R09945020 朱彥如

Homework 4

Write programs which do binary morphology on a binary image:
(a) Dilation
(b) Erosion
(c) Opening
(d) Closing
(e) Hit-and-miss transform

The requirements above are accomplished by Python. All the histograms are created by matplotlib package, and all the processed images are exported by cv2 package.

(a) Dilation
For the dilation function, I use a binary octagon as the kernel implemented by 5×5 np.array. The inputs of the function are original three dimensional image and the kernel array. The kernel is applied on each pixel of lina.bmp image to detect the neighbor pixels needed to be performed AND operation with corresponding kernel pixels. For each pixel of original image, the boundary of neighbor pixels needed to be detected is calculated before the AND perforation. The two dimension array including length and width is created after all the AND operations. Then, I use np.stack to stack three same two dimensional dilation binary array into the three dimensional image array. At last, the function will return the image processed by dilation.

(b) Erosion
For the erosion function, I also use a binary octagon as the kernel implemented by 5×5 np.array. The inputs of the function are original three dimensional image and the kernel array. The kernel is applied on each pixel of lina.bmp image to detect the neighbor pixels needed to be performed OR operation with corresponding kernel pixels. Different from the dilation, all the pixels needed to be detected for OR operation is pre-calculated before the double for-loops because there is no need for the boundary pixels of the image where kernel cannot fit to be detecting for OR operation. Those boundary pixels of the returned image need to be performed erosion, so I just set them as 0 in the first place. Then, I use np.stack to stack three same two dimensional erosion binary array into the three dimensional image array. At last, the function will return the image processed by erosion.

(c) Opening
According to the definition of opening, I apply the erosion function on original image first, and then apply the dilation function to the eroded image to get the opening image.

(d) Closing
According to the definition of closing, I apply the dilation function on original image first, and then apply the erosion function to the dilated image to get the opening image.

(e) Hit-and-miss transform
For the hit-and-miss transform function, the inputs are original image, kernel, the origin coordinate of J kernel, and the origin coordinate of K kernel. The kernel is implemented by 2×2 np.array. In the function, the boundaries of detecting J kernel and K kernel are calculated respectively in the first place. Then, because of the difference between J and K kernel boundaries, the erosion on each pixels for J kernel and K kernel is operated in different for-loops, and also the erosion by K kernel is applied on complement of the

original image. The hit-and-miss transformed image is produced by AND operation on two images created with K and J kernels respectively.

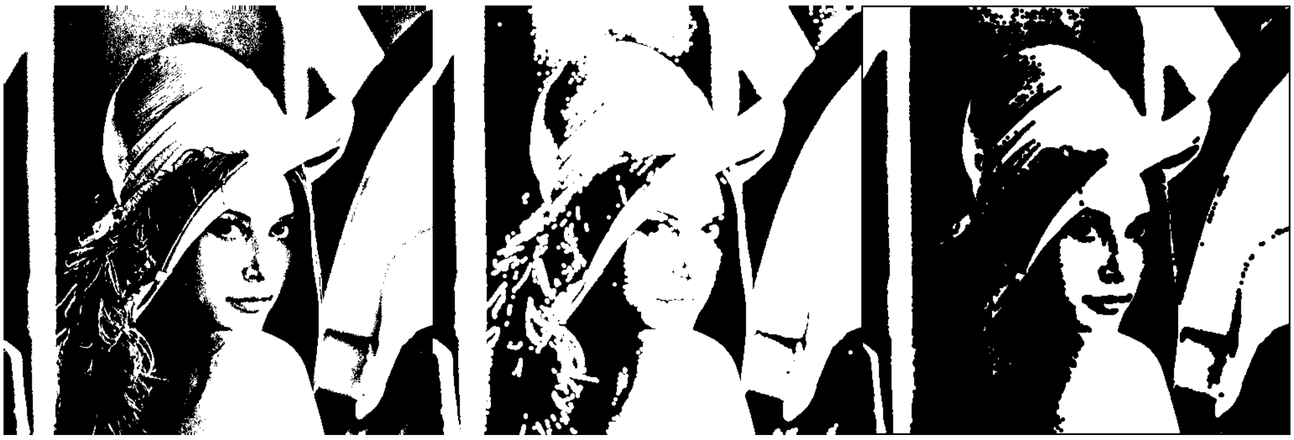
Figure 1. The left one is the binary image of original lena.bmp with the threshold 128, and the middle one is the binary image with dilation. The right one is the binary image with erosion.

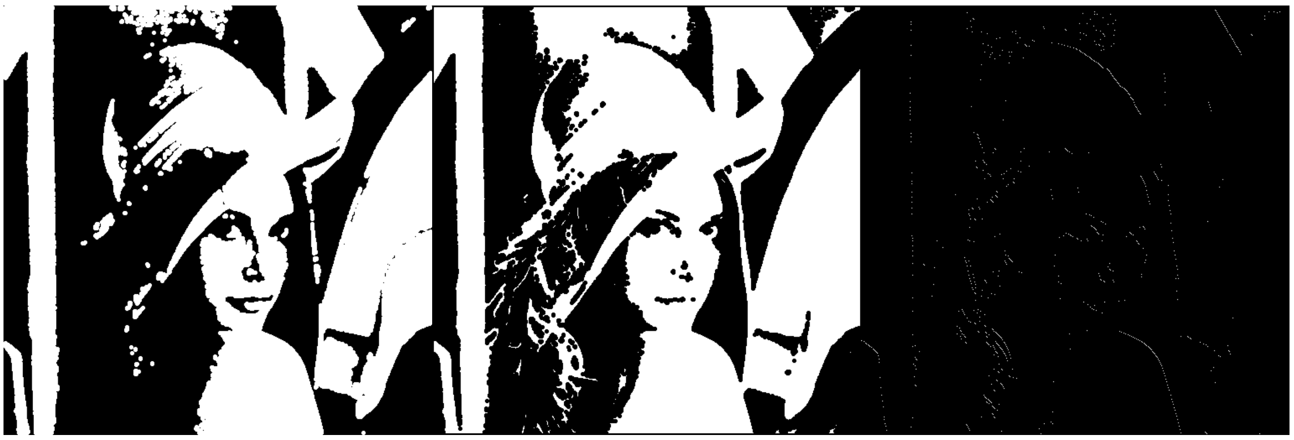
Figure 2. The left one is the binary image of opening, and the middle one is the binary image with closing. The right one is the binary image with hit-and-miss transform.