## CSE585/EE555:  Digital Image Processing II

## Computer Project # 1:

## Mathematical Morphology: Hit-or-Miss Transform

#### Sweekar Sudhakara, Savinay Nagendra, Nagarjuna Pampana, Prapti Panigrahi

#### Date: 02/06/2021

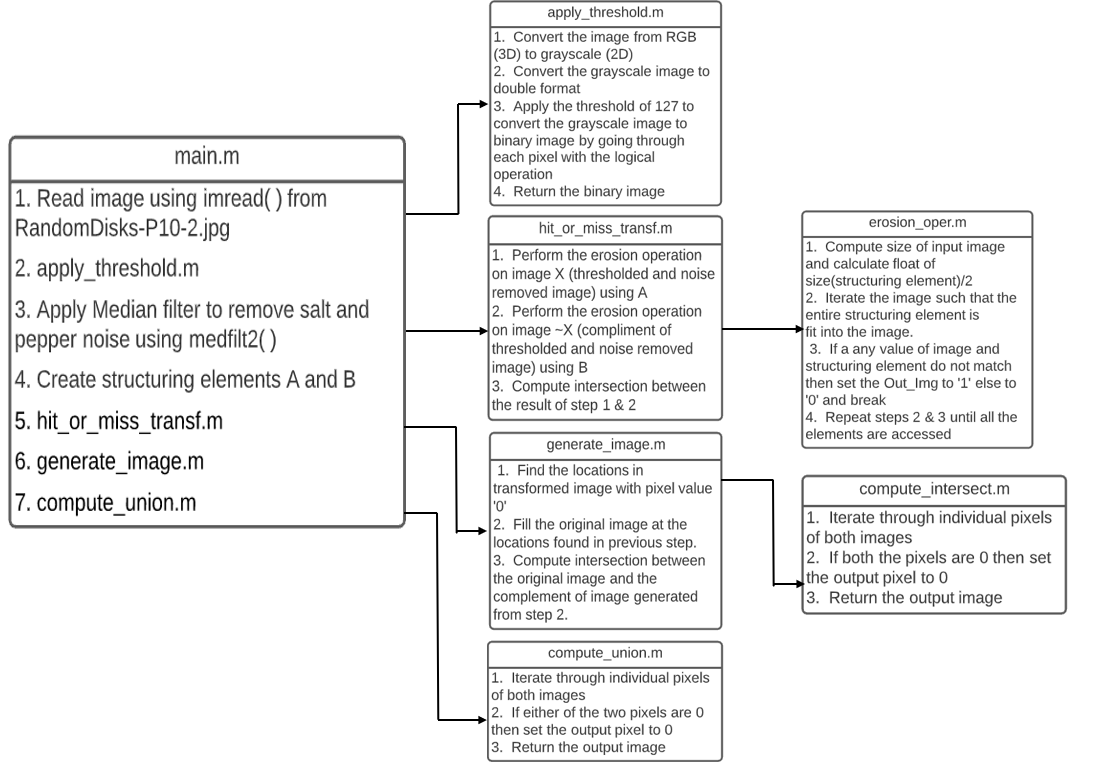
1. **Objectives**

The objective of this project is to detect patterns of a specific shape and size in noisy images by using morphological transformations. We design and implement one appropriate Hit-or-Miss transformation for detecting and extracting smallest and largest disks in the given image with added noise. This project will help us understand the practical implementations of set functions such as intersection and union and operations such as erosion. It will also help in understanding the effect of noise on the performance of morphological transformations.

1. **Methods**

The process flow followed involved in the project implementation is as follows:

1. Read the image “RandomDisks-P10.jpg”.
2. Convert the image to binary image with threshold of 127 (max pixel intensity is 255).
3. Remove the salt & pepper noise from the binary image using median filter.
4. Create disk shape structuring elements (A) using MATLAB ***strel*** with radii 30 (for largest disk) & 8 (for smallest disk), chosen after trial and error.
5. Create window based structuring elements (B) MATLAB ***strel*** with size 35 (for largest disk) & 10 (for smallest disk) chosen after trial and error.
6. Perform Hit-or-Miss transformation to obtain disks with a specific size.
7. Highlight the disks and save them.
8. The combined final output is obtained by taking the **union** of both the outputs.



**Figure 1:** Flowchart of MATLAB function calls.

**Conversion to Binary**

The grayscale image is converted to binary by using a threshold of 127. A logical operation stating that all pixels greater than 127 are labeled as 1 (background) and less or equal to 127 are labeled as 0 (foreground). The given image is converted to binary because Hit-or-Miss Transform is a binary morphological operation that requires 2 classes of pixels – foreground and background. We convert the image to binary by calling **apply\_threshold( )**.

**Denoising**

After converting the image from 3 channel to a single channel gray image, a threshold is applied to make the image binary. The binary image is noisy and filled with foreground and background cavities. To reduce the noise in the thresholded image, we use the median filter with the MATLAB call **medfilt2( )**. The median filter does a good job in preserving the useful detail in the image. The median filter considers each pixel and its neighbors to determine whether it gives a representation of its neighborhood. The value of each pixel is replaced by the median of its neighborhood pixels. Being a non-linear filter, it is very effective in removing the salt and pepper noise in the image.

**Hit-or-Miss Transformation**

The Hit-or-Miss transform is a binary morphological operation used in pattern recognition for detecting objects of different shapes and sizes an image. The structuring element used can contain both foreground and background pixels as opposed to erosion and dilation where the structuring element contains only foreground pixels. The origin of the structuring element is translated to all pixels in the given image and the structuring element is compared with the underlying overlapping pixels. If there is an exact match between the foreground and background pixels in the structuring element and that of the overlapping region, the pixel underneath the origin of the structing element is labeled as foreground. Otherwise, the pixel is labeled as background.

If is the given image with different sized disks, all in the foreground the objective of this project is to find the largest and smallest sized disks. To find disks of a particular size, we need two structuring elements and where contains foreground pixels and is the compliment of slightly bigger version of . The sizes of and are chosen with a trial-and-error search which is related to the size of the disk to be found. As a thumb rule, we make A slightly smaller than the size of the disk to be found.

The output of the Hit-or-Miss transform is given by:

**Visualization using MATLAB *imfill()***

After performing the Hit-or-Miss transformation, we obtain the image containing only the center of the disks in the form of single (or multiple) pixels. However, the single pixel is not clearly visible to the naked eye on the screen since it is “very small”. Hence, to clearly depict the disks whose locations have been extracted employing Hit-or-Miss transformation, we use the flood fill function available on MATLAB, i.e. ***imfill()*** using the locations of the single pixels to visualize the disks clearly.

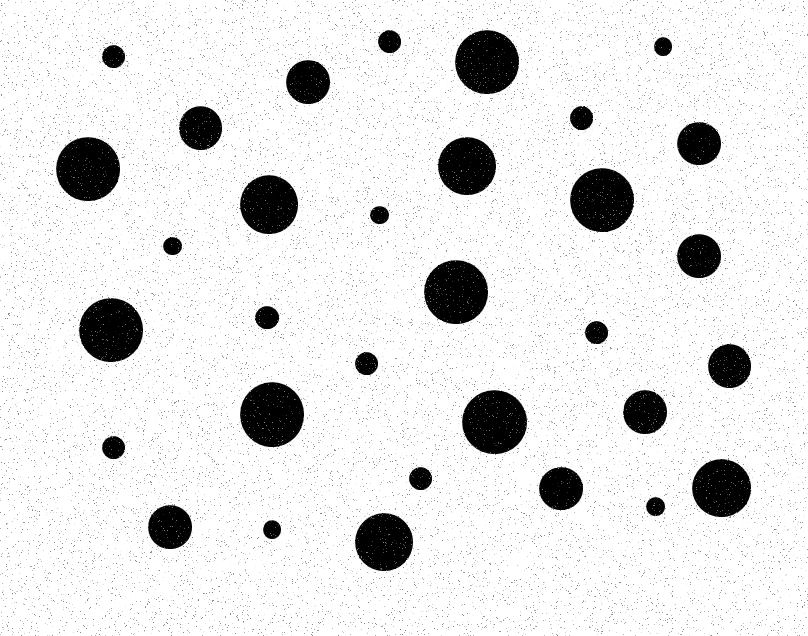
**Image Union**

To obtain the final output image containing both largest and smallest disks together, we perform the union operation between the largest disks image and the smallest disk image, i.e. if either of the pixels are black (pixel value=0) we include that pixel in the output.

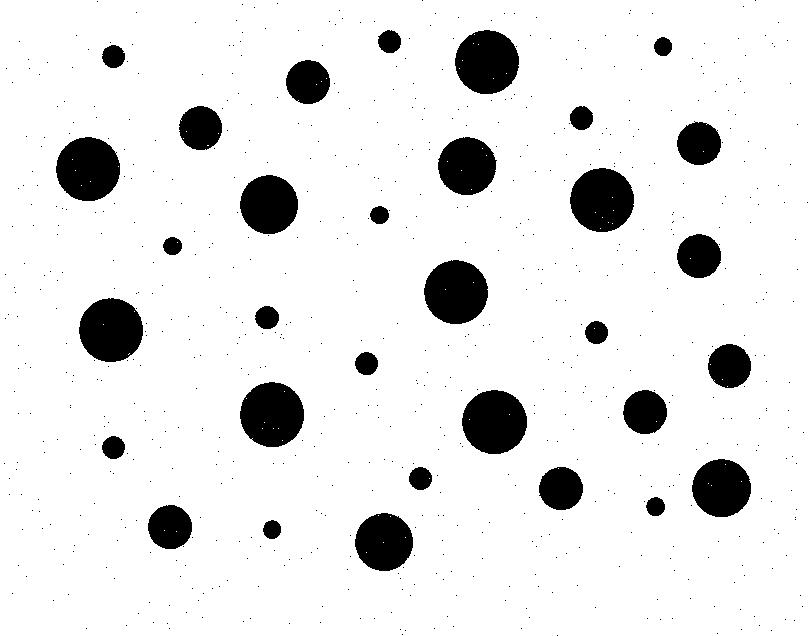
**NOTE:** It is enough to run the ***main.m*** file presented in the folder as it will make use of all the functions and will generate & save all the images locally.

1. **Results**

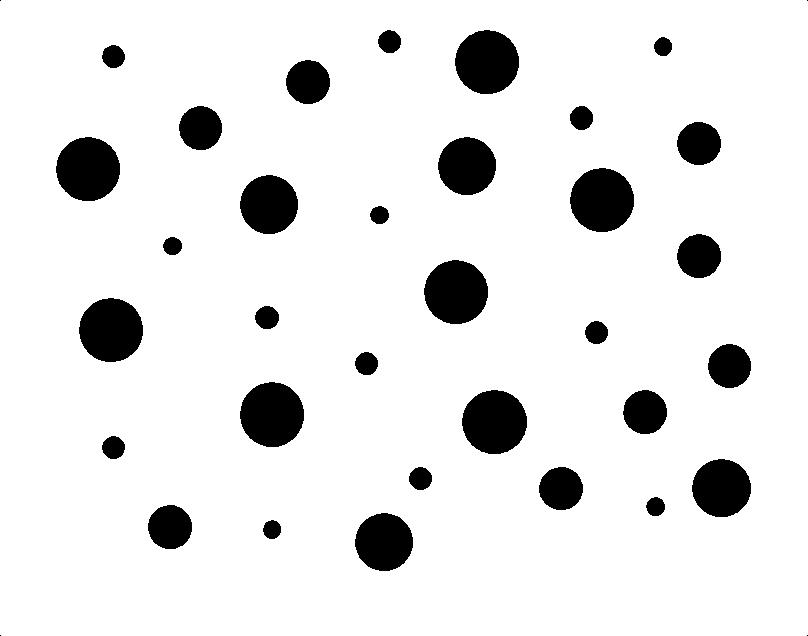
This section shows the results of applying Hit-or-Miss Transform to the original image.



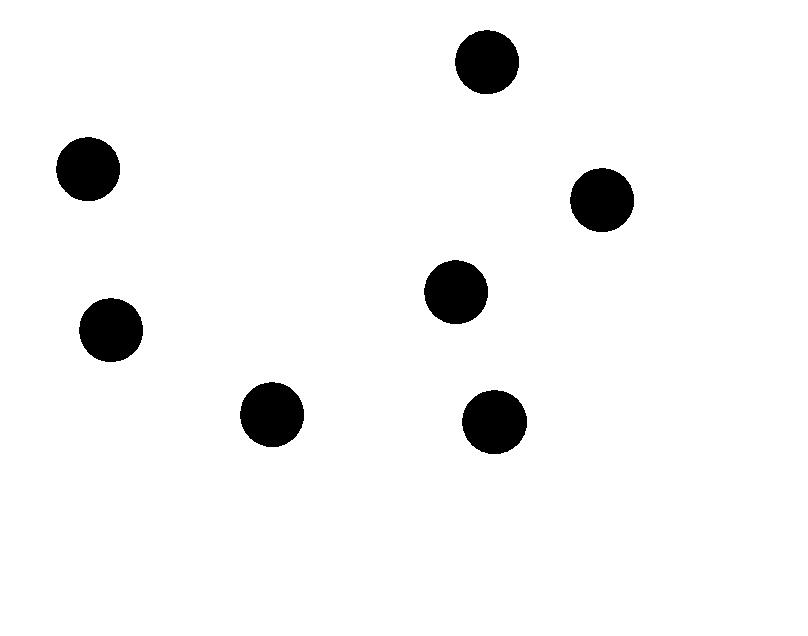
**Figure 2:** Original image RandomDisks-P10-2.jpg



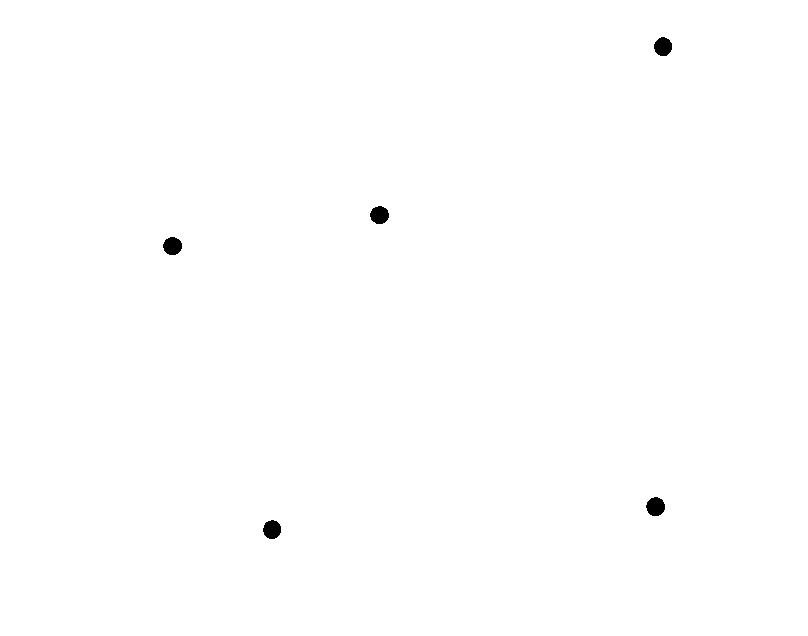
**Figure 3:** Image after applying converting to Binary



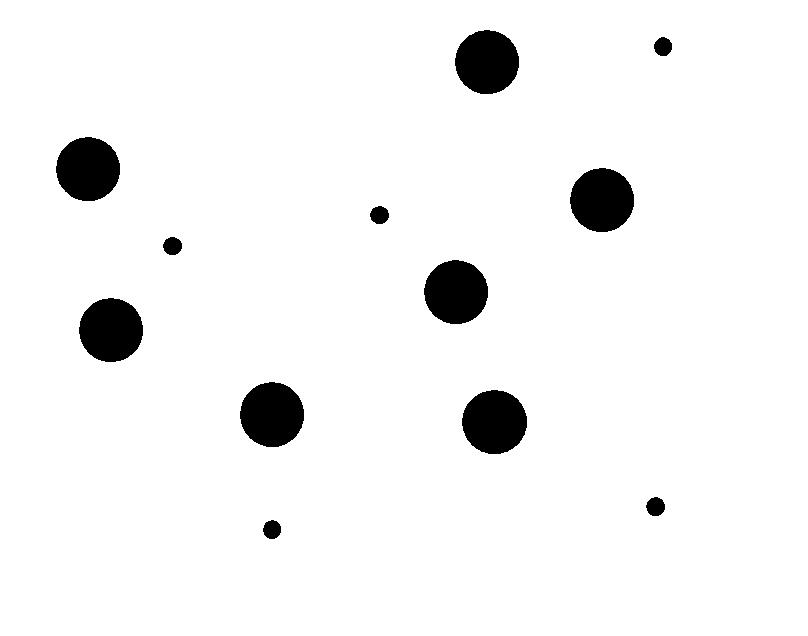
**Figure 4:** Image after applying Median Filter for denoising

****

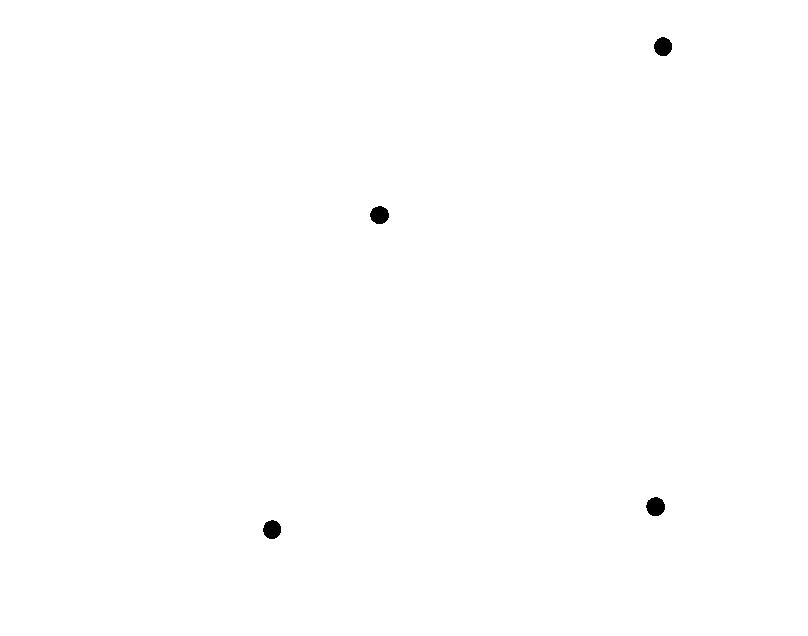
**Figure 5:** Image containing only large disks after applying Hit-or-Miss Transform



**Figure 6:** Image containing only small disks after applying Hit-or-Miss Transform

****

**Figure 7:** Image containing both smallest and largest disks, which is obtained from the Union of Figure 5 and Figure 6



**Figure 8:** This is the image containing both smallest and largest disks when Hit-or-Miss Transform is applied on the noisy image.

We get expected results. Figures 2 through 8 show the results of our implementation of the Hit-or-Miss transform applied to a noisy image to detect smallest and largest disks. It can be observed from Figure 3 that conversion to binary and thresholding reduces the noise. Applying the median filter further eliminates the noise, as can be observed in Figure 4. After the locations of the large and small disks are found, imfill function in MATLAB is used to reconstruct the disks. The reconstructed largest and smallest disks can be seen in Figures 5 and 6. The union of these two images gives the final output, which is shown in Figure 7. Figure 8 shows the effect of applying Hit-or-Miss Transform to an image which is noisy. It can be observed that all the positions of the disks are not captured and only a subset of the small disks are captured, which did not have salt and pepper noise. Large disks were not detected as they all have salt and pepper noise. Only white patches appear at the positions of disks noise. This shows that the output will be distorted when morphological operations are applied on noisy images.

1. **Conclusion**

We conclude that Hit-or-Miss transform, which is a binary morphological operation is useful in pattern recognition to detect objects of different shapes and sizes. We observe that denoising is a crucial preprocessing step before applying morphological transformations, without which the output will be distorted / incorrect.