谭树杰 11849060

# Lab11 Report

1. **Use Otus’s method on original image**

**large\_septagon\_gaussian\_noise\_mean\_0\_std\_50\_added.pgm and its 5x5**

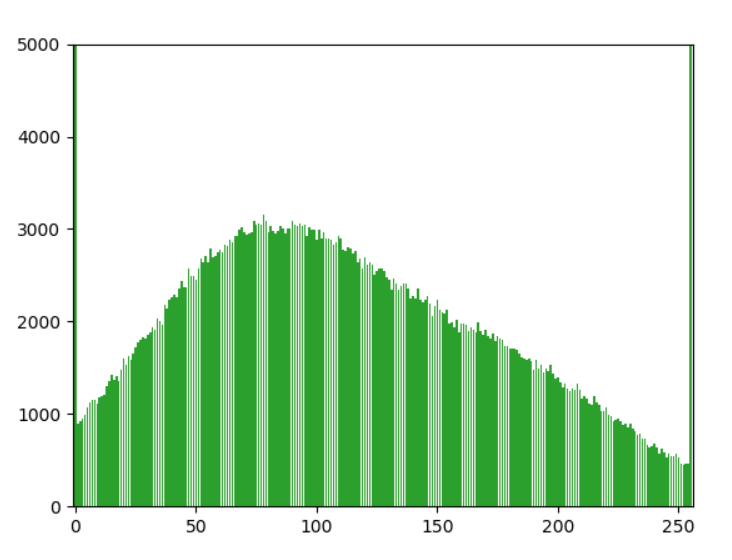
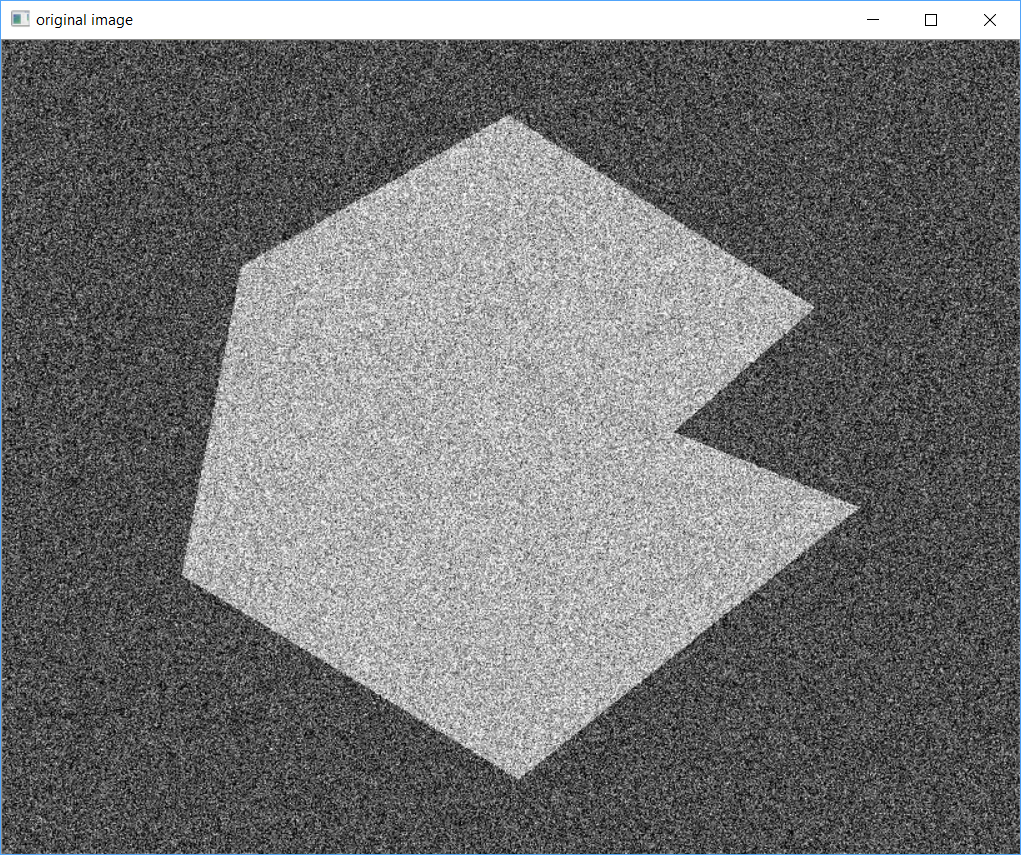
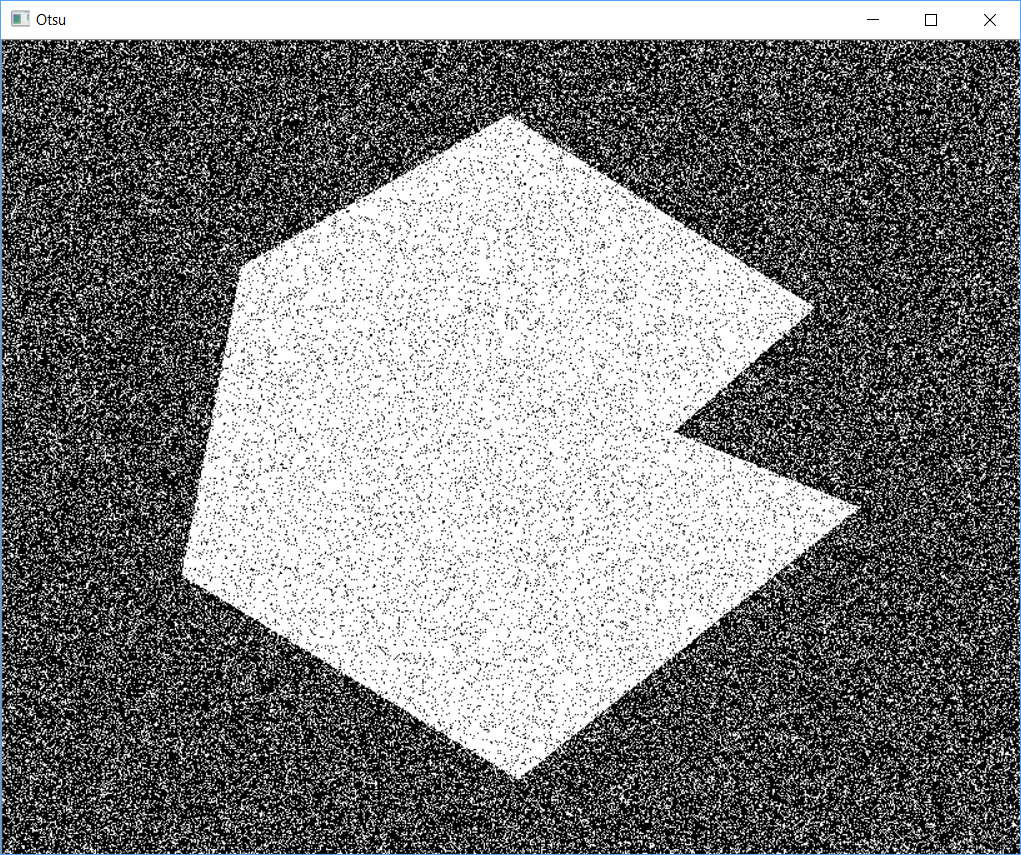
**smoothed image to perform segmentation to output binary images.**

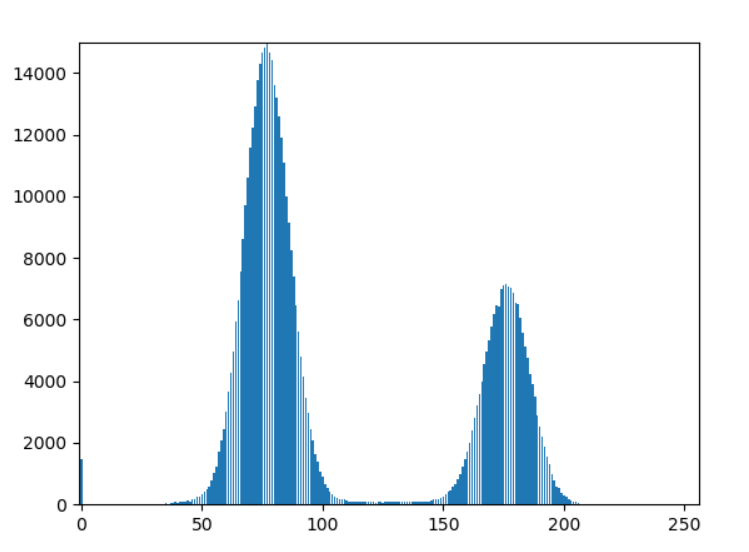
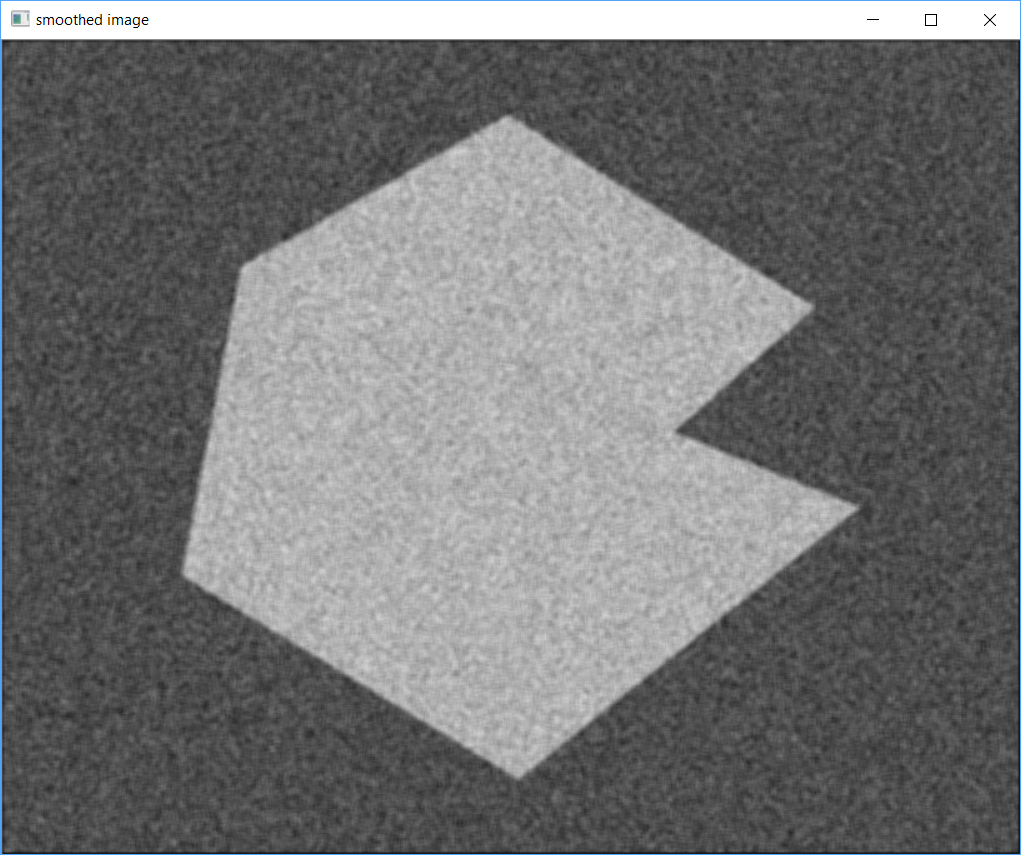
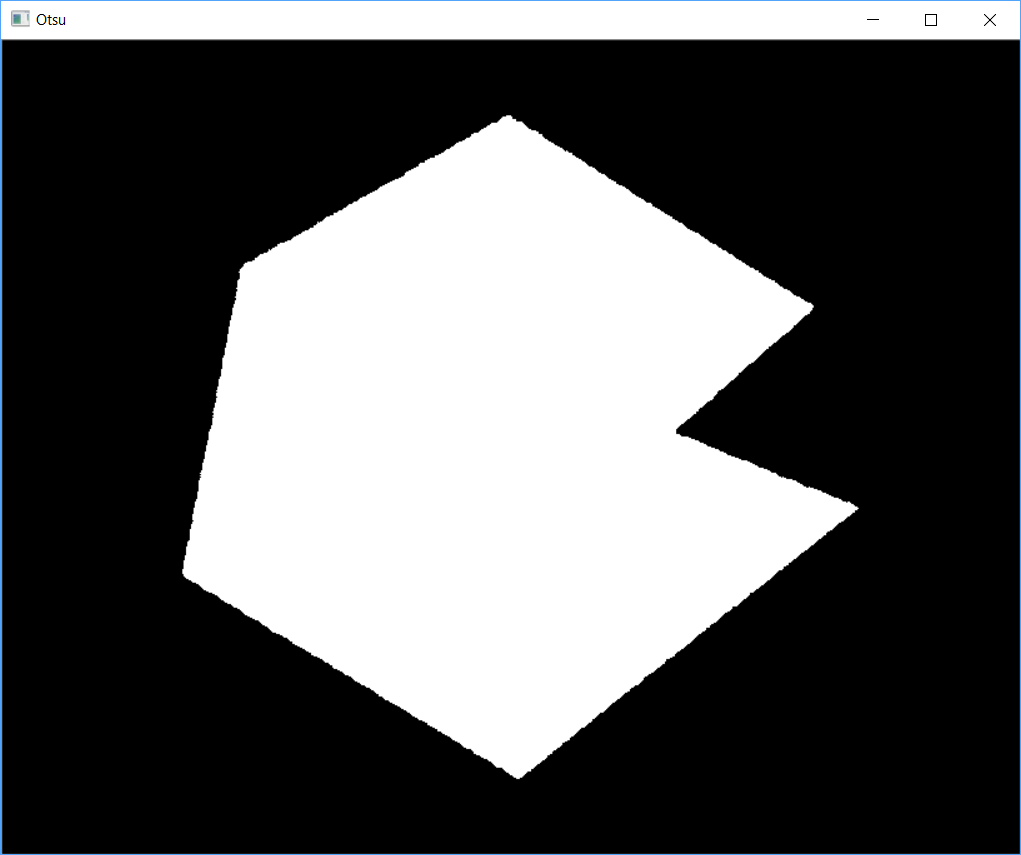
**Results:**

1. The first row below are the original image and its histogram and Result obtained using Otsu’s method respectively. (threshold = 108)

The second row below are the smoothed image, its histogram and Result obtained using Otsu’s method respectively. (threshold = 125)

The smooth mask is 5 \* 5 averaging

**Analysis:**

Every black point in the white region and every white point in the black region is a thresholding error, so the segmentation was highly unsuccessful.

To solve this problem, we smooth the noisy image with an averaging mask of size 5x5. The improvement in the shape of the histogram due to smoothing is evident, and we would expect thresholding of the smoothed image to be nearly perfect. As final picture shows, this indeed was the case.The slight distortion of the boundary between object and background in the segmented, smoothed image was caused by the blurring of the boundary. In fact, the more aggressively we smooth an image, the more boundary errors we should anticipate in the segmented result.

**The implementation code is shown below:**



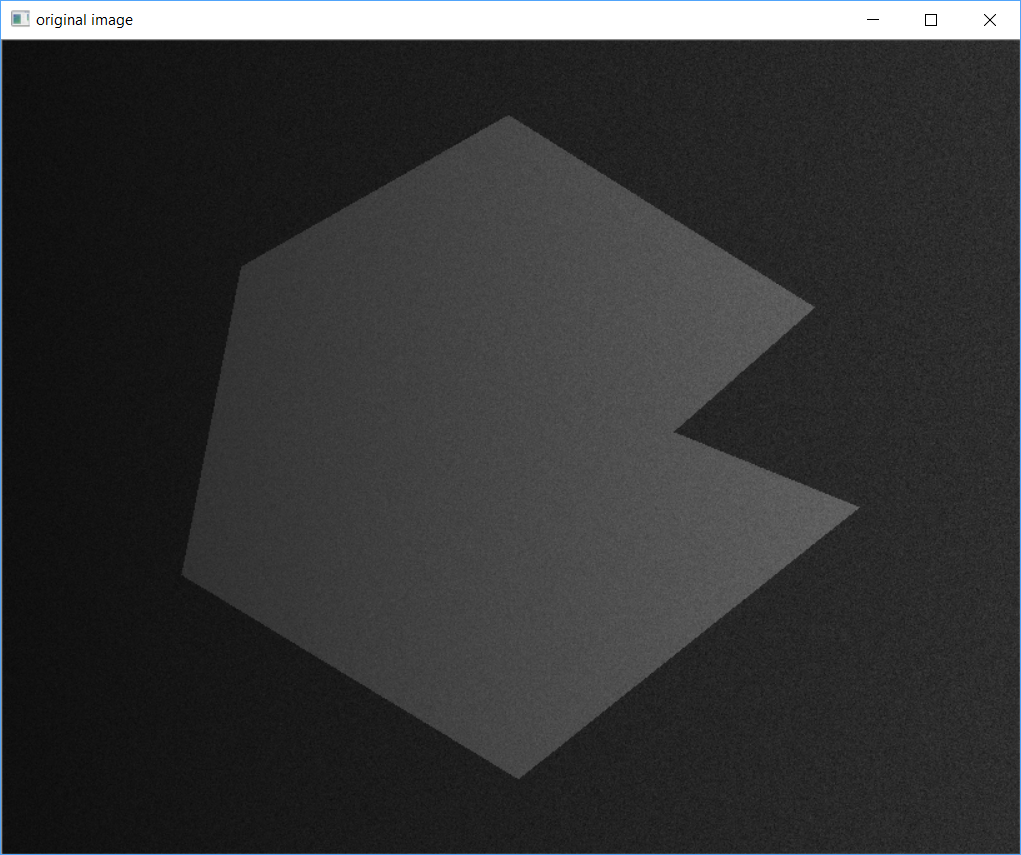
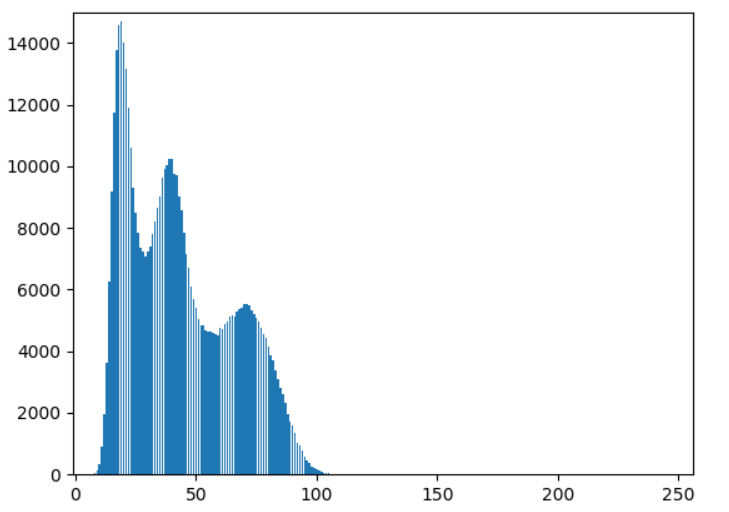
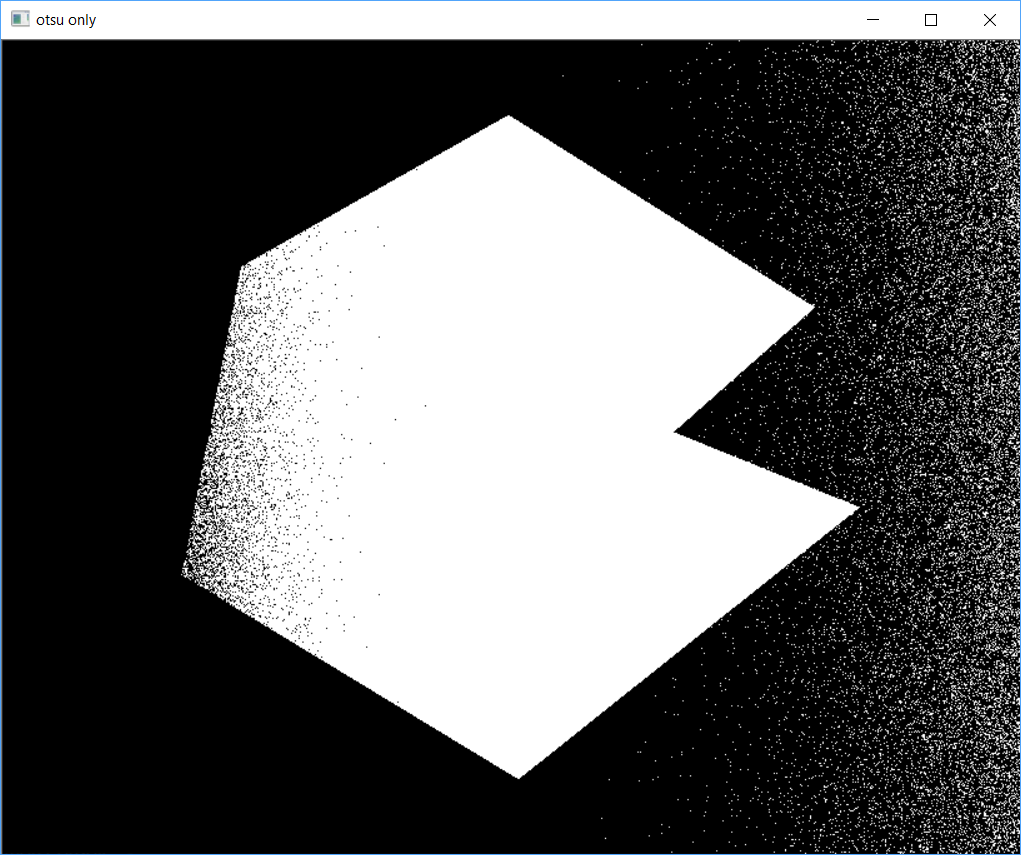


**2. Partition method first and then Otsu’s method to segment**

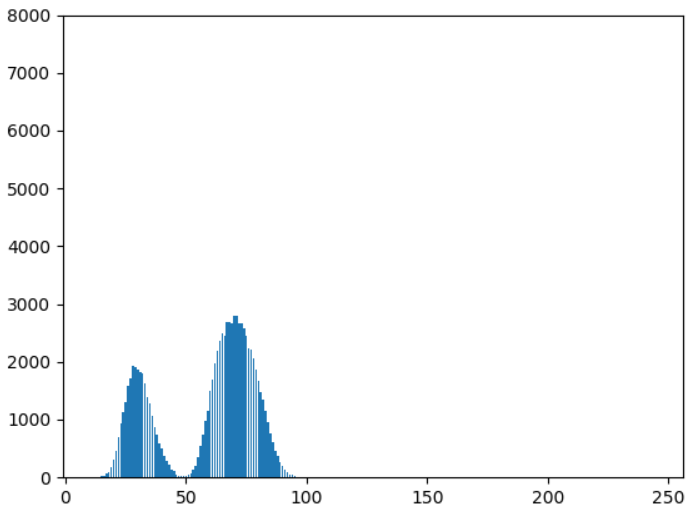
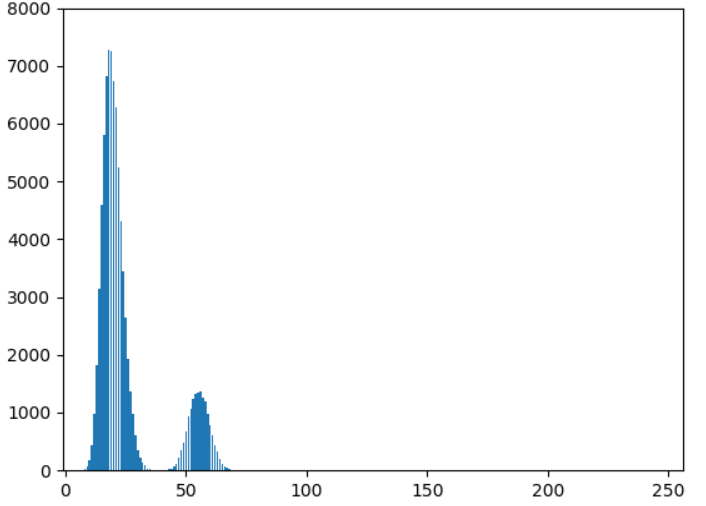
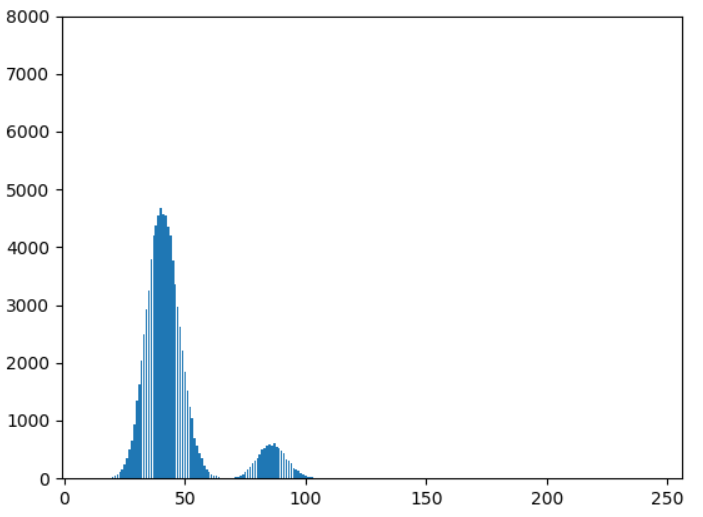
**septagon\_noisy\_shaded.pgm**.

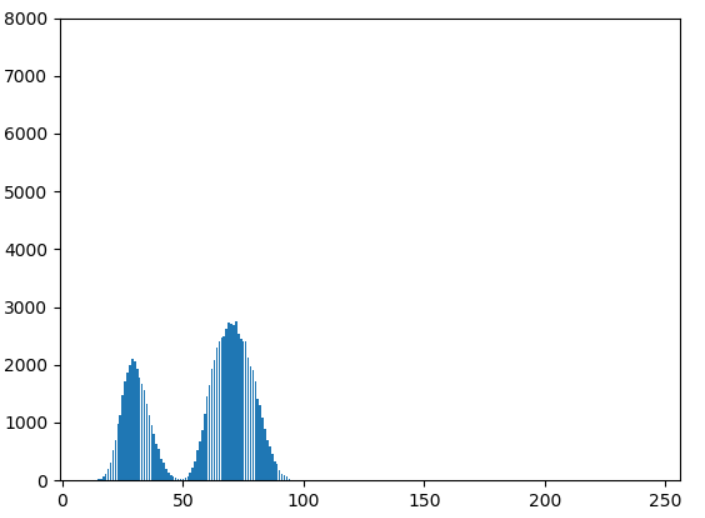
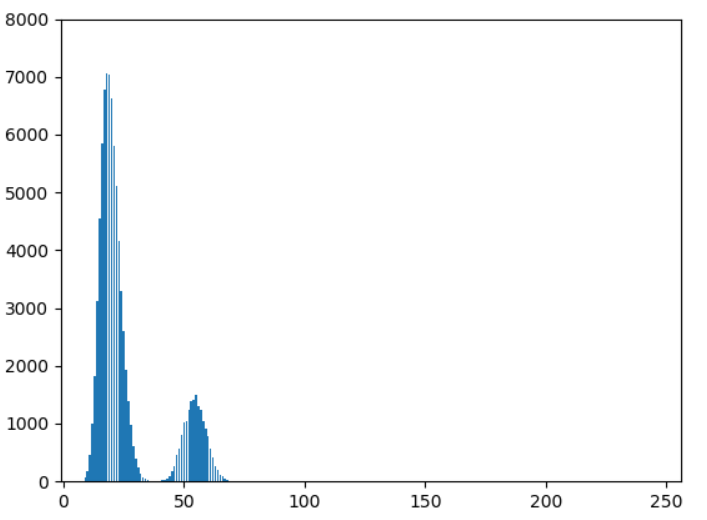
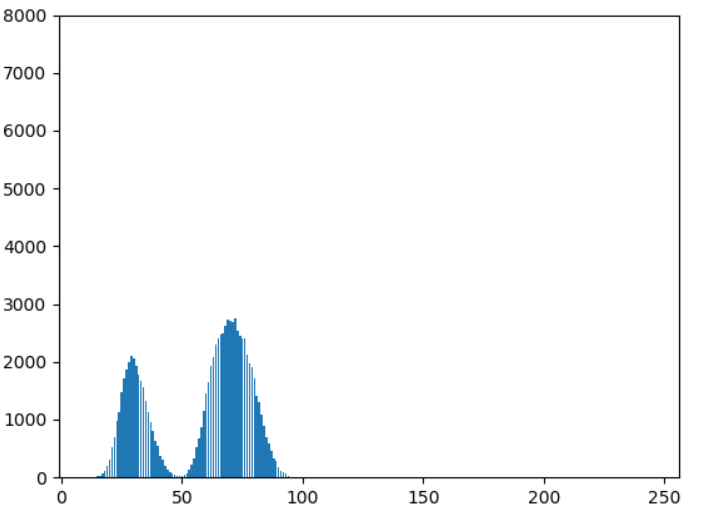
**Results:**

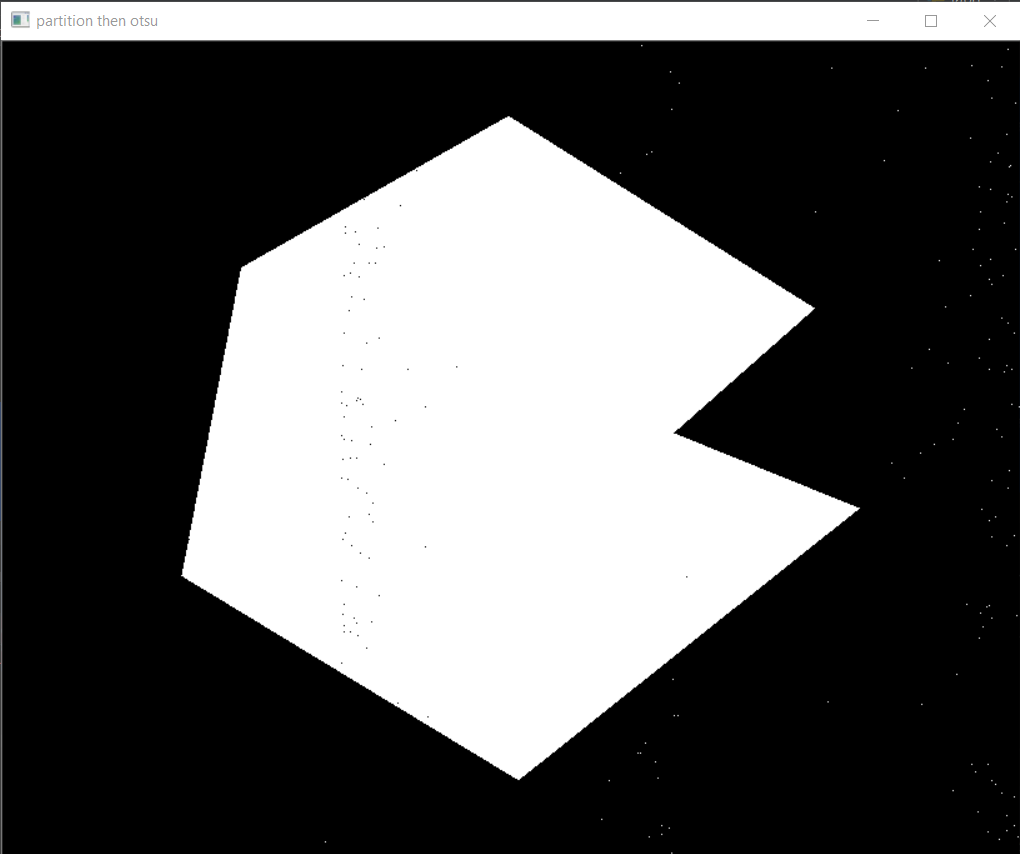
1. The septagon\_noisy\_shaded.pgm, its histogram and result image by directly applying Otsu’s algorithm are shown below.

1. The histogram of six parts of the original image and the result image by partitioning first then applying Otsu’s algorithm are shown below.



**Analysis:**

We can observe that the original image has three peak values, therefore the Otsu’s method can not be used directly, as we can see from the result image.

We can observe that every subimage have only one distinct valley, hence we can use Otsu’s method on each subimage separately, then assemble these subimages to get the final result image. We can observe the improvement clearly.

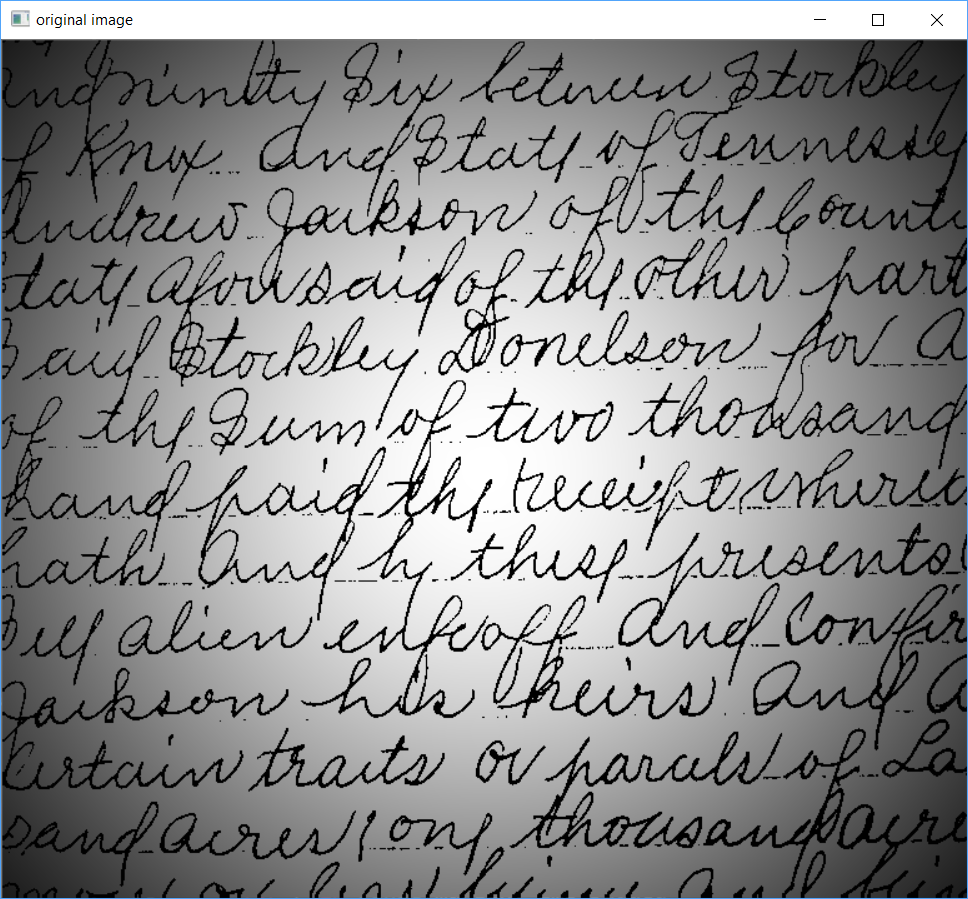
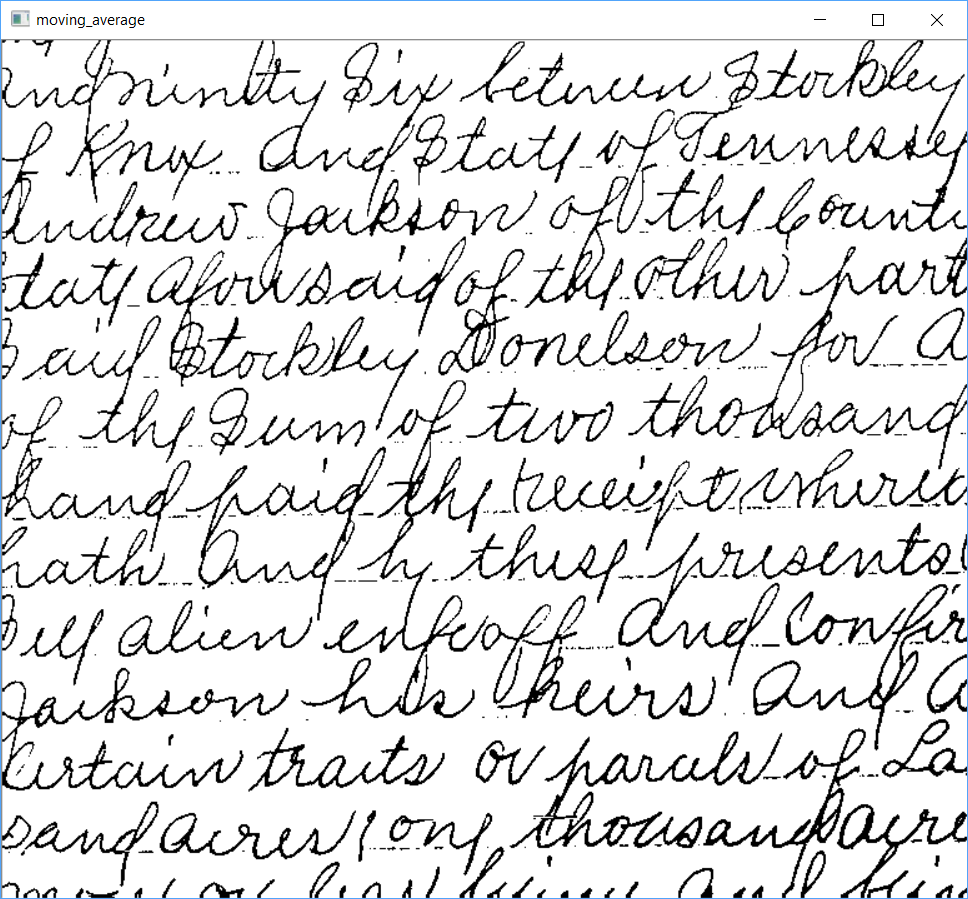
**The implementation code is shown below:**

****

**3. Use moving average thresholding to segment spot\_shaded\_text\_image.pgm**

**Result:**

The spot\_shaded\_text\_image.pgm and thresholded image are shown below.

**Analysis:**

Because using the Otsu global thresholding method could not overcome the intensity variation. We choose local thresholding using moving averages. We can see that the handwriting characters of the image is more legible after processing. In general, thresholding based on moving averages works well when the objects of interest are small (or thin) with respect to the image size, a condition satisfied by images of typed or handwritten text.

**The implementation code is shown below:**

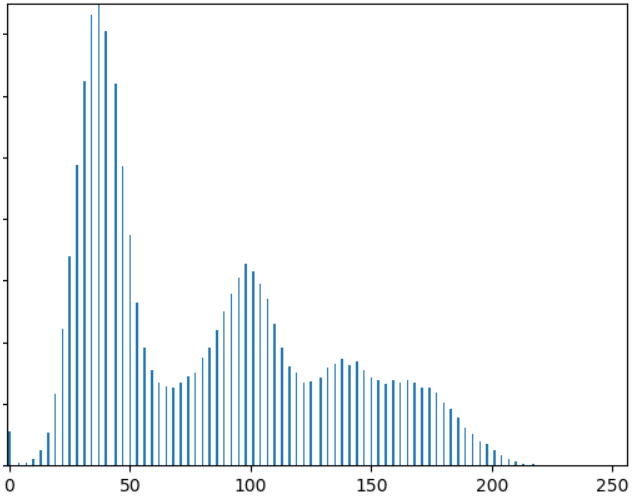
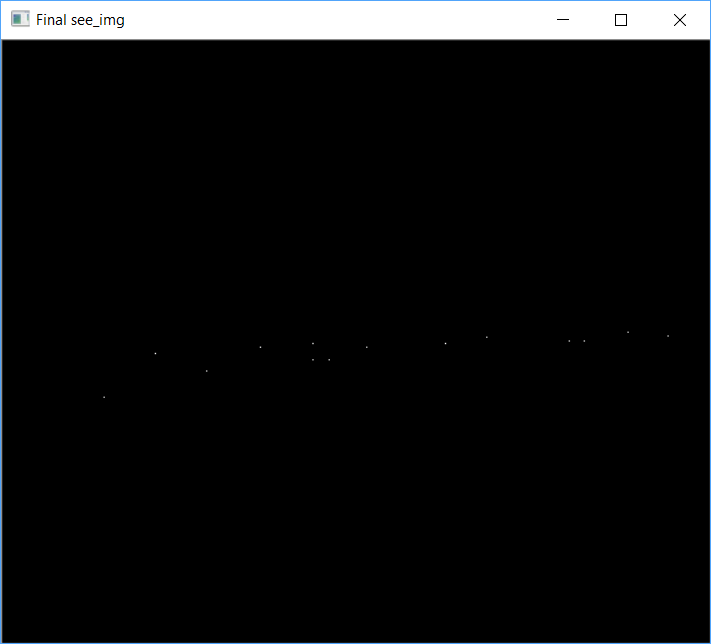
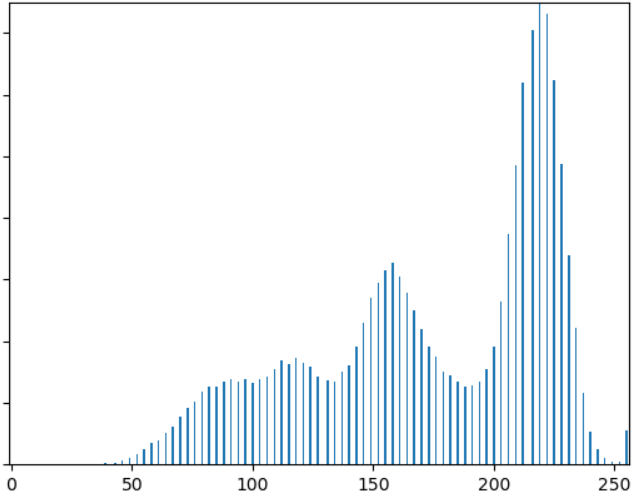
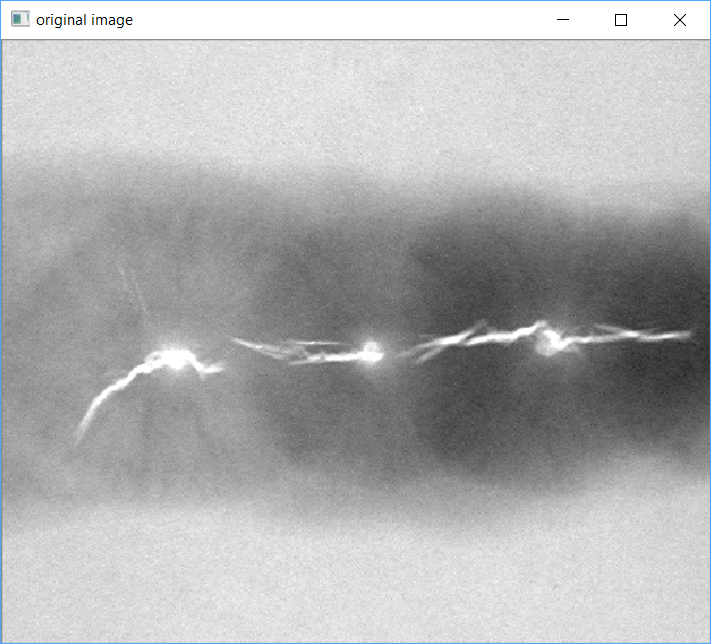
****

**4. Use region growing method to perform segmentation on**

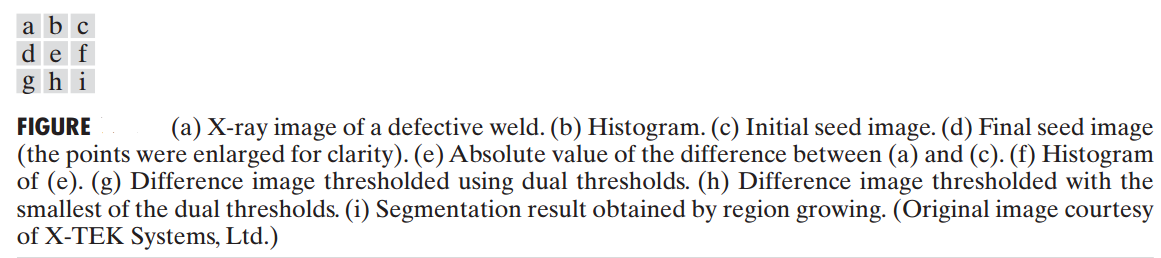
**defective\_weld.pgm and noisy\_region.pgm**

**Result:**

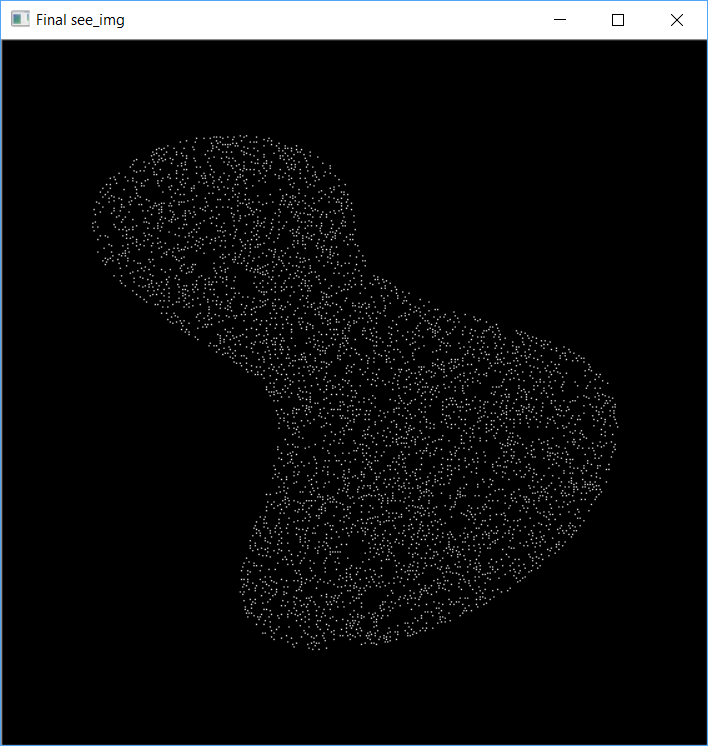
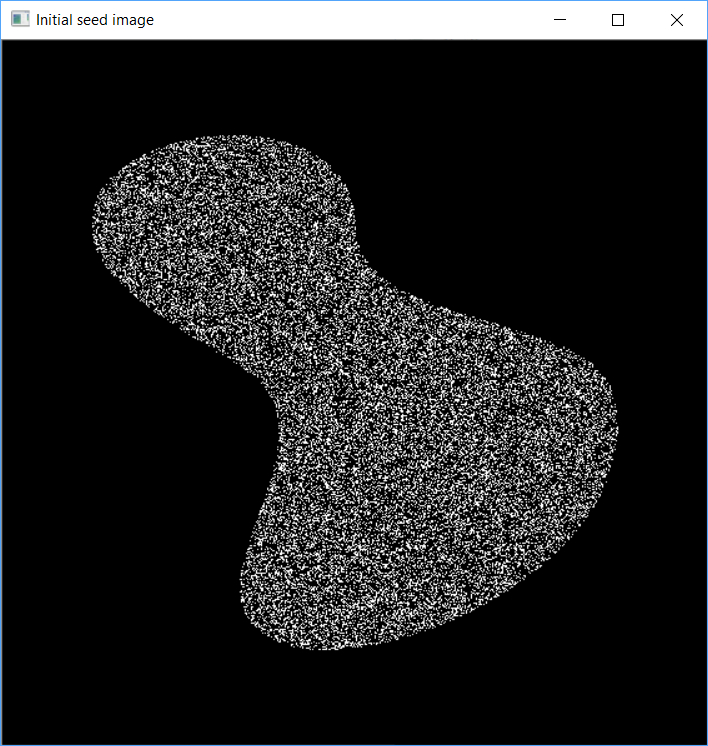
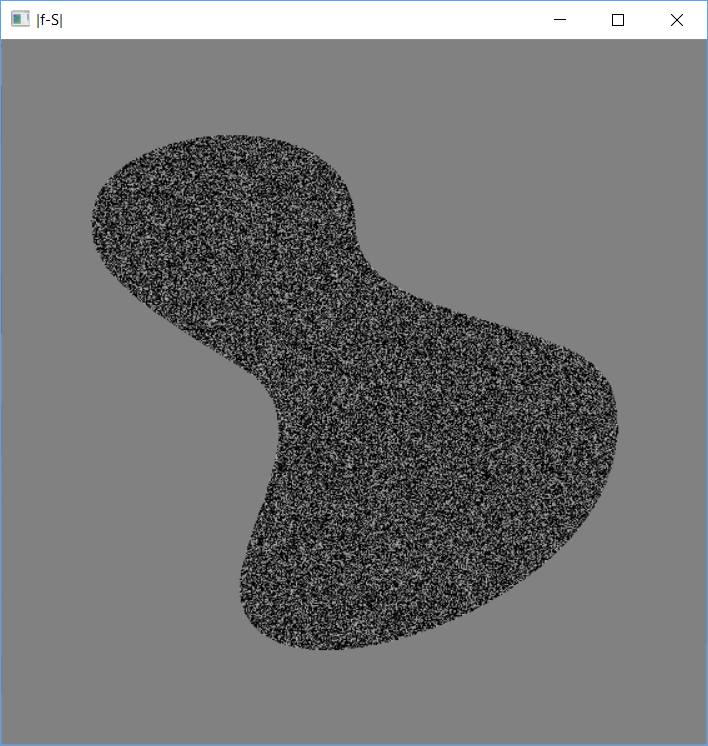
(1)The defective\_weld.pgm and related images(as in textbook) are shown below.







（2）The noisy\_region.pgm and related images as previous are shown below.

**Analysis:**

We illustrate the use of region growing by segmenting the defective weld regions. These regions could be used in applications such as weld inspection, for inclusion in a database of historical studies, or for controlling an automated welding system. The result in Fig. (g) shows that the problem of segmenting the defects cannot be solved using dual thresholds, even though the thresholds are in the main valleys.Fig(i) shows, this step resulted in the correct segmentation, indicating that the use of connectivity was a fundamental requirement in this case.

**The implementation code is shown below:**

****

****

****

****