CSE585/EE555: Digital Image Processing II

Computer Project # 1:

Mathematical Morphology: Hit or Miss Transform

Mudit Garg, Mayank Murali and Niranjan Thirusangu

Date: 01/31/2020

A. Objective:

This project aims to implement various mathematical morphological operations like erosion, dilation, open, close and hit-or-miss transform to extract specific sizes of disk on the image 'RandomDisks-P10.jpg'. It also helps in understanding how noise affects the hit-or-miss transform.

B. Methods:

The process flow followed for the project implementation is as follows:

- Convert the given grayscale formatted image 'RandomDisks-P10.jpg' into a binary image.
- Remove noise from the given image using open and close operation.
- Apply Hit-or-Miss transformation using disk shaped structuring element of various radii.
- Visualize the smallest and the largest disks' locations by selecting only the disks' whose locations are acquired by the Hit-or-Miss transformation.

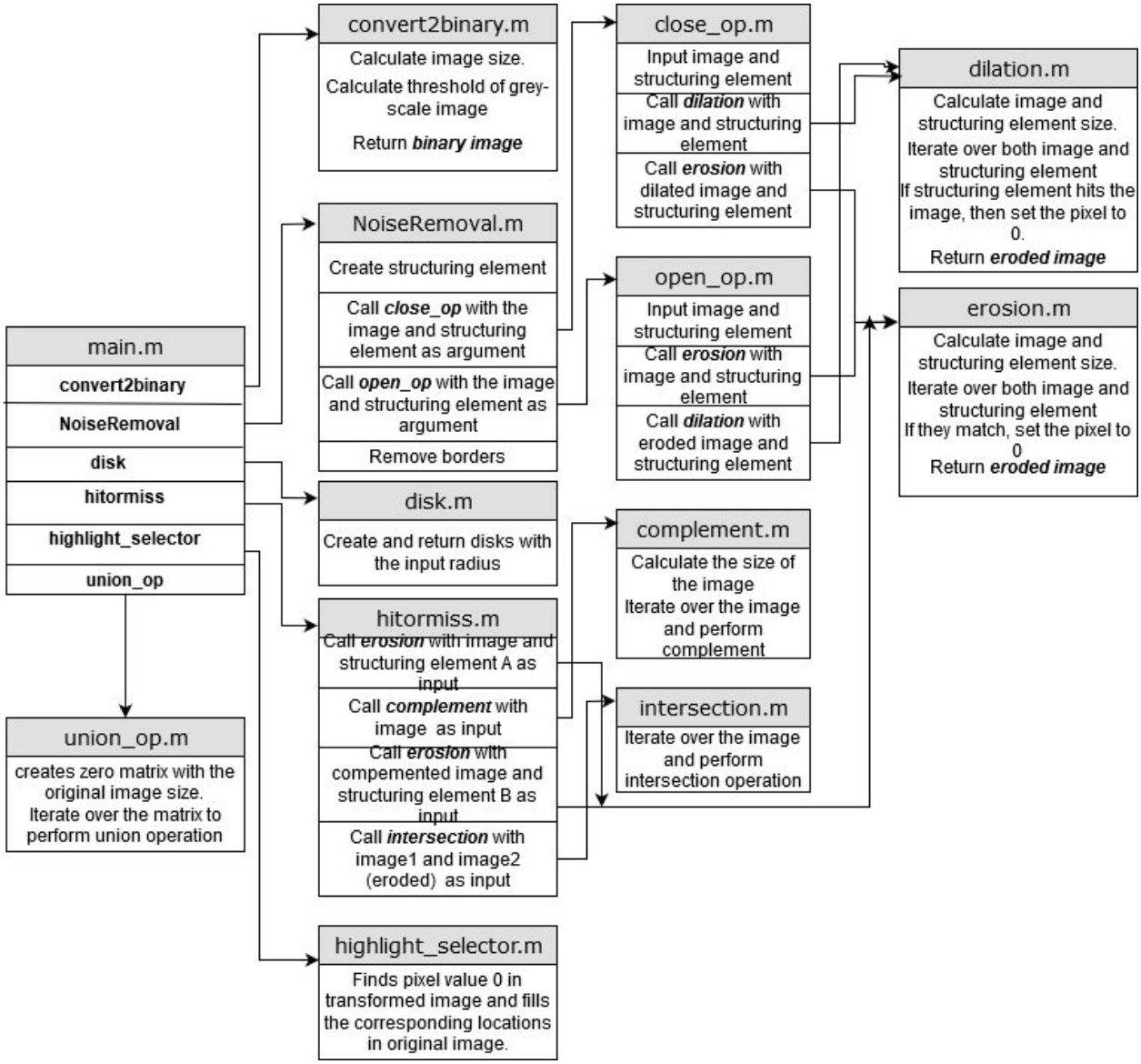

Fig1: Flowchart for the whole transformation with MATLAB files and their respective function calls

Conversion to binary

The conversion of the given image to binary is processed by implementing the convert2binary(img) function. This function thresholds all the pixels whose values are above 127 value(mid value of the 8-bit range) in the grayscale as 1 (white) and others as 0 (black).

Close and Open Filter

In order to remove the salt and pepper noises in the image, we performed close and open operations on the image. As it is evident that the sizes of the noises are only one pixel large, we went with a 3x3 as the structuring element for the close and open operations. These operations were performed as per the formula mentioned in L4-11.

We first did the close operation to fill the details (salt noise in disks) followed by open operation to remove the details smaller than structuring element (remove the pepper noise in the background).

The below equation is of open operation. It first does erosion of the image with a structuring element followed by dilation with same structuring element :

$$X_B = X \circ B = (X \ominus B) \oplus B$$

In close operation, we do the dilation first followed by erosion with the same structuring element. The equation of the close operation is given as :

$$X^B = X \bullet B = (X \oplus B) \ominus B$$

Since the structuring element is centered around origin, $B = B^S$.

Hit-or-Miss Transformation

The Hit-or-Miss Transformation is performed to extract the locations of the disks of radii with specific range. Ideally by the end of the transformation operation, the resultant image will contain the center of all the disks selected. The Hit-or-Miss transformation is given by the mathematical morphological equation as per the L4-8 as,

$$X \circledast (A, B) = (X \ominus A^S)/(X \oplus B^S) = (X \ominus A^S) \cap (X^c \ominus B^S)$$

The range of radii to select the disks can be providing different radius for the A structuring element and the B structuring element. Structuring element B is a window with a hole in between. For B, we mean that there is hole of radius which we mentioned in a window of size $2 \times radius + 2$. As we were able to find the radius of the small disks as 8 pixels through Matlab's interactive window, we took a disk with a radius 8 pixels for A and 10 pixels for B, for selecting the smallest disks in the image. And for the

largest disks whose radius is 32 pixels, we took the structuring elements as disks with 30 pixels radius for A and 37 as radius for B ()

Visualization using Flood Fill

After the Hit-or-Miss transformation we will get only the center of the disks as the result, which in most cases only 1 pixel. As 1 pixel is so tiny to view on a computer screen, we wanted to show the disks whose locations are extracted using Hit-or-Miss transformation. To do that, we filled the “filtered_img.gif” with the locations we have extracted from the Hit-or-Miss transformation operation. We did that using inbuilt flood-fill function in Matlab, called imfill() as that is purely for visualization and out of the scope of the project.

Union of images

In order to obtain an image containing only largest and smallest disk, we did the union of both the images (small disks and large disks). After this operation, we got our final image which is shown in the results section.

Note: In order to reproduce the results that are obtained, just run the `main.m` file. It will take care of structuring elements and save the images in your machine that are shown under results section.

C. Results:

Below are the images obtained after performing various mathematical morphological operations and transformation to the original image `'RandomDisks-P10.jpg'`.

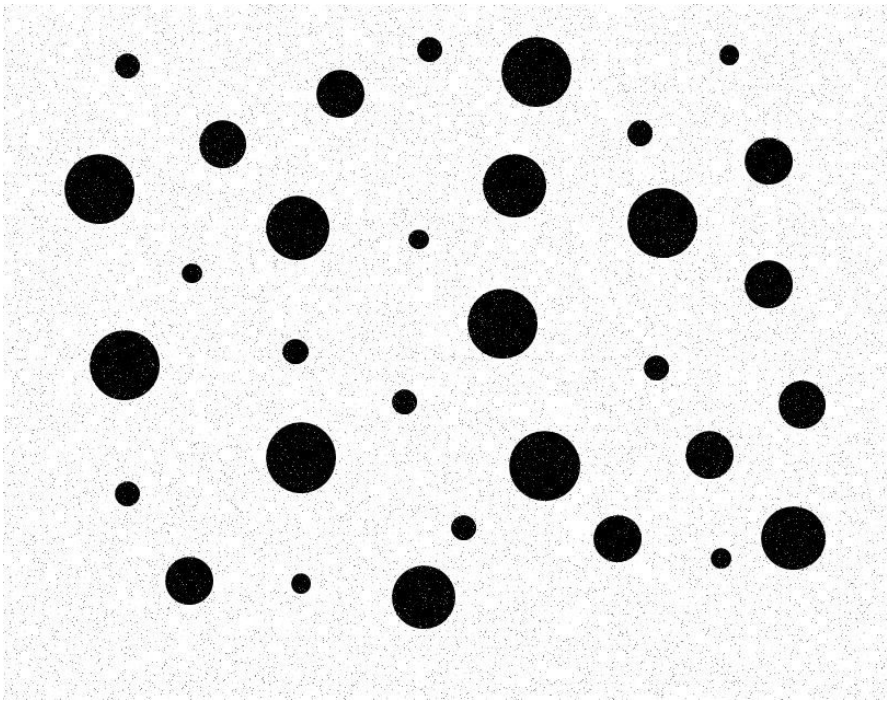

Fig2: Original Random Disks Image 'RandomDisks-P10.jpg' itself.

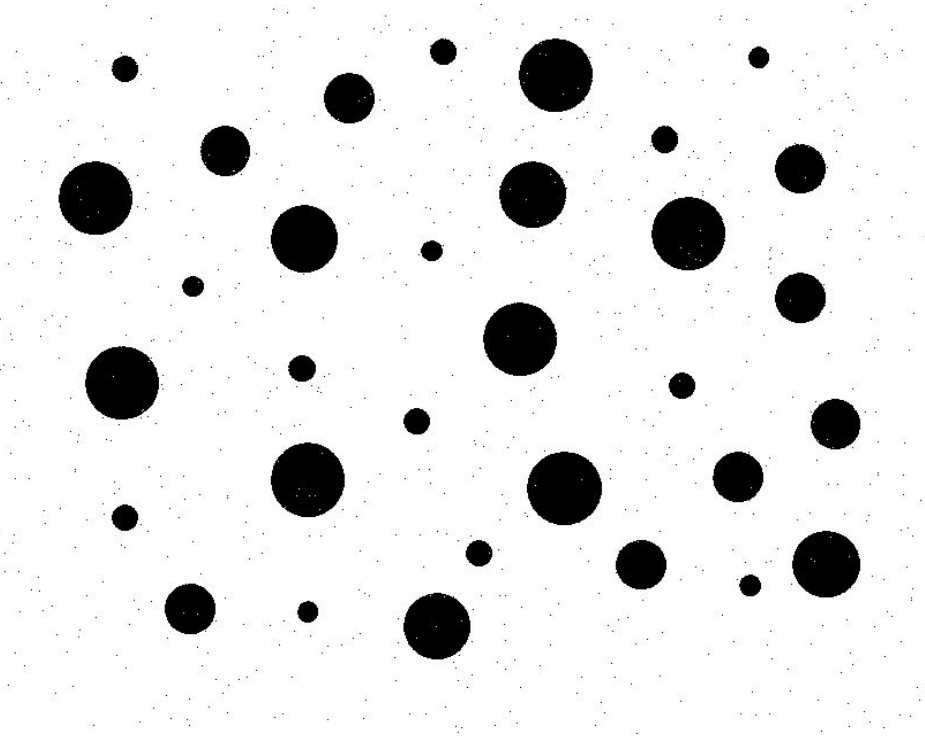

Fig3: Binary Image (after applying threshold to Fig2)

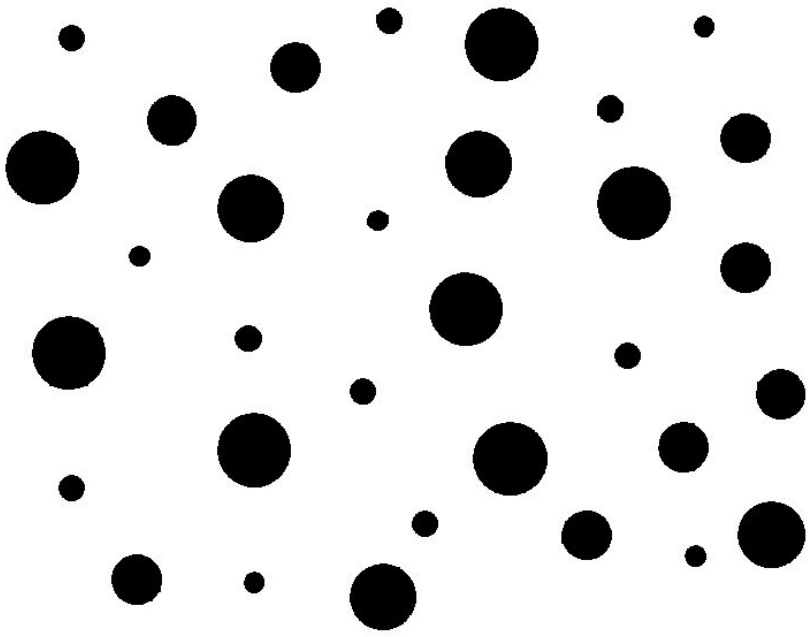

Fig4: Image after removing noise using close and open operations

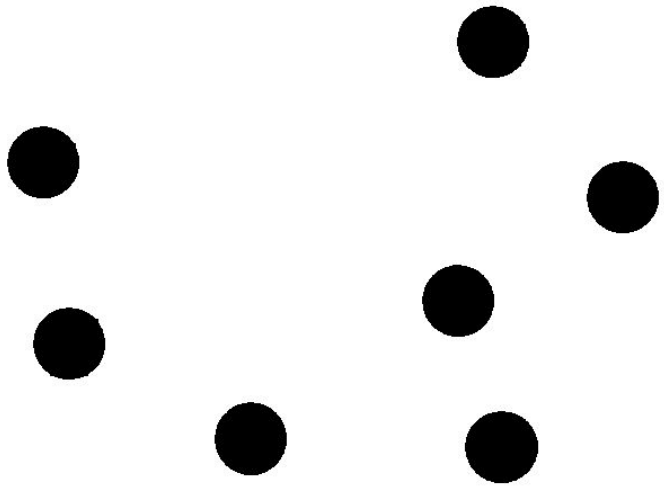

Fig5: Image with the largest disks. They are attained after using structuring element of disk shape of radius 30.

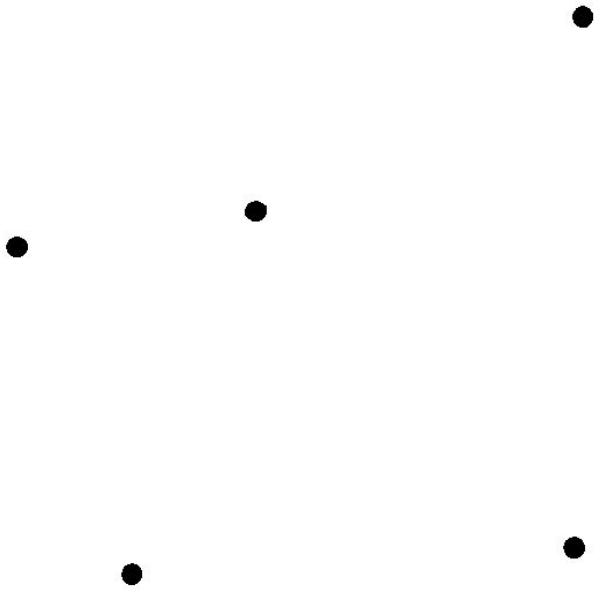

Fig6: Image showing disks with the smallest radius. They are obtained using disk shaped structuring element of size 8.

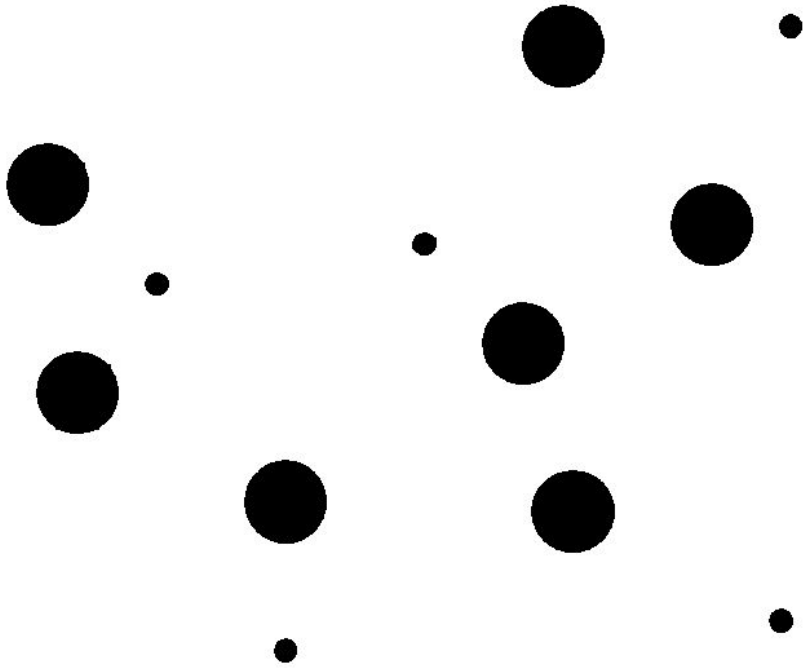

Fig7: Image showing only largest and smallest disks (union of the images obtained above).

If we do not remove the noise, the result obtained on applying hit-or-miss transform on the original image is shown below on the next page.

Fig8: Image after performing hit-or-miss on noisy image with a disk shaped structuring element of radius 8

All the results obtained for each step are totally as expected. The structuring elements are able to detect largest and smallest disks only when the salt and pepper noise is removed. When the noise is not removed, we are not able to detect any of the largest size disks. However, we are able to detect 3 small sized disks. This is because those three small disks did not have any salt noise. All the large sized disks have salt noise. Thus, only a white image is obtained when we applied hit-or miss transform with a disk shaped structuring element of radius 30.

D. Conclusion:

From this project, we conclude that hit-or-miss transform is useful for pattern detection, thickening as well as thinning the images. It is very crucial to remove the noise from the image before applying hit-or-miss transform. Otherwise, the results will be distorted.
We also learned that in Matlab, value ‘1’ is white and value ‘0’ is black. Initially, we implemented using 1 as black value and vice versa and the results obtained were totally opposite.