Computer Vision
Homework 4 Report

林義聖
B03902048

October 31, 2016

Introduction

I use *Python* as my programming language and *Pillow* as my Image Library.

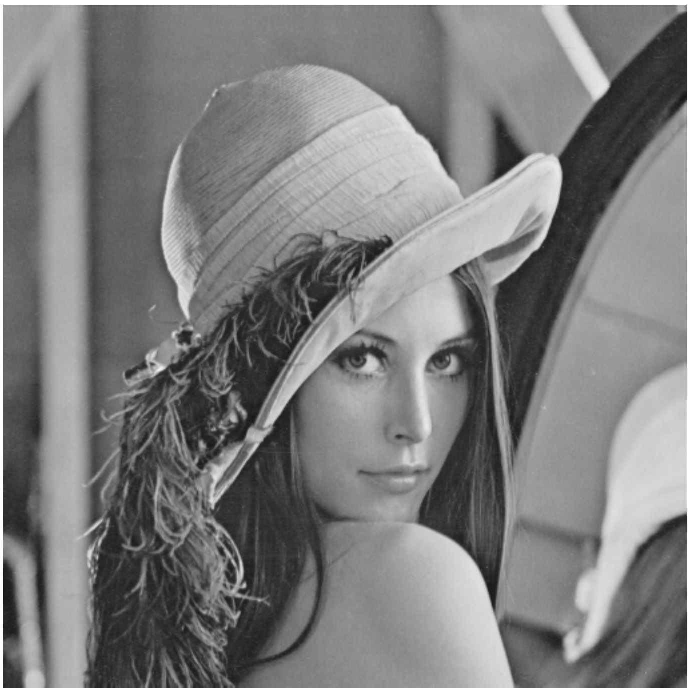

Figure 1: original lena.bmp

Erosion

Listing 1: Erosion

```
mask_2d = [\
    (−1,−2),(0,−2),(1,−2),\
    (−2,−1),(−1,−1),(0,−1),(1,−1),(2,−1),\
    (−2,0),(−1,0),(0,0),(1,0),(2,0),\
    (−2,1),(−1,1),(0,1),(1,1),(2,1),\
    (−1,2),(0,2),(1,2)\
]

for y in range(height):
    for x in range(width):
        if data_seq[y * width + x] == WHITE:
            for m in mask_2d:
                p = (x + m[0], y + m[1])
                if p[0] < 0 or p[0] >= width or p[1] < 0 or p[1] >=
                   height:
                    break
                elif data_seq[p[1] * width + p[0]] != WHITE:
                    break
            else:
                out_data[y * width + x] = WHITE
```

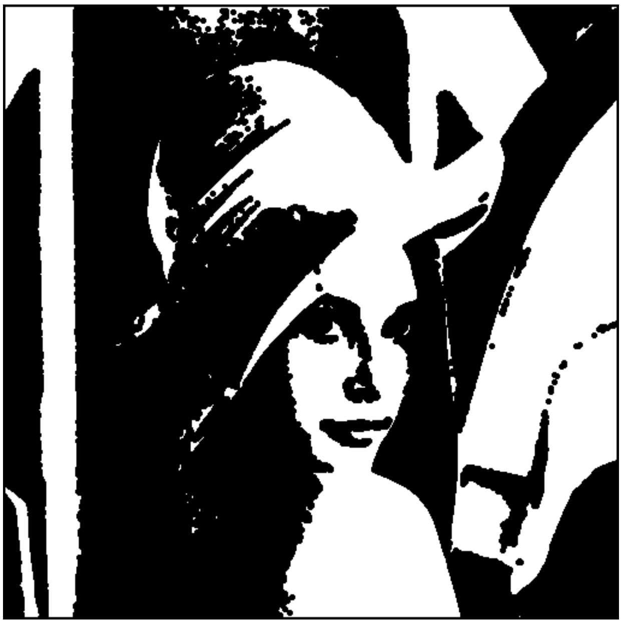

Figure 2: eroded lena.bmp

Dilation

When I am using dilation on the image, I use the same mask defined in previous part.

Listing 2: Dilation

```
for y in range(height):
    for x in range(width):
        if data_seq[y * width + x] == WHITE:
            for m in mask_2d:
                p = (x + m[0], y + m[1])
                if 0 <= p[0] < width and 0 <= p[1] < height:
                    out_data[p[1] * width + p[0]] = WHITE
```

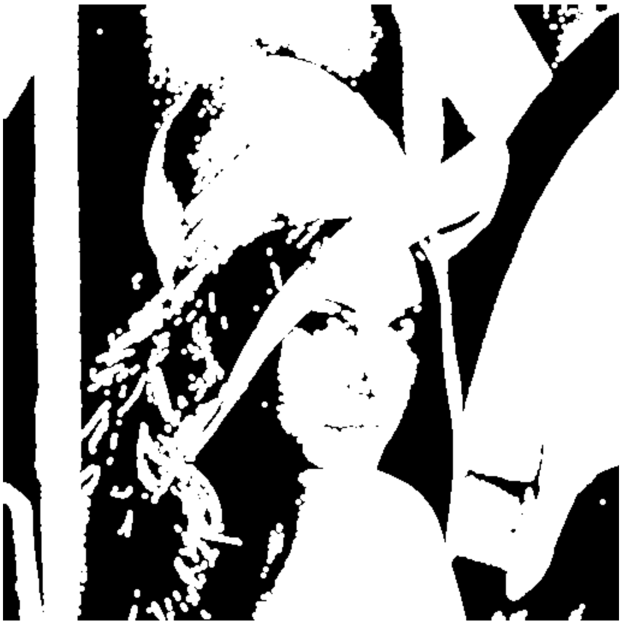

Figure 3: dilated lena.bmp

Opening

For doing *opening* on the image, I use erosion followed by dilation with the same kernel.

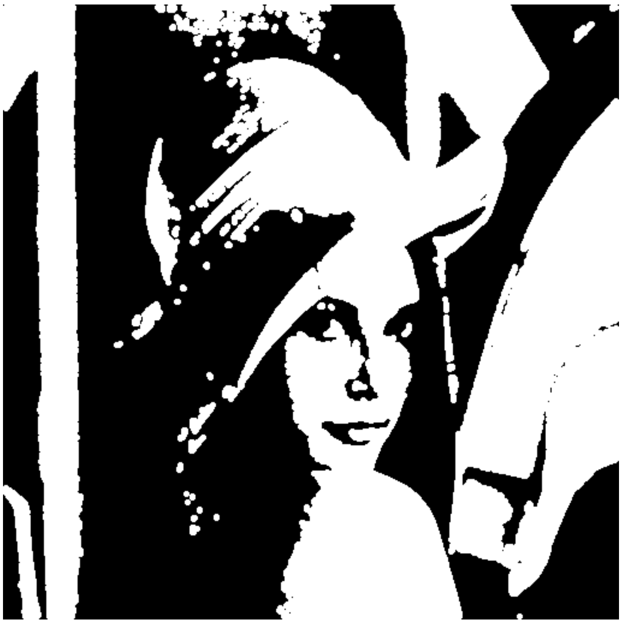

Figure 4: lena.bmp after opening

Closing

For doing *closing* on the image, I use dilation followed by erosion with the same kernel.

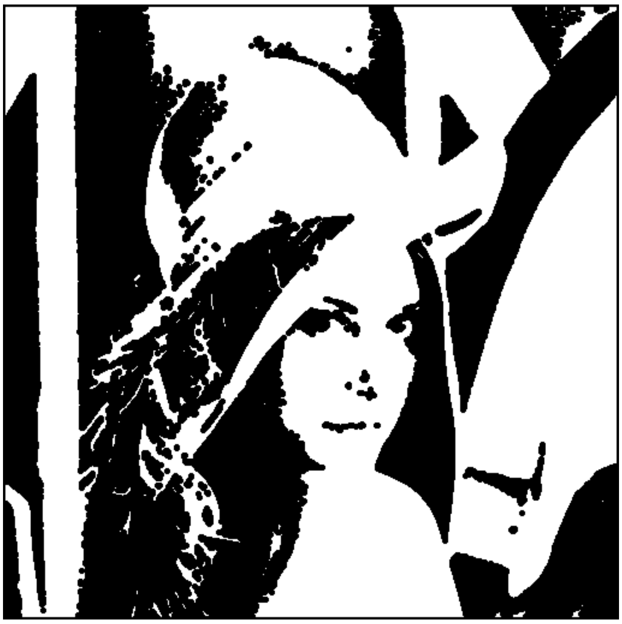

Figure 5: lena.bmp after closing

Hit-and-Miss

I draw white points on a black picture to represent the detected upper-right corners on lena.bmp.

Listing 3: Hit-and-Miss

```
mask_pos = [\
    (-1,-1),(0,-1),(1,-1),\
    (-1,0),(0,0),(1,0),\
    (-1,1),(0,1),(1,1)]

mask = [\
    0,-1,-1,\
    1,1,-1,\
    0,1,0]

for y in range(height):
    for x in range(width):
        if data_seq[y*width+x] == WHITE:
            for i in range(len(mask_pos)):
                p = (x + mask_pos[i][0], y + mask_pos[i][1])
                if 0 <= p[0] < width and 0 <= p[1] < height:
                    if mask[i] == 1 and data_seq[p[1]*width+p[0]] == WHITE:
                        continue
                    elif mask[i] == -1 and data_seq[p[1]*width+p[0]] == BLACK:
                        continue
                    elif mask[i] == 0:
                        continue
                    else:
                        break
            else:
                draw.point((x,y), fill=255)
```

Figure 6: lena.bmp after hit-and-miss

How to Use

There are 5 programs,

1. *erosion.py*

2. *dilation.py*

3. *opening.py*

4. *closing.py*

5. *hit-and-miss.py*

You need to use binarized picture as input and enter commands in this format: "*program [input image name] [output image name]*" to use it. For example, `./erosion.py lena-binarized.bmp output.bmp`.