

## Homework4

### Mathematical Morphology - Binary Morphology

#### Description

Write programs which do binary morphology on a binary image:

##### (-) binarization

Use the binarization in homework 3 to generate the binary image of lena.bmp

##### (a) Dilation

Scan each pixel to be the center and compare the 5x5 grid to the octagonal 3-5-5-5-3 kernel. If there are greater than or equal to one same pixel, set the current pixel (center) to be 255 on a new matrix.

Otherwise, set it to 0.

```
58 void dilation(Mat bin, Mat dil) {
59     for (int i = 2; i < img_rows - 2; i++) {
60         for (int j = 2; j < img_cols - 2; j++) {
61             int flag = 0;
62             for (int a = i - 2; a < i + 3; a++) {
63                 for (int b = j - 2; b < j + 3; b++) {
64                     if (a == i - 2 && b == j - 2) {
65                         continue;
66                     }
67                     else if (a == i - 2 && b == j + 2) {
68                         continue;
69                     }
70                     else if (a == i + 2 && b == j - 2) {
71                         continue;
72                     }
73                     else if (a == i + 2 && b == j + 2) {
74                         continue;
75                     }
76                     else {
77                         if (bin.at<uchar>(a, b) == 255) {
78                             flag = 1;
79                             break;
80                         }
81                     }
82                 }
83             if (flag == 1) {
84                 break;
85             }
86         }
87         if (flag == 1) {
88             dil.at<uchar>(i, j) = 255;
89         }
90         else {
91             dil.at<uchar>(i, j) = 0;
```

##### (b) Erosion

Scanning each pixel to be the center and comparing the 5x5 grid to the octagonal 3-5-5-5-3 kernel. If they are the same, set the current pixel (center) to be 255. Otherwise, set it to 0.

```

23 void erosion(Mat bin, Mat ero) {
24     for (int i = 2; i < img_rows-2; i++) {
25         for (int j = 2; j < img_cols-2; j++) {
26             int count = 0;
27             for (int a = i - 2; a < i + 3; a++) {
28                 for (int b = j - 2; b < j + 3; b++) {
29                     if (a == i - 2 && b == j - 2) {
30                         continue;
31                     }
32                     else if (a == i - 2 && b == j + 2) {
33                         continue;
34                     }
35                     else if (a == i + 2 && b == j - 2) {
36                         continue;
37                     }
38                     else if (a == i + 2 && b == j + 2) {
39                         continue;
40                     }
41                     else {
42                         if (bin.at<uchar>(a, b) == 255) {
43                             count++;
44                         }
45                     }
46                 }
47             }
48             if (count == 21) {
49                 ero.at<uchar>(i, j) = 255;
50             }
51             else {
52                 ero.at<uchar>(i, j) = 0;

```

### (c) Opening

Apply erosion and then dilation to the binary image. The program uses the result of (b) and applies dilation on it.

```

167 //Opening
168 Mat open = bin.clone();
169 dilation(ero, open);
170 imwrite("c_open.jpg", open);

```

### (d) Closing

Apply dilation and then erosion to the binary image. The program uses the result of (a) and applies erosion on it.

```

172 //Closing
173 Mat close = bin.clone();
174 erosion(dil, close);
175 imwrite("d_close.jpg", close);

```

### (e) Hit-and-miss transform

1. Create the complement of the binary image by using the binarization function.
2. Apply erosion with J-kernel on the binary image and apply erosion with K-kernel on the complement of the binary image.
3. Calculate the conjunction of the results above.

```

//hit and miss
Mat J = bin.clone(); //J
Mat bin_c = original.clone(); //A_com
Mat K = bin_c.clone(); //K
Mat hit(img_rows, img_cols, CV_8UC1, Scalar(0));

binarize(bin_c, 0, 255);
ero_J(bin, J);
ero_K(bin_c, K);

for (int i = 0; i < img_rows; i++) {
    for (int j = 0; j < img_cols; j++) {
        hit.at<uchar>(i, j) = (J.at<uchar>(i, j)* K.at<uchar>(i, j))/255;
    }
}

imwrite("e_hitandmiss.jpg", hit);

return 0;

```

```

122 void ero_K(Mat bin, Mat output) {
123     for (int i = 1; i < img_rows - 1; i++) {
124         for (int j = 0; j < img_cols - 2; j++) {
125             int count = 0;
126             for (int a = i - 1; a < i + 1; a++) {
127                 for (int b = j; b < j + 2; b++) {
128                     if (a == i && b == j) {
129                         continue;
130                     }
131                     else {
132                         if (bin.at<uchar>(a, b) == 255) {
133                             count++;
134                         }
135                     }
136                 }
137             }
138             if (count == 3) {
139                 output.at<uchar>(i, j) = 255;
140             }
141             else {
142                 output.at<uchar>(i, j) = 0;
143             }
144         }
145     }
146 }

```

```

97 void ero_J(Mat bin, Mat output) {
98     for (int i = 0; i < img_rows - 2; i++) {
99         for (int j = 1; j < img_cols; j++) {
100             int count = 0;
101             for (int a = i; a < i + 2; a++) {
102                 for (int b = j - 1; b < j + 1; b++) {
103                     if (a == i+1 && b == j - 1) {
104                         continue;
105                     }
106                     else {
107                         if (bin.at<uchar>(a, b) == 255) {
108                             count++;
109                         }
110                     }
111                 }
112             }
113             if (count == 3) {
114                 output.at<uchar>(i, j) = 255;
115             }
116             else {
117                 output.at<uchar>(i, j) = 0;
118             }
119         }
120     }
121 }

```

## Result

(a) Dilation



(b) Erosion



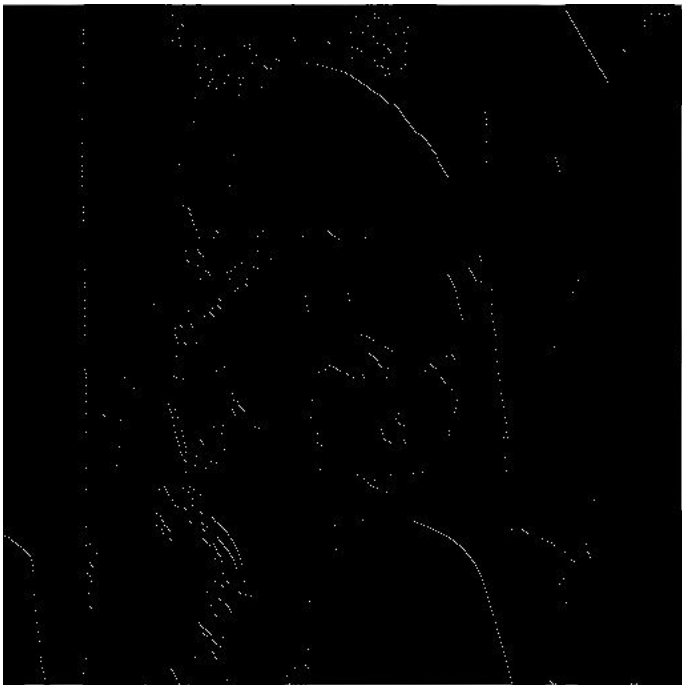
(c) Opening



(d) Closing



(e) Hit-and-miss transform



Reference:

1. [https://www.youtube.com/watch?v=HcpxKCdzjdY&ab\\_channel=RudraSingh](https://www.youtube.com/watch?v=HcpxKCdzjdY&ab_channel=RudraSingh)
2. lecture slide