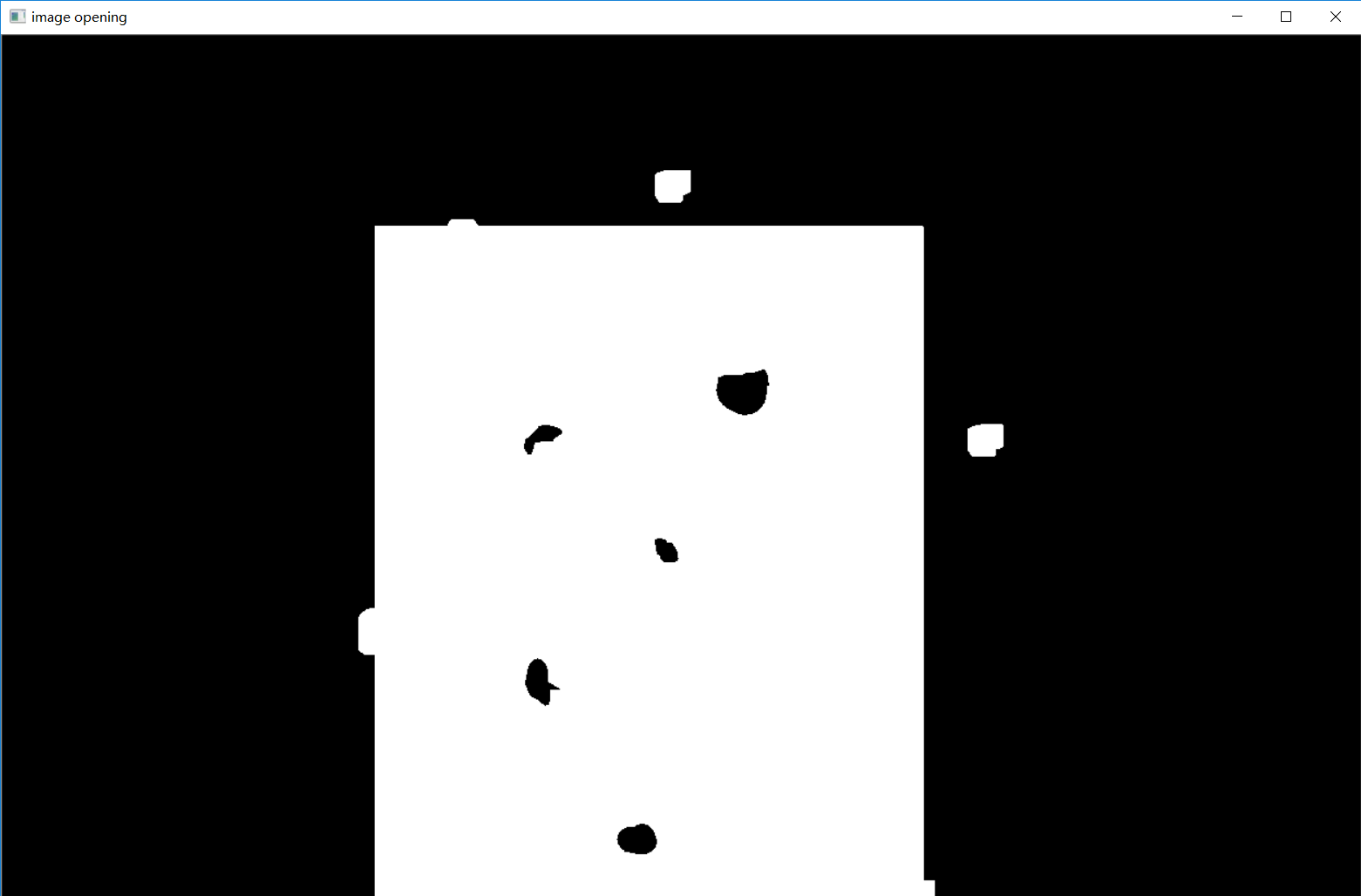
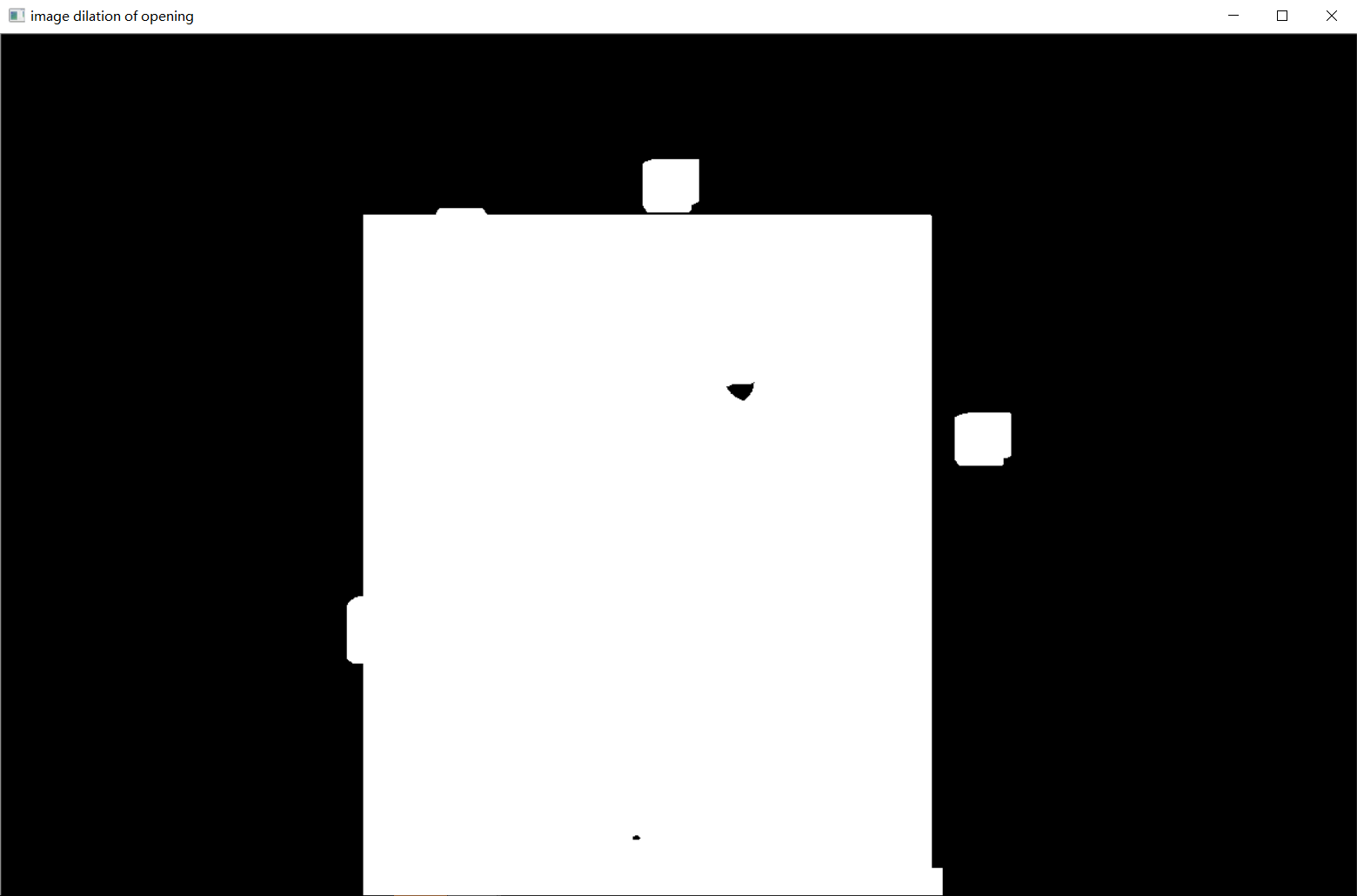
 

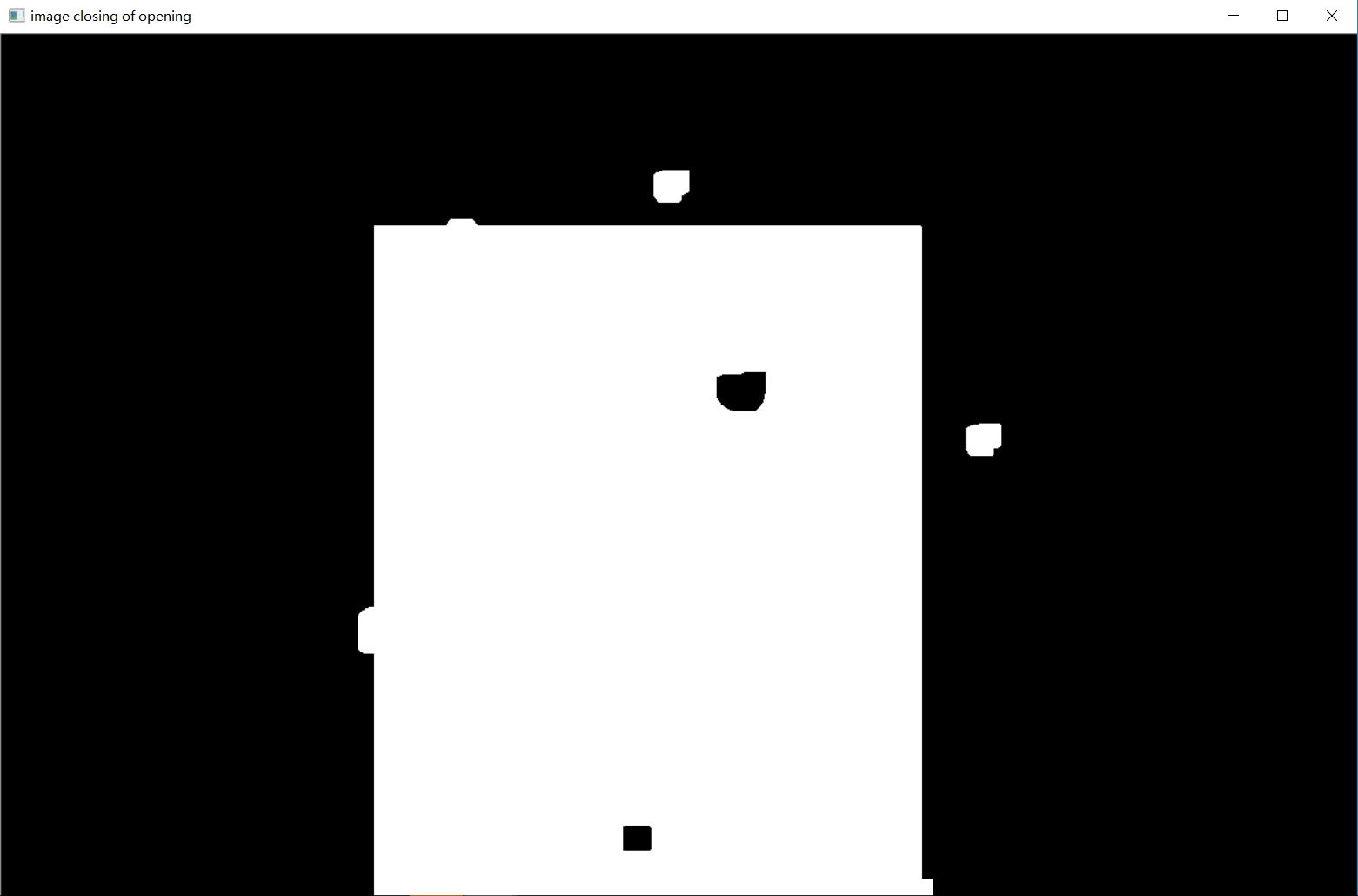
Fig. 5. Erosion, Opening, Dilation and Closing (5x5 cross SE)

When we change the type of SE to cross, with size as 3x3, we could not see many differences. Instead, we changed the size to 5x5. The result is shown in Fig.5. We can see that when the size of the SE is larger than most of the ridges, the erosion would thin the image severely. And through dilation, crosses took the places of smaller dots in the ridges.

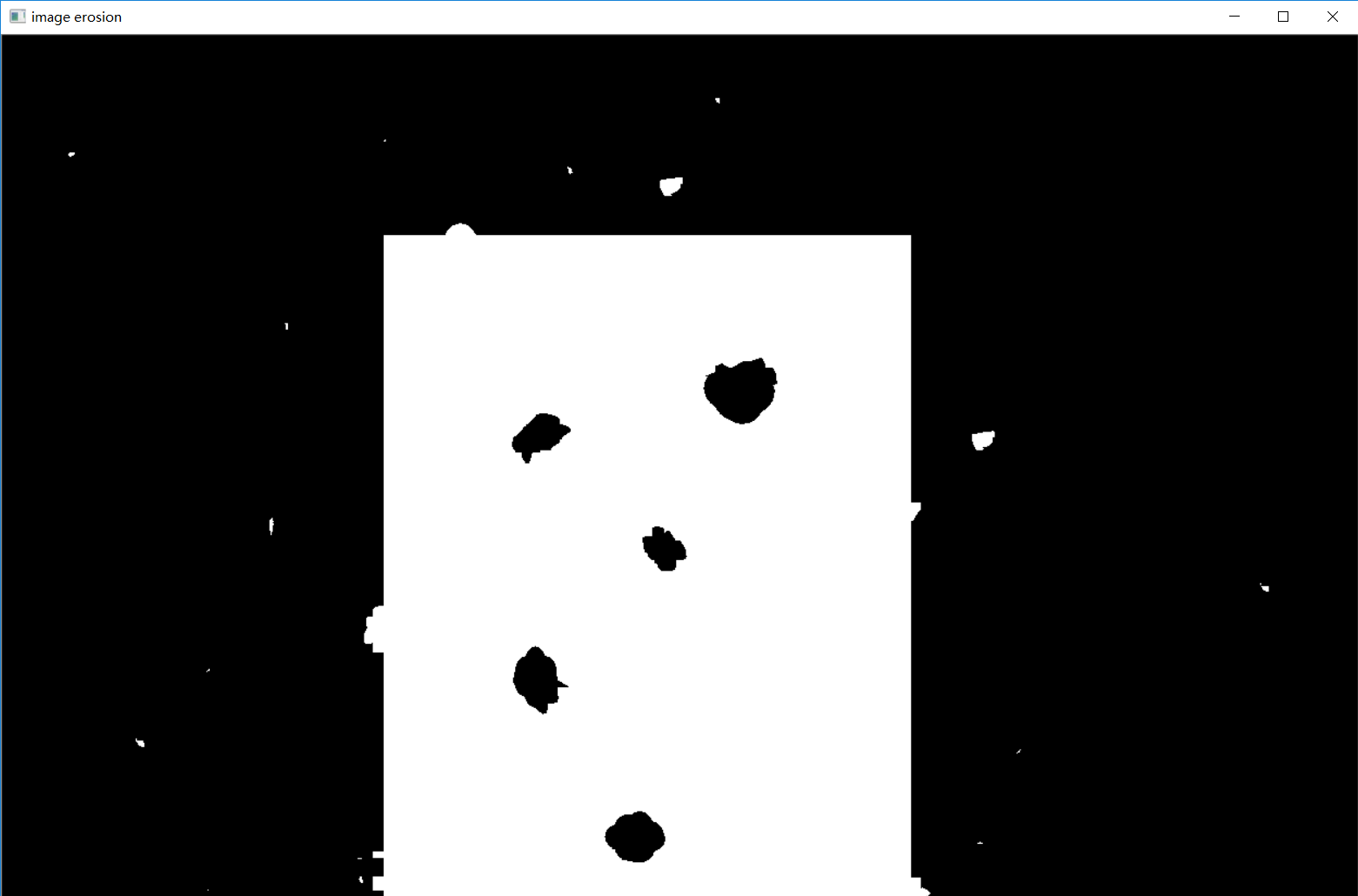
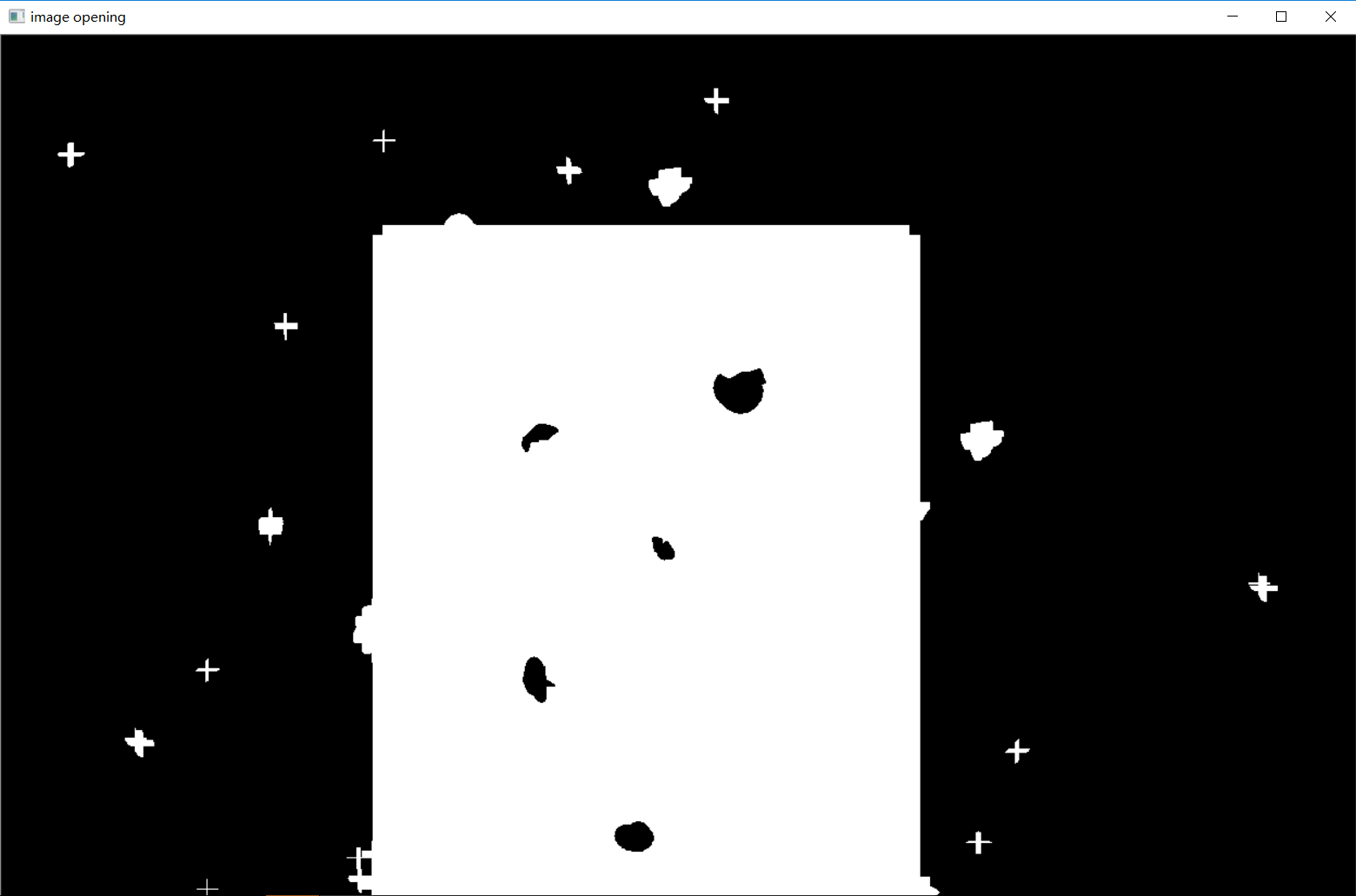
For noise\_rectangle.pgm, we set the size of SE to 20x20 because the noise component size is large. The result is shown in Fig.6 and Fig.7. The order of operations is the same as in noisy\_fingerprint.pgm. We can see that using a rectangular SE can better remove the noise outside the rectangle by doing erosion. However, the imperfections—the dark areas inside the rectangle would be enlarged. Comparing those operations, we can see that using 20x20 rectangular SE for dilations of the opening gave the best result.

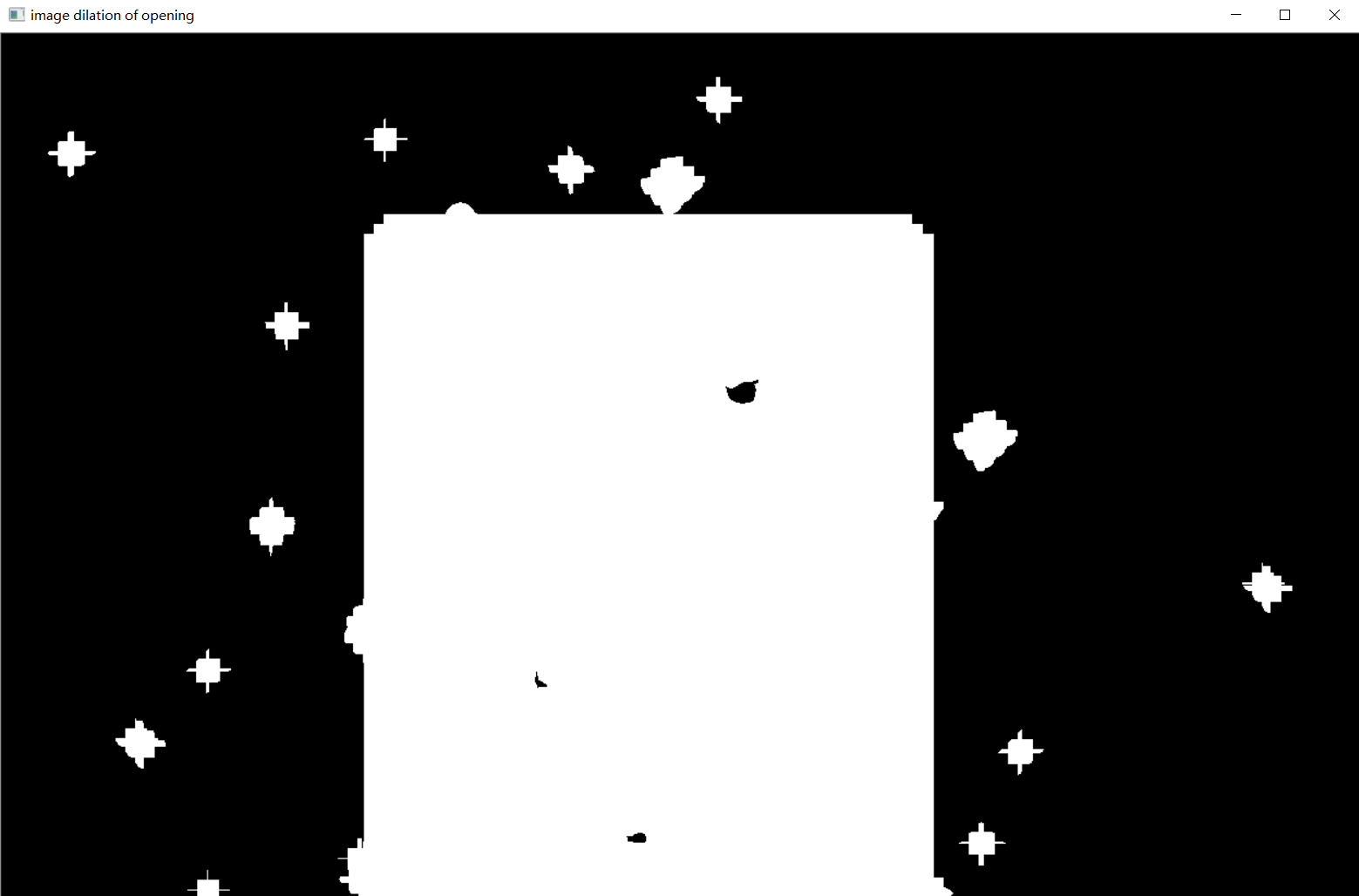
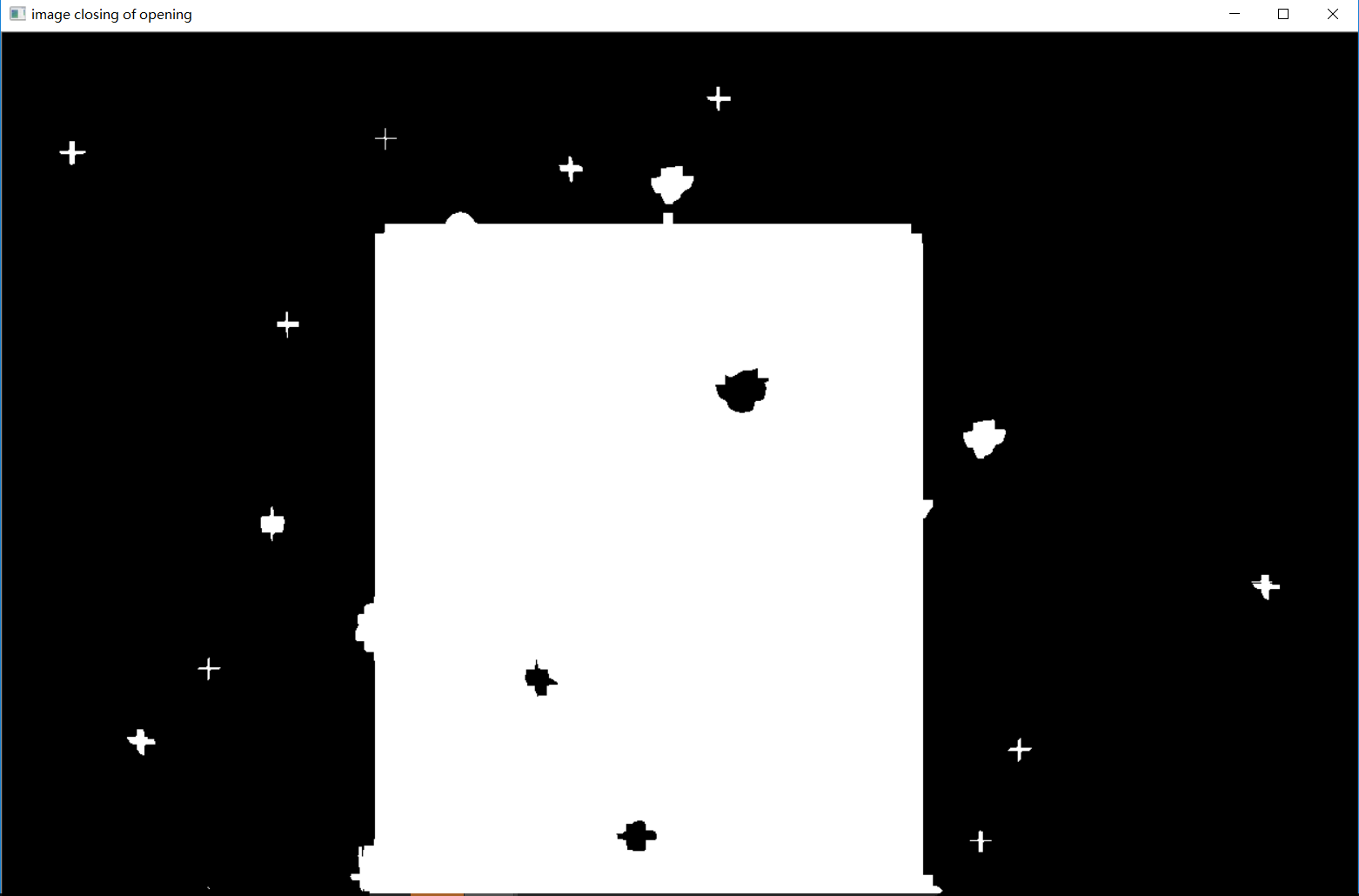
 



**Fig. 6**. Erosion, Opening, Dilation and Closing (20x20 rectangular SE)

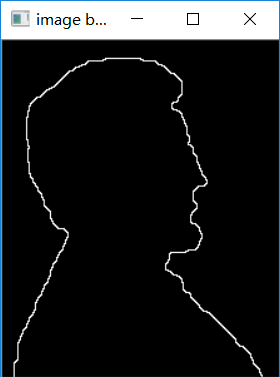
 

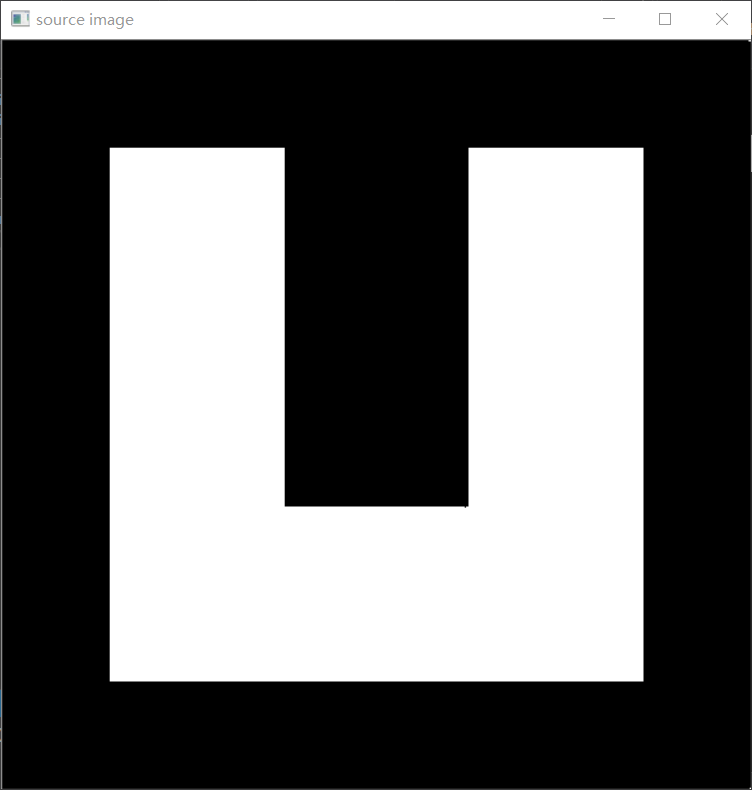
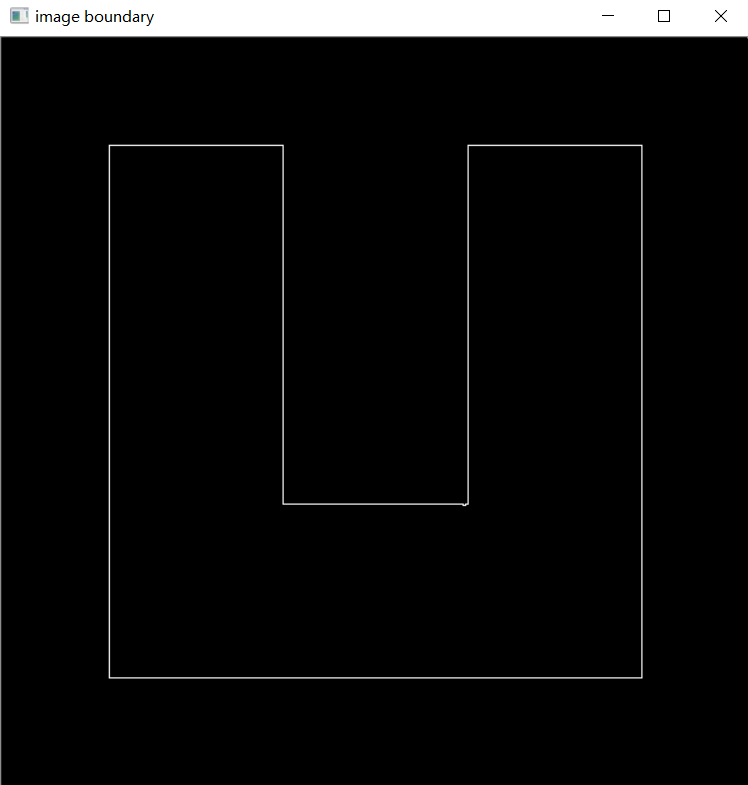
**Fig. 7**. Erosion, Opening, Dilation and Closing (20x20 cross SE)

Now, we are going to implement boundary extraction, which is based on erosion and given by:

 (8)

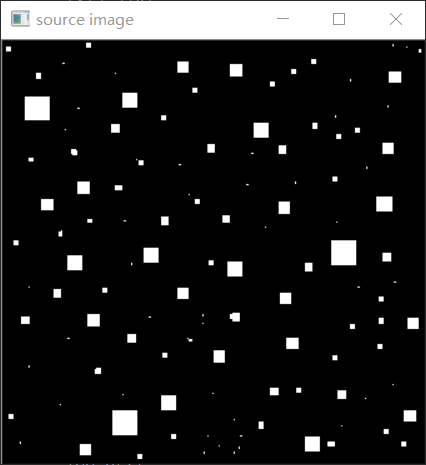
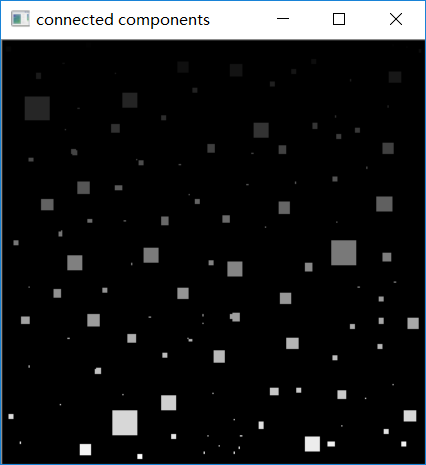
It is not hard to understand this—the eroded version is thinner than the original one and subtract the original one by the eroded image, we can get the boundary. The result is shown in Fig.8, where a 3x3 rectangular SE was applied. We can that such an algorithm has excellent performance in boundary extraction.

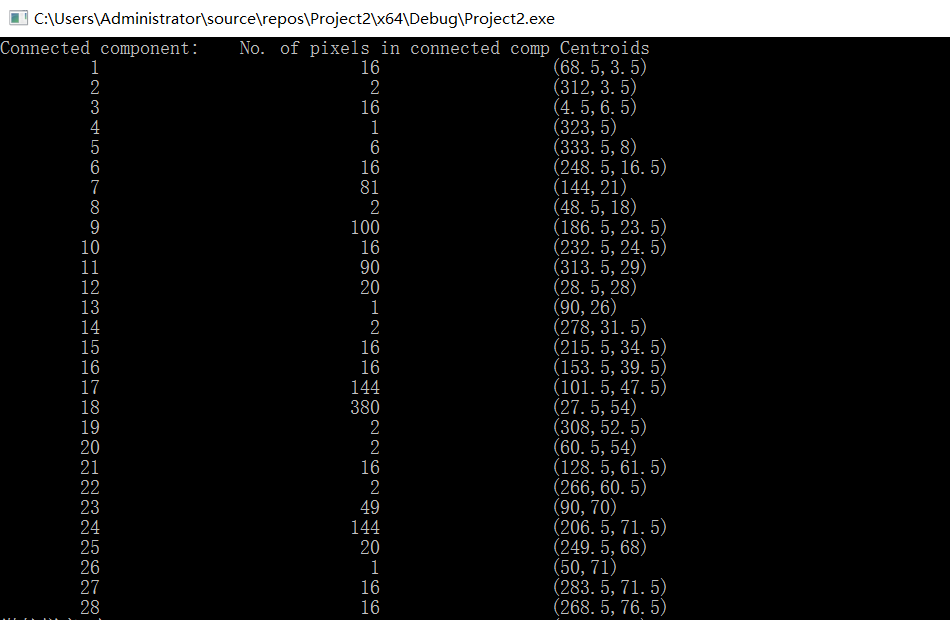
 

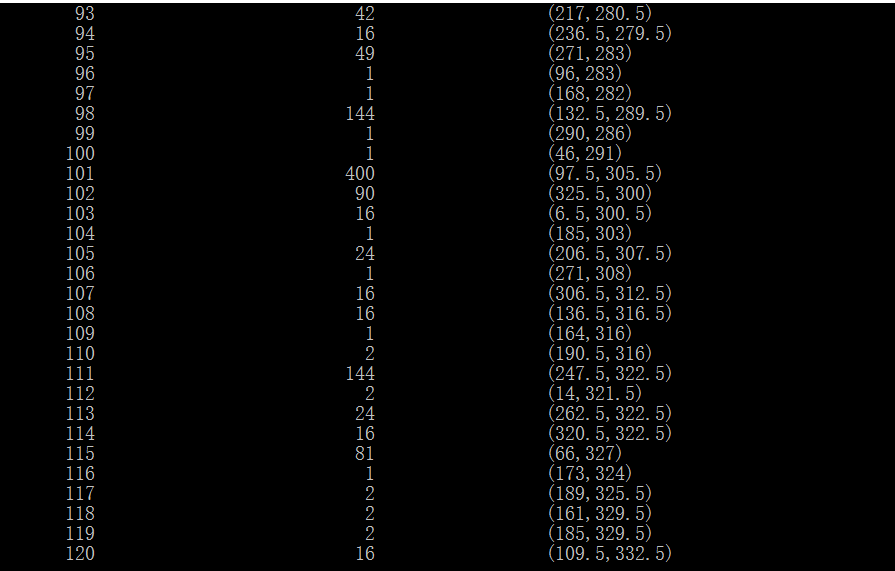
**Fig. 8.** Boundary Extraction

Now, we are going to extract connected components from connected.pgm. That is, to calculate the number of pixels in each connected component as well as given their locations, i.e. center of the connected component. To finish this task, it is actually quite a tough task because the extraction need a starting point from each connected components. Fortunately, there is an OpenCV function called connectedComponentsWithStats, which would count the number of connected components, label them as well as give information about number of pixels and location of centroid. With this function, we can get the resulting labeled image shown in Fig. 9 and print the information of each connected components, shown in Fig. 10. In Fig.9, the connected components are labeled by the function, that is, the first connected component is assigned with intensity value 1, and the second one is 2 and so on. Fig. 10 shows the statistics about the number of pixels in connected components as well as their centroid locations. We can see that there are 120 connected components in connected.pgm.

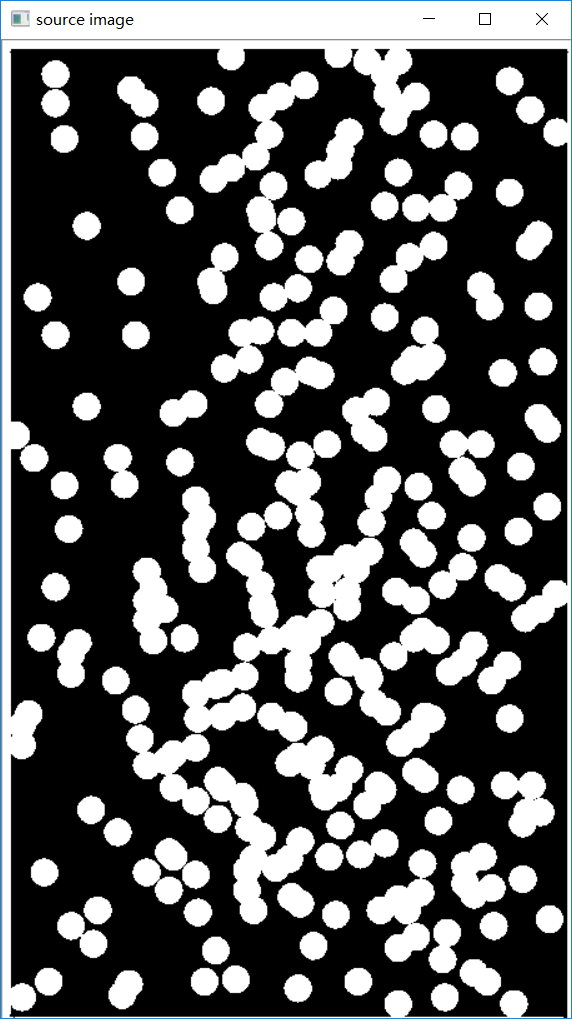
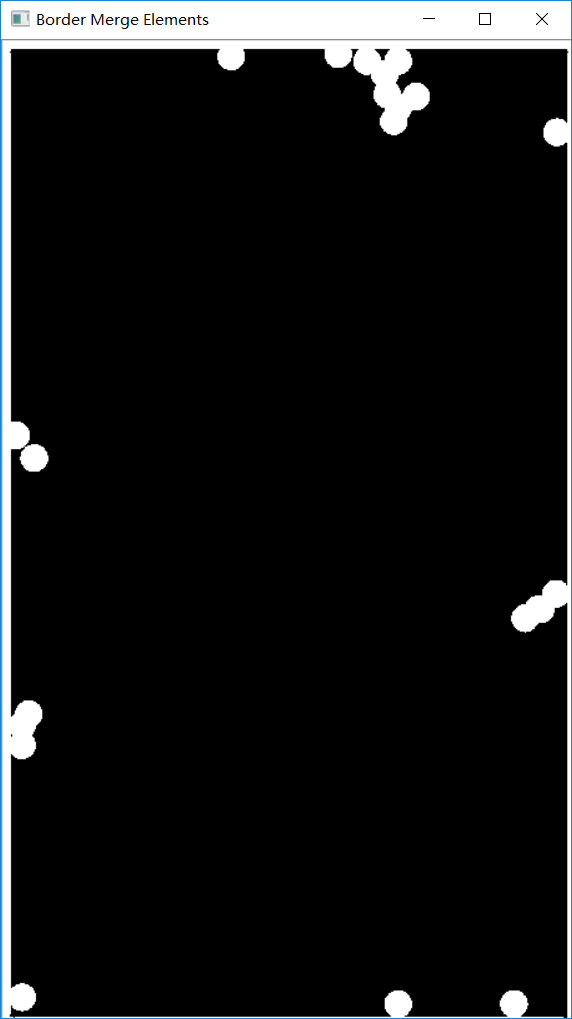
**Fig. 9.** Source Image(L) and Labeled Image(R)





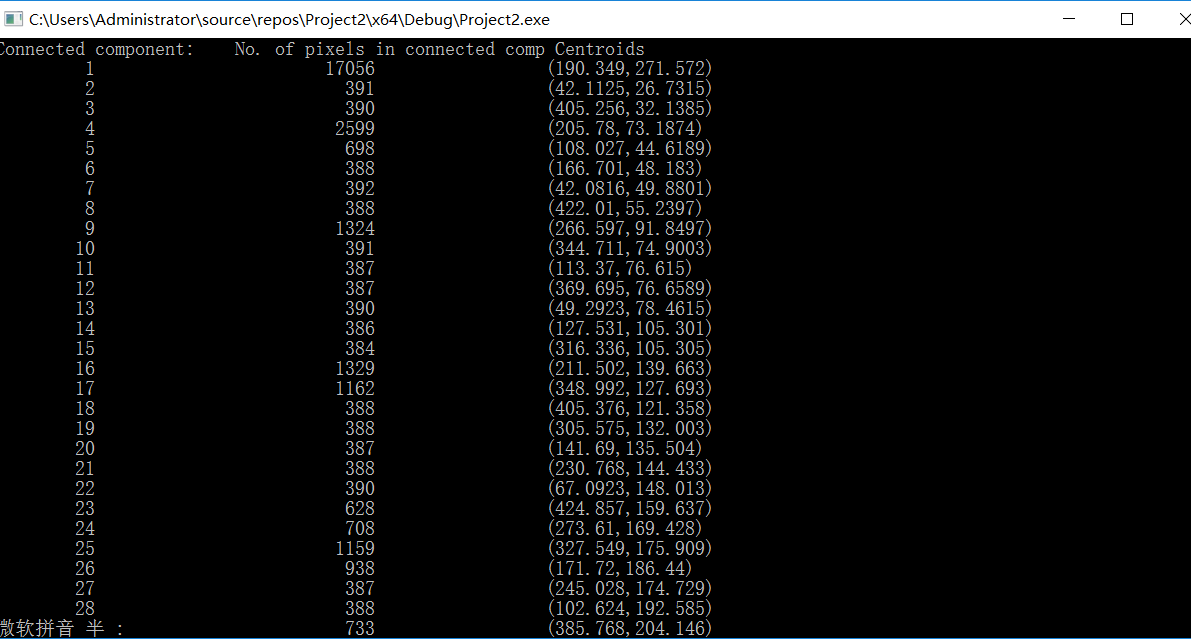
**Fig. 10.** No. of pixels in connected components and centrod locations

The final task in this lab is to separate three kinds of bubbles in bubbles\_on\_black\_background.pgm—bubbles merged with the borders, bubbles non-overlapped and bubbles overlapped (besides ones merged with borders). To extract bubbles with the boundary of the image, we could pad the image with white borders (the same intensity value with bubbles). And then we calculate the connected components using algorithms in task III. The first connected component represents the border. The we can just leave the first connected component in the image using a function called “compare”. The image given has been padded with white borders. However, when we observe it carefully, we can see that the bottom white border is disconnected with other two borders. Therefore, we need to take the last connected component into account as well. Based on the statistic provided by connectedComponentsWithStats, we could know there are 121 connected components totally. Leaving the first and the 121th components, we got the results in Fig. 11. We can see that the merged bubbles were extracted perfectly.

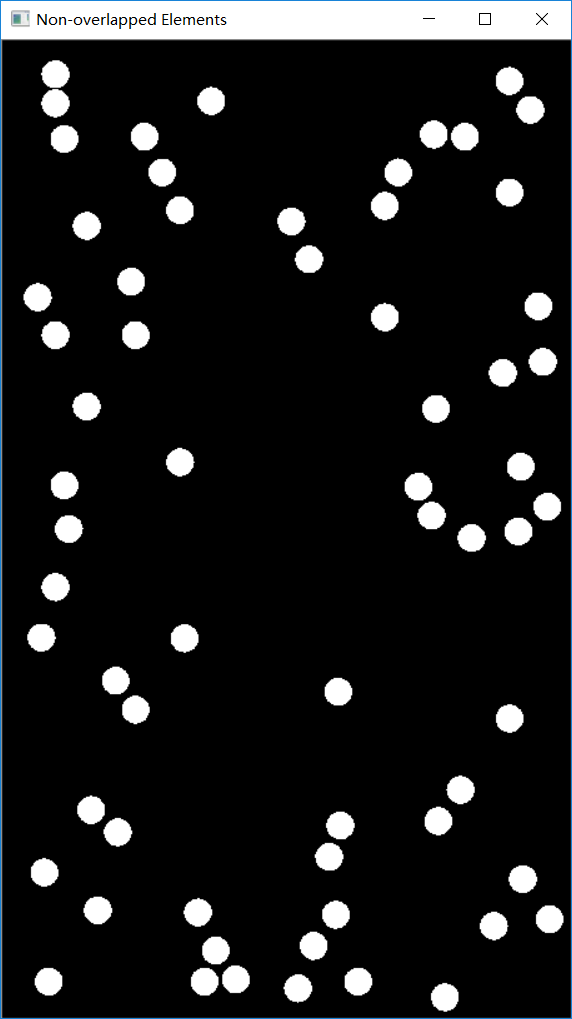
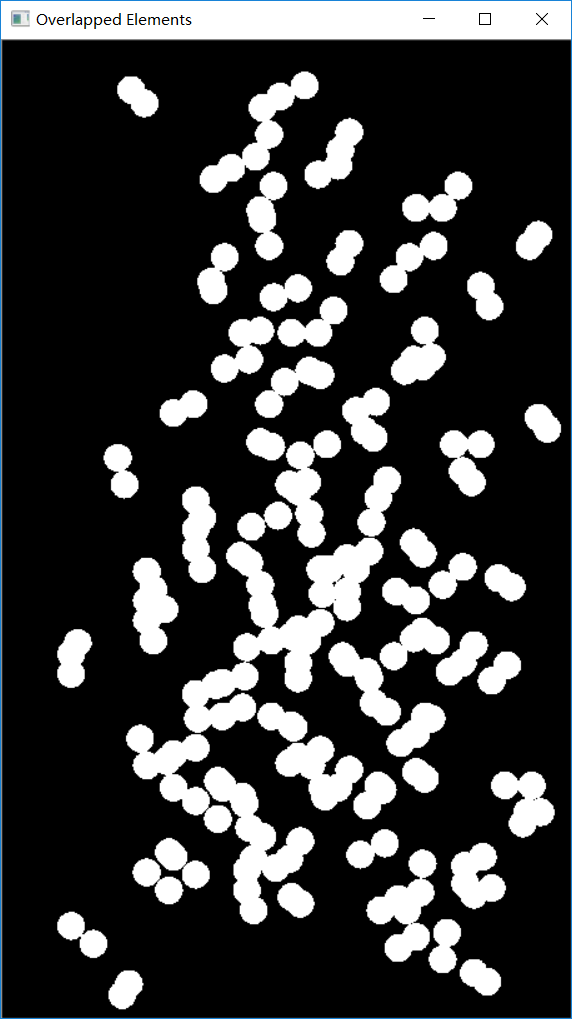
 

**Fig. 11.** Source Image and border-merged bubbles

Then we extracted the non-overlapped bubbles from the image. To do this, we used the connected components again. When we observe the list of number of pixels in connected components in Fig. 12. We can see that in a single bubble is about 387 pixels. And we set a small range to allow some variations of size due to noise. The range is between 382 and 392. The we got the results in Fig. 13, where non-overlapped bubbles were extracted. And subtract from the image single particles and the particles that have merged with the border, and the remaining particles are overlapping particles, which is shown in Fig. 13 as well.



**Fig. 12.** Statistics of bubbles\_on\_black\_background.pgm

**Fig. 13**. Non-overlapped bubbles and overlapped bubbles

# Conclusion

In this lab, we implemented a number of basic morphological techniques, such as erosion, dilation, opening, closing. And combining those techniques, we could finish some practical tasks, such as boundary extraction, connected component extraction. Extracting connected components, we can separate specified parts of the image.

# Reference

[1] *Digital Image Processing* Gonzalez 4th edition

[2] https://stackoverflow.com/questions/29108270/opencv-2-4-10-bwlabel-connected-components/30265609#30265609

1. Chengpin Luo is with Electrical and Electronic Engineering Department, Southern University of Science and Technology, 518055, CHINA (Student Number: 11510880, e-mail: 11510880@mail.sustc.edu.cn). [↑](#footnote-ref-1)