Computer vision

B06902091 資工四 羅寶瑩

Homework5

Mathematical Morphology - Gray Scaled Morphology

Description

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

Each part of the program is based on homework4.

(a) Dilation

Scan each pixel to be the center. For each center, scan with the octagonal 3-5-5-5-3 kernel and find the local maxima to be the new value of the current pixel.

```
41  void dilation(Mat ori, Mat dil) {
42      for (int i = 2; i < img_rows - 2; i++) {
43          for (int j = 2; j < img_cols - 2; j++) {
44              int maximum = -10;
45              for (int a = i - 2; a < i + 3; a++) {
46                  for (int b = j - 2; b < j + 3; b++) {
47                      if (a == i - 2 && b == j - 2) {
48                          continue;
49                      }
50                      else if (a == i - 2 && b == j + 2) {
51                          continue;
52                      }
53                      else if (a == i + 2 && b == j - 2) {
54                          continue;
55                      }
56                      else if (a == i + 2 && b == j + 2) {
57                          continue;
58                      }
59                      else {
60                          if (ori.at<uchar>(a, b) >= maximum) {
61                              maximum = ori.at<uchar>(a, b);
62                          }
63                      }
64                  }
65              }
66              dil.at<uchar>(i, j) = maximum;
67          }
68      }
69  }
```

(b) Erosion

Scan each pixel to be the center. For each center, scan with the octagonal 3-5-5-5-3 kernel and find the local minima to be the new value of the current pixel.

```
11  void erosion(Mat ori, Mat ero) {
12      for (int i = 2; i < img_rows - 2; i++) {
13          for (int j = 2; j < img_cols - 2; j++) {
14              int minimum = 999;
15              for (int a = i - 2; a < i + 3; a++) {
16                  for (int b = j - 2; b < j + 3; b++) {
17                      if (a == i - 2 && b == j - 2) {
18                          continue;
19                      }
20                      else if (a == i - 2 && b == j + 2) {
21                          continue;
22                      }
23                      else if (a == i + 2 && b == j - 2) {
24                          continue;
25                      }
26                      else if (a == i + 2 && b == j + 2) {
27                          continue;
28                      }
29                      else {
30                          if (ori.at<uchar>(a, b) <= minimum) {
31                              minimum = ori.at<uchar>(a, b);
32                          }
33                      }
34                  }
35              }
36              ero.at<uchar>(i, j) = minimum;
37          }
38      }
39  }
```

(c) Opening

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

```
87      //Opening
88      Mat open = original.clone();
89      dilation(ero, open);
90      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.

```
92      //Closing
93      Mat close = original.clone();
94      erosion(dil, close);
95      imwrite("d_close.jpg", close);
```

Result

(a) Dilation

(b) Erosion

(c) Opening

(d) Closing

Reference:

1. *lecture slide*