Computer Vision I: Homework 5

Department: CSIE Student ID: d11922016 Name: Jia-Wei, Liao

October 8, 2022

Write programs which do gray-scale morphology on a gray-scale image (lena.bmp):

(a) Dilation

(b) Erosion

(c) Opening

(d) Closing

Let I be the original image and $K = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}$ be the kernel.

1 Dilation

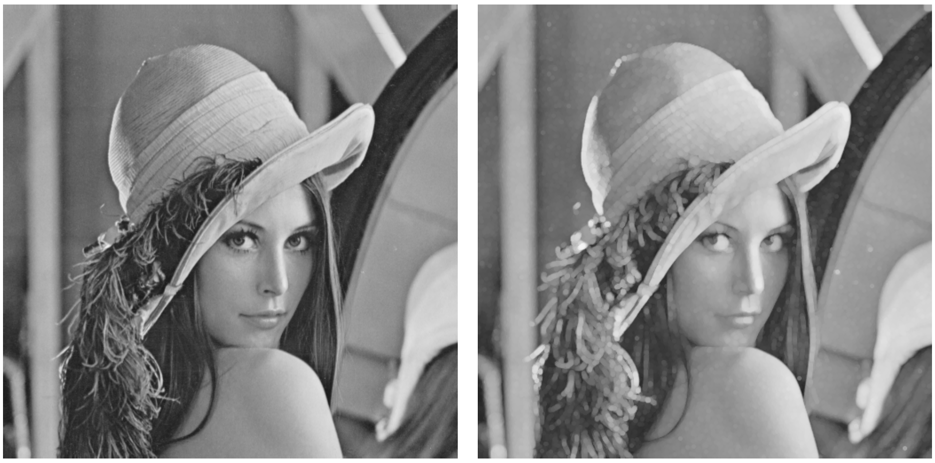
The dilation of I by K is

$$(I \oplus K)_{i,j} = \max_{(p,q)\in \text{supp}(K)} I(i+p-3, j+q-3).$$

```matlab
[m, n] = size(img); [k, ~] = size(kernel);
dilation_image = zeros(m, n);
img = padding(img, floor(k / 2));
condition =@(pixel, val) pixel > val;

for i = 1:m
    for j = 1:n
        local_image = get_local_image(img, kernel, i, j);
        dilation_image(i, j) = get_pixel(local_image, kernel, condition);
    end
end

dilation_image = uint8(dilation_image);
```



(a) gray-scale image

(b) gray-scale image after dilation

Figure 1: dilation

2 Erosion

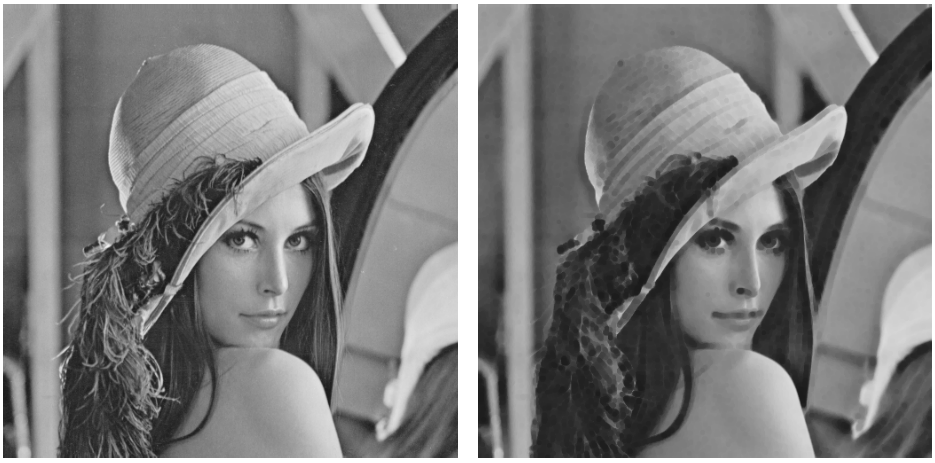
The erosion of I by K is

$$(I \ominus K)_{i,j} = \min_{(p,q)\in \mathrm{supp}(K)} I(i+p-3, j+q-3).$$

```
[m, n] = size(img); [k, ~] = size(kernel);
erosion_image = zeros(m, n);
img = padding(img, floor(k / 2));
condition =@(pixel, val) pixel < val;

for i = 1:m
    for j = 1:n
        local_image = get_local_image(img, kernel, i, j);
        erosion_image(i, j) = get_pixel(local_image, kernel, condition);
    end
end

erosion_image = uint8(erosion_image);
```



(a) gray-scale image (b) gray-scale image after erosion

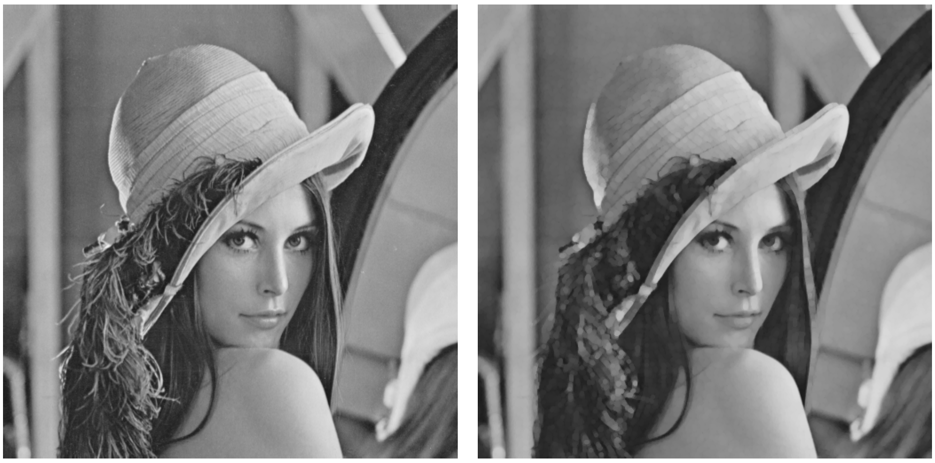
Figure 2: erosion

3 Opening

The opening of I by K is

$$I \circ K = (I \ominus K) \oplus K.$$

```
opening_image = erosion(img, kernel);
opening_image = dilation(opening_image, kernel);
opening_image = uint8(opening_image);
```



(a) gray-scale image (b) gray-scale image after opening

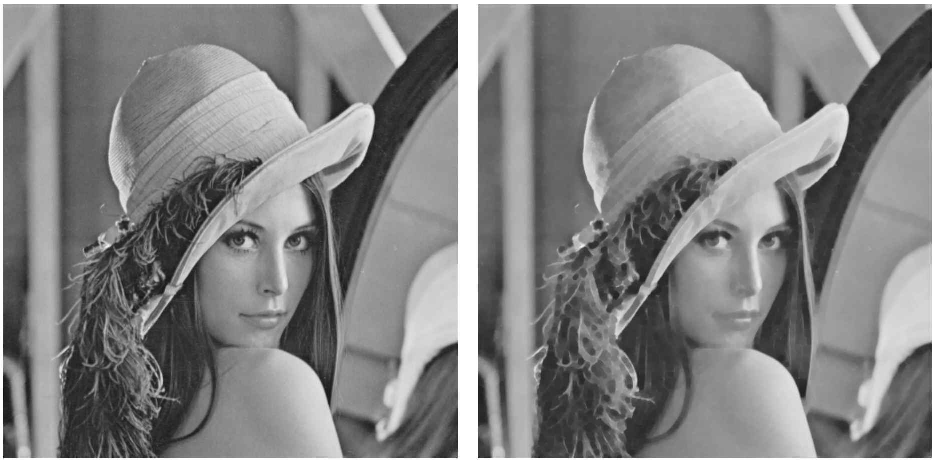
Figure 3: opening

4 Closing

The closing of I by K is

$$I \bullet K = (I \oplus K) \ominus K.$$

```
closing_image = dilation(img, kernel);
closing_image = erosion(closing_image, kernel);
closing_image = uint8(closing_image);
```



(a) gray-scale image (b) gray-scale image after closing

Figure 4: closing

Summary.

In this homework, we use a kernel-based algorithm to implement morphological operations on grayscale images. Unlike binary images, grayscale images require dilation to locate local maxima without four corners. Also, grayscale images erode to find local minima without four corners. We show the results of the image after dilation, erosion, opening and closing.