

浙江大学

本科实验报告

Proj11-01: Boundary following

课程名称: 数字图像处理

姓 名: 叶慷鹏

学 院: 信息与工程学院

专 业: 信息工程

学 号: 3190102932

指导老师: 李东晓

2022 年 5 月 30 日

浙江大学实验报告

专业： 信息工程
姓名： 叶慷鹏
学号： 3190102932
日期： 2022 年 5 月 30 日
地点： _____

课程名称： 数字图像处理 指导老师： 李东晓 成绩： _____
实验名称： Proj11-01: Boundary following 实验类型： 设计仿真 同组学生姓名： _____

一、 实验任务

(a) 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出来。

(b) 下载图 9.14 来应用算法

二、 算法设计

本次设计的边界跟踪算法是基于 Moore 算法的，其具体过程如下。

(1) 找到一个黑色像素，并将它定为你的起始像素。(定位一个起始像素可以以多种方式来完成的；我们将从网格的左下角开始，自下而上扫描每一列像素，从最左向右的每列像素，直到遇到一个黑色的像素，我们将其作为我们的起始像素)。

(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白色像素就是目标的边界像素。

三、 代码实现

具体细节如下。

基于 Moore 算法的边界跟踪程序

```
1  img=imread("1.tif");
2  figure;
3  imshow(img);
4  title('原图')
5  %首先找出初始点（最左上角那个）
6  [row,col]=size(img);
7  img_large=zeros(row+2,col+2);%外部加一圈0值
8  img_large(2:row+1,2:col+1)=img;
9  img_end=zeros(row+2,col+2);%用来记录边界的过渡图
10 b0_r=0;
11 b0_c=0;
12 for i = 2:1:row+1
```

```
13     for j = 2:1:col+1
14         if (img_large(i,j-1)==0&&img_large(i-1,j)==0&&img_large(i,j)==1)
15             b0_r=i;
16             b0_c=j;
17             break
18         end
19     end
20 end
21 %找到初始点设为b0,再设置c0
22 c0_r=b0_r;
23 c0_c=b0_c-1;
24 img_end(b0_r,b0_c)=1;
25 %设置循环中的b, c以及用来判断两次到达起点而停止循环的b2
26 b_r=b0_r;
27 b_c=b0_c;
28 c_r=c0_r;
29 c_c=c0_c;
30 cb_r=0;%这个cb用来记录上一个c的位置
31 cb_c=0;%
32 b2_r=0;%先初始化为0值,防止一开始循环就结束
33 b2_c=0;
34 while (b2_r~=b0_r || b2_c~=b0_c)%b2与初始b0重合时循环停止
35     if(img_large(c_r,c_c)~=1)%说明c还没遍历到边界,还需继续遍历,下面根据c的位置进行判断走向
36         if(c_r==b_r&&c_c==b_c-1&&img_large(c_r,c_c)~=1)%说明c在b左边,应往上走
37             cb_r=c_r;%记录上一个c的位置方便回去
38             cb_c=c_c;
39             c_r=c_r-1;
40         end
41         if(c_r==b_r-1&&c_c==b_c-1&&img_large(c_r,c_c)~=1)%说明c在b左上边,应往右走
42             cb_r=c_r;%记录上一个c的位置方便回去
43             cb_c=c_c;
44             c_c=c_c+1;
45         end
46         if(c_r==b_r-1&&c_c==b_c&&img_large(c_r,c_c)~=1)%说明c在b上边,应往右走
47             cb_r=c_r;%记录上一个c的位置方便回去
48             cb_c=c_c;
49             c_c=c_c+1;
50         end
51         if(c_r==b_r-1&&c_c==b_c&&img_large(c_r,c_c)~=1)%说明c在b上边,应往右走
52             cb_r=c_r;%记录上一个c的位置方便回去
53             cb_c=c_c;
54             c_c=c_c+1;
55         end
56         if(c_r==b_r-1&&c_c==b_c+1&&img_large(c_r,c_c)~=1)%说明c在b右上角,应往下走
57             cb_r=c_r;%记录上一个c的位置方便回去
58             cb_c=c_c;
59             c_r=c_r+1;
60         end
61         if(c_r==b_r&&c_c==b_c+1&&img_large(c_r,c_c)~=1)%说明c在b右边,应往下走
62             cb_r=c_r;%记录上一个c的位置方便回去
```

```
63         cb_c=c_c;
64         c_r=c_r+1;
65     end
66     if(c_r==b_r+1&&cb_c==b_c+1&&img_large(c_r,c_c)~=1)%说明c在b右下, 应往左走
67         cb_r=c_r;%记录上一个c的位置方便回去
68         cb_c=c_c;
69         c_c=c_c-1;
70     end
71     if(c_r==b_r+1&&c_c==b_c&&img_large(c_r,c_c)~=1)%说明c在b下, 应往左走
72         cb_r=c_r;%记录上一个c的位置方便回去
73         cb_c=c_c;
74         c_c=c_c-1;
75     end
76     if(c_r==b_r+1&&c_c==b_c-1&&img_large(c_r,c_c)~=1)%说明c在b左下角, 应往上走
77         cb_r=c_r;%记录上一个c的位置方便回去
78         cb_c=c_c;
79         c_r=c_r-1;
80     end
81 end
82 if(img_large(c_r,c_c)==1) %说明c点遍历到了一个边界
83     b_r=c_r;%b变成c
84     b_c=c_c;
85     c_r=cb_r;%c变成上一个c
86     c_c=cb_c;
87     b2_r=b_r;
88     b2_c=b_c;
89     img_end(b_r,b_c)=1;
90 end
91 end
92 img_out=img_end(2:row+1,2:col+1);
93 figure;
94 imshow(img_out);
95 title('输出的边界')
```

四、 实验结果

原图



图 1: result

应用程序输出的边界:

输出的边界

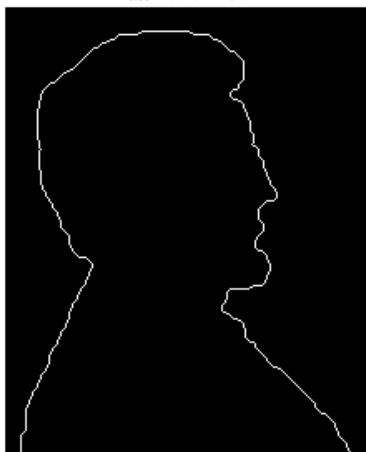


图 2: result

上图为原图，下图为根据 Moore 算法跟踪并输出的边界，显然边界跟踪效果良好，说明我们的程序设计成功。

五、 实验总结

本次实验主要采用了 MATLAB 里的 while 循环和一些内置函数来实现 c 点的遍历。实验主要考察对 Moore 算法跟踪边界的原理。通过本次实验我对 Moore 算法图像边界跟踪更深的理解,同时也对其设计和应用有了一定的掌握。