

# 浙江大学

## 本科实验报告

边界追踪

课程名称： 数字图像处理

姓 名：

学 院： 信息与电子工程学院

专 业： 信息工程

学 号：

指导老师： 李东晓

2023 年 6 月 4 日

## 一、 实验任务

本次选择的是 PROJECT-10-02 题目。

- (1) 编写一个基于 Moore 边界跟踪算法的边界跟踪程序，这里我们使得被跟踪得边界通过图像得方式输出。
- (2) 下载图 9.14 来应用算法。

## 二、 算法设计

算法实现如下：

- (1) 找到一个黑色像素，并将它定为你的起始像素。
- (2) 每次遇到黑色像素，将设置为当前边界像素点
- (3) 然后原路返回到先前到达的白色像素，以顺时针方向搜索的摩尔邻域内的每一个像素，直到遇到下一个黑色像素。
- (4) 重复这个过程，当起始像素被第二次访问时算法终止，在整个运行过程走过的 色像素就是目标的边界像素。

## 三、 代码实现

本次实验编程语言选择的是 Matlab。

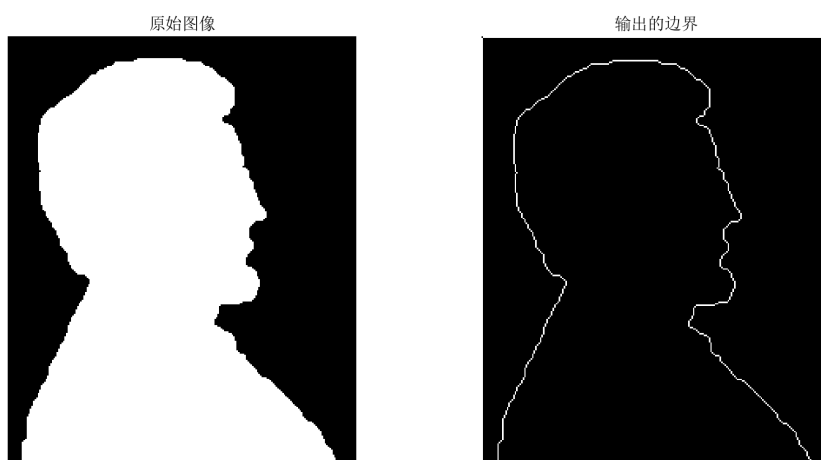
```
1 img = imread("test.tif");
2
3 [m, n] = size(img);
4 % 边界扩展
5 zero_ac = ones(m, 2);
6 zero_ar = ones(2, n + 4);
7 img_new = [zero_ar; [zero_ac, img, zero_ac]; zero_ar];
8
9 img_mid = zeros(m + 2, n + 2);
10
11 b_0_m = 0;
12 b_0_n = 0;
13
14 for i = 2:1:m + 1
15
16     for j = 2:1:n + 1
17
18         if (img_new(i, j - 1) == 0 && img_new(i - 1, j) == 0 && img_new(i, j) == 1)
19             b_0_m = i;
20             b_0_n = j;
21             break
22         end
23
24     end
```

```
25
26 end
27
28
29 c_0_m = b_0_m;
30 c_0_n = b_0_n - 1;
31 img_mid(b_0_m, b_0_n) = 1;
32
33 b_m = b_0_m;
34 b_n = b_0_n;
35 c_m = c_0_m;
36 c_n = c_0_n;
37 cb_m = 0;
38 cb_n = 0;
39 b_2_m = 0;
40 b_2_n = 0;
41
42 while (b_2_m ~= b_0_m || b_2_n ~= b_0_n)
43
44     if (img_new(c_m, c_n) ~= 1)
45
46         if (c_m == b_m && c_n == b_n - 1 && img_new(c_m, c_n) ~= 1)
47             cb_m = c_m;
48             cb_n = c_n;
49             c_m = c_m - 1;
50         end
51
52         if (c_m == b_m - 1 && c_n == b_n - 1 && img_new(c_m, c_n) ~= 1)
53             cb_m = c_m;
54             cb_n = c_n;
55             c_n = c_n + 1;
56         end
57
58         if (c_m == b_m - 1 && c_n == b_n && img_new(c_m, c_n) ~= 1)
59             cb_m = c_m;
60             cb_n = c_n;
61             c_n = c_n + 1;
62         end
63
64         if (c_m == b_m - 1 && c_n == b_n && img_new(c_m, c_n) ~= 1)
65             cb_m = c_m;
66             cb_n = c_n;
67             c_n = c_n + 1;
68         end
69
70         if (c_m == b_m - 1 && c_n == b_n + 1 && img_new(c_m, c_n) ~= 1)
71             cb_m = c_m;
72             cb_n = c_n;
73             c_m = c_m + 1;
74         end
75     end
76 end
```

```
75
76     if (c_m == b_m && c_n == b_n + 1 && img_new(c_m, c_n) ~= 1)
77         cb_m = c_m;
78         cb_n = c_n;
79         c_m = c_m + 1;
80     end
81
82     if (c_m == b_m + 1 && c_n == b_n + 1 && img_new(c_m, c_n) ~= 1)
83         cb_m = c_m;
84         cb_n = c_n;
85         c_n = c_n - 1;
86     end
87
88     if (c_m == b_m + 1 && c_n == b_n && img_new(c_m, c_n) ~= 1)
89         cb_m = c_m;
90         cb_n = c_n;
91         c_n = c_n - 1;
92     end
93
94     if (c_m == b_m + 1 && c_n == b_n - 1 && img_new(c_m, c_n) ~= 1)
95         cb_m = c_m;
96         cb_n = c_n;
97         c_m = c_m - 1;
98     end
99
100 end
101
102 if (img_new(c_m, c_n) == 1)
103     b_m = c_m;
104     b_n = c_n;
105     c_m = cb_m;
106     c_n = cb_n;
107     b_2_m = b_m;
108     b_2_n = b_n;
109     img_mid(b_m, b_n) = 1;
110 end
111
112 end
113
114 img_out = img_mid(2:m + 1, 2:n + 1);
115
116 figure;
117 subplot(1, 2, 1)
118     imshow(img)
119     title('原始图像', fontsize = 18)
120 subplot(1, 2, 2)
121     imshow(img_out);
122     title('输出的边界', fontsize = 18)
```

#### 四、 实验结果

实验结果如下：



#### 五、 总结

本次实验主要是通过 Matlab 编程语言实现了课程中所讲过的 Moore 算法跟踪边界原理。通过本次实验我对 Moore 算法图像边界跟踪更深的理解。