

数字图像处理

Pro11-1 Boundary following

信息与电子工程学院 信息工程

2023 年 6 月 10 日

1 实验任务

- (1) 实现边界跟随算法，确保输出边界中的点按顺时针或逆时针顺序组织；
- (2) 下载图 9.14，用设计的算法得到其边界；

2 算法设计

算法步骤：

1. 遍历图像；
2. 标记第一个遇见像素块的前景像素 (i, j) ；
3. 对这个像素周围八邻域逆时针搜索，如果搜索到周围有前景像素，那么更新坐标 (i, j) 为 (i', j') ，并标记；
4. 不断执行第 3 步直到再次遇见此像素块第一次标记的像素；
5. 继续执行第 1 步；

3 代码实现

```
clear all;
close all;
clc;

img=imread('fig.tif');
[m n]=size(img);

imgn=zeros(m,n);           % 边界标记图像
ed=[-1 -1;0 -1;1 -1;1 0;1 1;0 1;-1 1;-1 0]; % 从左上角像素，逆时针搜索
for i=2:m-1
for j=2:n-1
    if img(i,j)==1 && imgn(i,j)==0           % 当前是没标记的白色像素
        if sum(sum(img(i-1:i+1,j-1:j+1)))~=9 % 块内部的白像素不标记
```

```

ii=i;          % 像素块内部搜寻使用的坐标
jj=j;
imgn(i,j)=2;   % 本像素块第一个标记的边界，第一个边界像素为 2

while imgn(ii,jj)~=2 % 是否沿着像素块搜寻一圈了。
    for k=1:8 % 逆时针八邻域搜索
        tmpi=ii+ed(k,1); % 八邻域临时坐标
        tmpj=jj+ed(k,2);
        if img(tmpi,tmpj)==1 && imgn(tmpi,tmpj)~=2 % 搜索到新边界，并且没有
            ii=tmpi; % 更新内部搜寻坐标，继续搜索
            jj=tmpj;
            imgn(ii,jj)=1; % 边界标记图像该像素标记，普通边界为 1
            break;
        end
    end
end

end
end
end
end
end
figure;
imgn=imgn>=1;
subplot(1,2,2);imshow(imgn,[]);title('边界')
subplot(1,2,1);imshow(img);title('原图')

```

4 实验结果

将算法应用与图 9.14 得到结果如下：



图 1: 边界跟踪

5 总结

本次实验实现了一个边界跟踪算法，其本质就是不断对目标像素做八邻域遍历的迭代算法，最终的实现效果也比较不错。