CSE585/EE555: Digital Image Processing II

Computer Project # 1

Mathematical Morphology: Hit-or-Miss Transform

Mandar Parab, Amogh S. Adishesha, Lyuzhou Zhuang

Date: 1/29/2017

A. Objectives

In this project, we need to implement hit-or-miss transformation to filter out the biggest and the smallest disks on an image. Also get familiar with erosion, dilation and other filters we learned in class. And observe how noise cancelling affects the result of hit-or-miss transform.

B. Methods

As is shown in the flowchart (next page). We first convert image "RandomDisks-P10" into a binary-valued image X. Then use open/close filters to clean out the salt-and-pepper noise. Create structuring elements for hit-or-miss transform. Perform hit-or-miss transform on image X and get 2 images, one with only the smallest disks in X, one biggest disks. Merge the 2 images together and yields a final image Z with both the biggest and smallest disks.

Below is the MATLAB processing flow (Note: Run main.m and it will automatically call subfunctions to perform any operations required)

1. Convert the original image into an binary-valued image.

   Using a threshold value of 127, Pixels in the original image whose value are greater than threshold are set to "1" (white), otherwise, set to "0" (black). Let the result image be X.

2. Clean salt-and-pepper noise with open/close filters.

   Using the formula on lecture notes L4-11

1) Create a 3x3 structuring element B

2) Opening

First perform erosion on X with structuring element B, then perform Minkowski Set Addition on the previous result again with B.

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

3) Closing

First perform dilation on the result of opening with structuring element B, then perform Minkowski Set Subtraction on the previous result again with B.

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

3. Create structuring elements and perform hit-or-miss transform on image X, yielding result image Y1 with only the smallest disks in image X. (Based on lecture notes L4-8 and L4-9)

1) Create structuring elements a disk A and a holed-mask B

To create disk A, we generate a 16x16 square matrix with “0”s at the center and “1”s at the boundaries. The shape of the “0”s is similar to a disk whose diameter is slightly smaller than that of the smallest disks in image X. To create holed-mask B, we generate a 20x20 square matrix with “1” at the center and “0”s at the boundaries. The shape of the “1”s is similar to a disk whose diameter is slightly bigger than that of the smallest disks in image X.

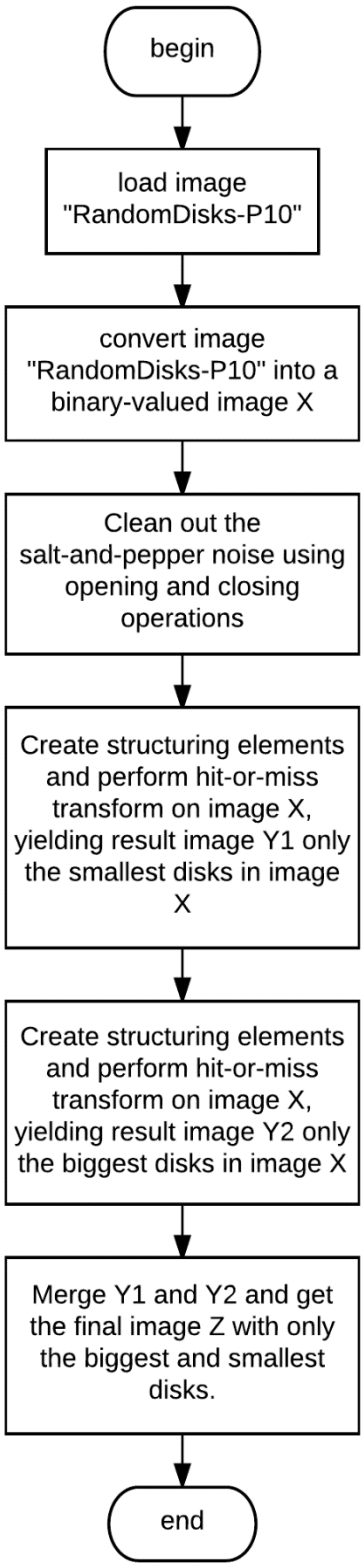

Figure 2.1 program flowchart

2) Perform hit-or-miss transform on image X

Following formula $X \circledast (A, B) = (X \ominus A^S) / (X \oplus B^S) = (X \ominus A^S) \cap (X \oplus B^S)^C$ we first perform erosion on X ($X \ominus A^S$), then dilation & complementary $(X \oplus B^S)^C$, after that we "and" the 2 results to get dots corresponding to the smallest disks in X. Finally, we blow the dots back to their original sizes by doing Minkowski Set Addition using a 5x5 structuring element B, then "and" the result with the X. After certain iterations, the dots are blew back to their original sizes. Let the result image be Y1. Note: the erosion/dilation subfunctions basically "and" or "or" the mask with the area it covers on the image then make decisions by how total number of "1"s or "0"s changes.

4. Create structuring elements and perform hit-or-miss transform on image X, yielding result image Y2 with only the biggest disks in image X. (Based on lecture notes L4-8 and L4-9)

1) Create structuring elements a disk A and a holed-mask B

To create disk A, we generate a 62x62 square matrix with "0"s at the center and "1"s at the boundaries. The shape of the "0"s is similar to a disk whose diameter is slightly smaller than that of the biggest disks in image X. To create holed-mask B, we generate a 72x72 square matrix with "1" at the center and "0"s at the boundaries. The shape of the "1"s is similar to a disk whose diameter is slightly bigger than that of the biggest disks in image X.

2) Perform hit-or-miss transform on image X

Following formula $X \circledast (A, B) = (X \ominus A^S) / (X \oplus B^S) = (X \ominus A^S) \cap (X \oplus B^S)^C$ we first perform erosion on X ($X \ominus A^S$), then dilation & complementary $(X \oplus B^S)^C$, after that we "and" the 2 results to get dots corresponding to the smallest disks in X. Finally, we blow the dots back to their original sizes by doing Minkowski Set Addition using a 5x5 structuring element B, then "and" the result with the X. After certain iterations, the dots are blew back to their original sizes. Let the result image be Y2.

5. Merge Y1 and Y2 and get the final image Z with only the biggest and smallest disks. ($Z = Y1 \cap Y2$)

6. To test if the hit-or-miss transform still works without pre-cleaning the noise, we skipped step 3 (open/close operation) but follows every other step. Result is shown in Fig 3.7.

C. Results

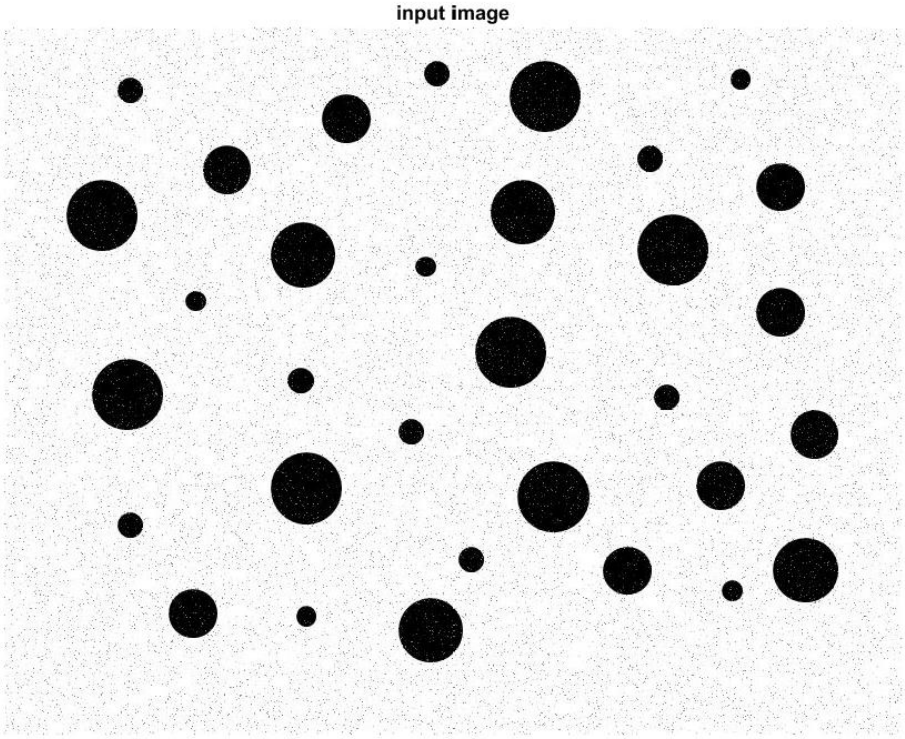

Figure 3.1 Original Image

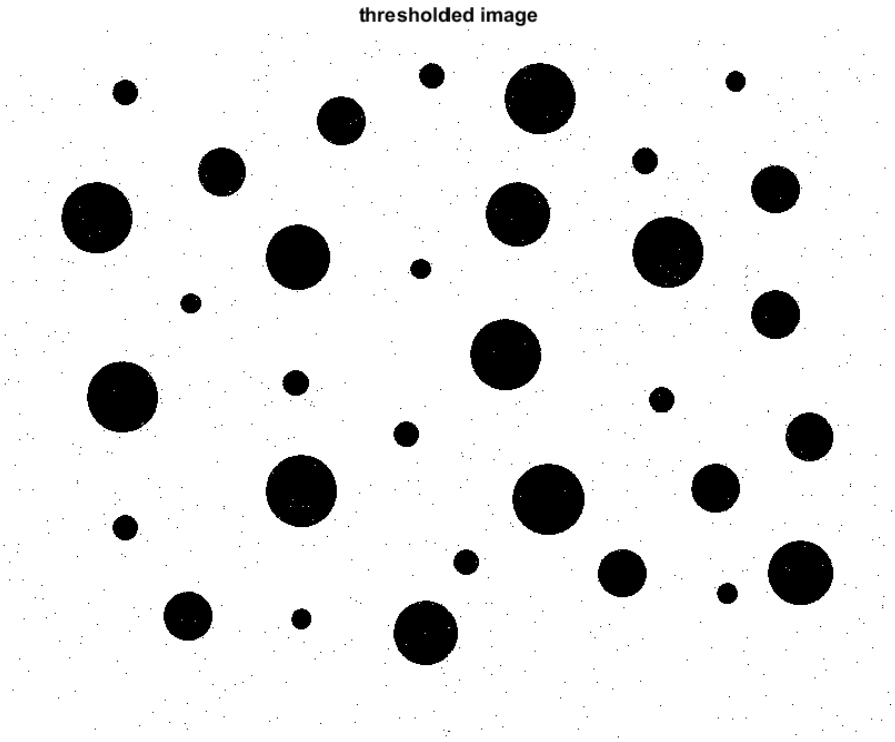

Figure 3.2 Result after thresholding the original image (binary-valued image) with threshold value 127

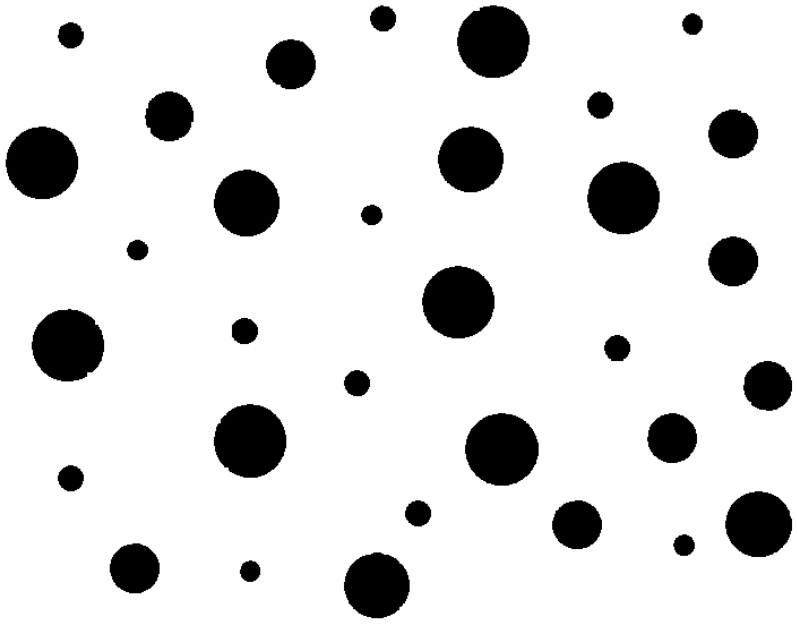

Figure 3.3 Result after applying the closing operation to the binary image with an 3x3 erosion operator and an 3x3 dilation operator

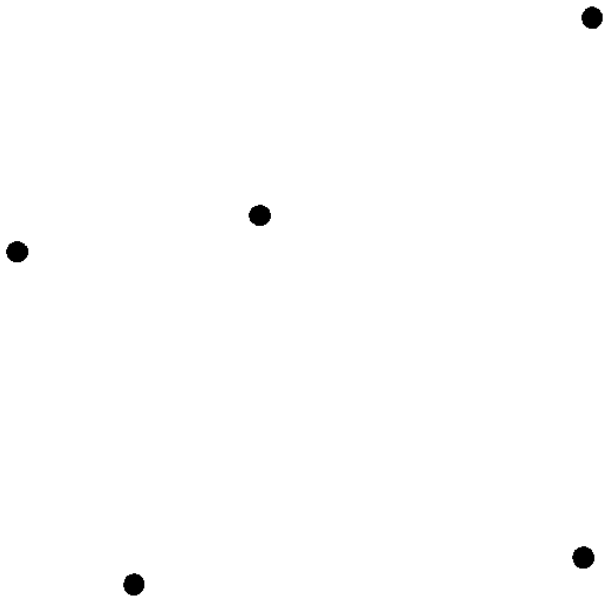

Figure 3.4 Result (Y1) after applying the hit-or-miss operation detecting the smallest disks

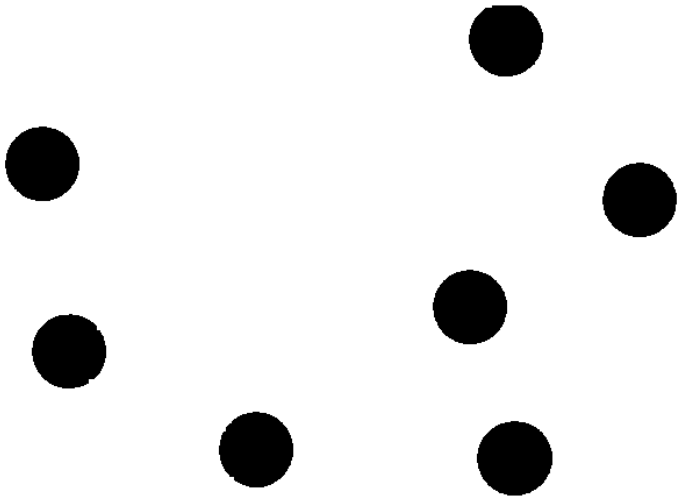

Figure 3.5 Result (Y2) after applying the hit-or-miss operation detecting the biggest disks

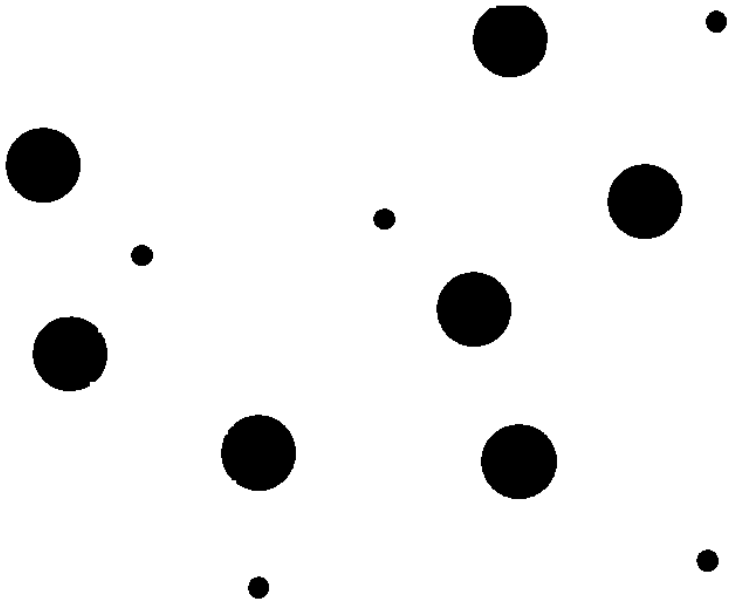

Figure 3.6 Final image (Z) with only the largest and the smallest disks (Merge Y1 and Y2)

If we do not clean out the salt-and-pepper noise prior to hit-or-miss transform. The transform does not work properly and the result shows as follow.

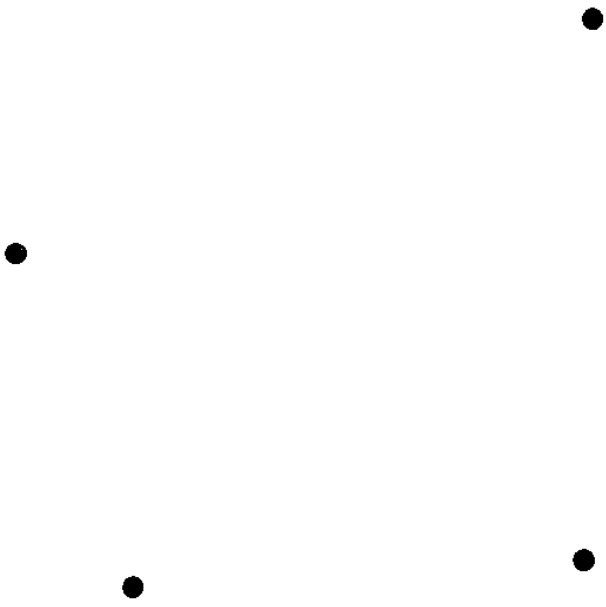

Figure 3.7 Final image without open/close operation

All results above shows exactly the outcome of each step in the process correctly as expected. If we clean out the salt and pepper noise before we perform hit-or-miss transform. The structuring elements A and B can detect all the desired disks as they are designed to do. If we do not clean out the noise (open/close) pre hit-or-miss, there'll be cavities in the disks and spots in the background which "confuse" A and B. So it's important to clean out the noise before performing hit-or-miss transform.

D. Conclusions

Through the project, we learn that the hit-or-miss transform is useful when detecting desired objects on an image. And pre-cleaning the noise (using open/close operations in this example) is essential for hit-or-miss transform to work properly. Also notice that the hit-or-miss transform is computationally intensive. On our own machines, performing the whole task takes about 5minutes.

Another lesson we learned through the project is about programming. During the code implementation, we were stuck at the part of getting X minkowski addition with W-A of the hit-or-miss transform. As discussed in the class, hit-or-miss transform has two parts one in which an image X is minkowski subtracted from mask A and the other in which the same image is minkowski added with mask W-A. It was after many tests that we could obtain the desired result for minkowski addition. Our algorithm for minkowski addition with mask B(W-A) has two part and both the parts require the image X to be compared with the mask B. In it we place the mask B over the image and do the following 1) check if the image pixel contained in the mask are all white 2) if the zeros in the image coincides with the zeros in the mask B then the we get a white pixel as the output at the center of the image where the mask is applied. This algorithm gives the result we want however the comparison in the two steps mentioned above slows down the whole hit-or-miss transform.