形态学

上次课程

- OSTU二值化算法
- 线扫描
- 旋转变换
- 临近插值法

本次课程

- 形态学腐蚀膨胀
- 形态学重构
- 连通域查找表
- 投影定位

形态学

腐蚀和膨胀是形态学中的基本方法。以下描述只对二值图像适应。

腐蚀记为 $A \ominus B$

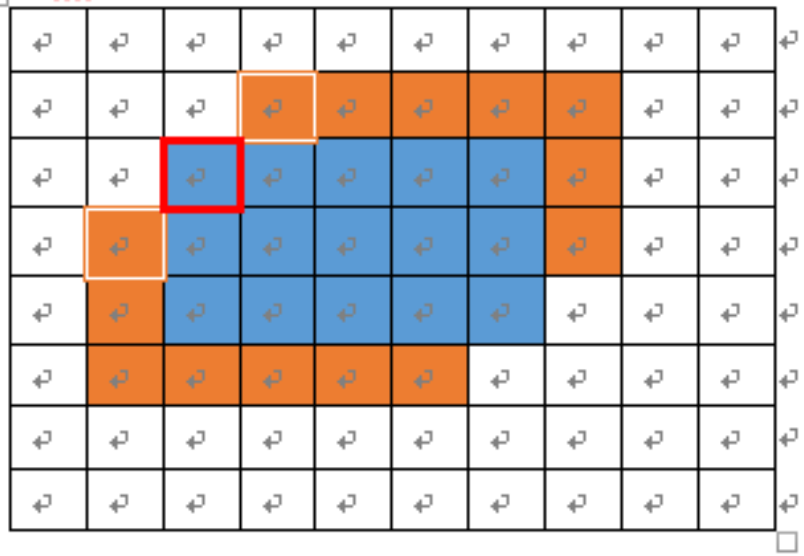
腐蚀是结构元素B与图像A上前景像素完全匹配的原点位置的集合。

膨胀记为$A \oplus B$

膨胀是结构元素B在图像A上的前景像素上平移后的集合。

膨胀

存在结构元素 B 和待膨胀的目标图像 A，膨胀结果是 B 的原点沿着 A 的像素点平移后的的集合。



如图所示，蓝色的是 A 的前景像素，橙色的是膨胀后的结果，红框是结构元素的原点。Ps：必须先确定结构元素 B 的原点。

Matlab实现

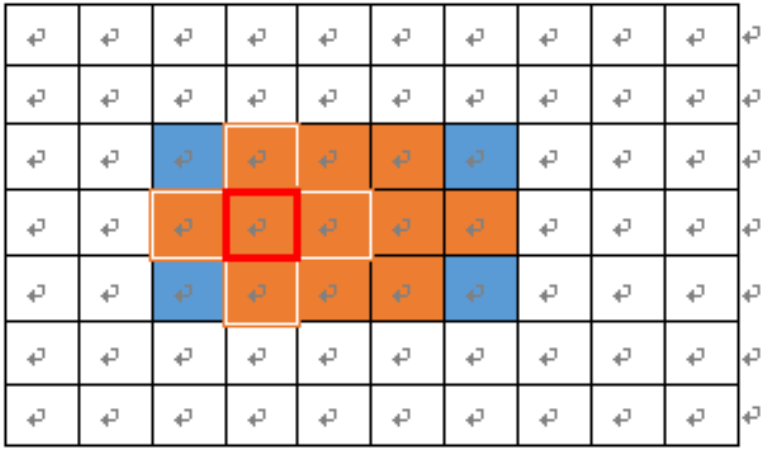
1. 建立一个与结构元素同样大小的窗口。
2. 滑动窗口，如果此窗口中原点位置的像素值不为0，则此点邻域赋值为结构元素。

```
for xxx
    遍历原图像的每一个点
% if  当前结构体原点不为0
    for xxx
        检查结构体的每一个点，若结构体中该点不为0，则将原点赋值为1；
    end
end
```

腐蚀

b) → 腐蚀算法

腐蚀算法可以看作是膨胀算法的对偶。方法是拿 B 的中心点和 A 上的点一个一个地对比，如果 B 上的所有点都在 A 的范围内，则该点保留，否则将该点去掉；右边是腐蚀后的结果。可以看出，它仍在原来 A 的范围内，且比 A 包含的点要少，就象 A 被腐蚀掉了一层。



如图所示，蓝色的是 A 的前景像素，橙色的是腐蚀后的结果，红框是结构元素的原点。

Matlab实现

1. 建立一个与结构元素同样大小的窗口。
2. 滑动窗口，如果此窗口中的像素与结构元素完全匹配，则窗口的原点的位置的像素值置1，否则置0。

for xxx

遍历原图像的每一个点

若窗口内所有点与结构元素点完全匹配

则将原点赋值为1

end

这样实现腐蚀会有一个问题，是什么呢？

形态学重构

- **重构**：是一种涉及到两幅图像和一个结构元素的形态学变换。一副图像，即标记（marker），是变换的开始点，用来约束变换经过的过程。结构元素用于定义连接性。重构的步骤如下
- 将h_1初始化为标记图像marker。
- 创建构造元素B=ones(3)
- 其中f∈g，g是掩膜mask（控制变换的限制区域）
- 重复 $h_{k+1} = h_k \oplus B \cap g$
- 直到 $h_{k+1} = h_k$
- f是标记，则从f重构g可以标记为$R_g(f)$ 。

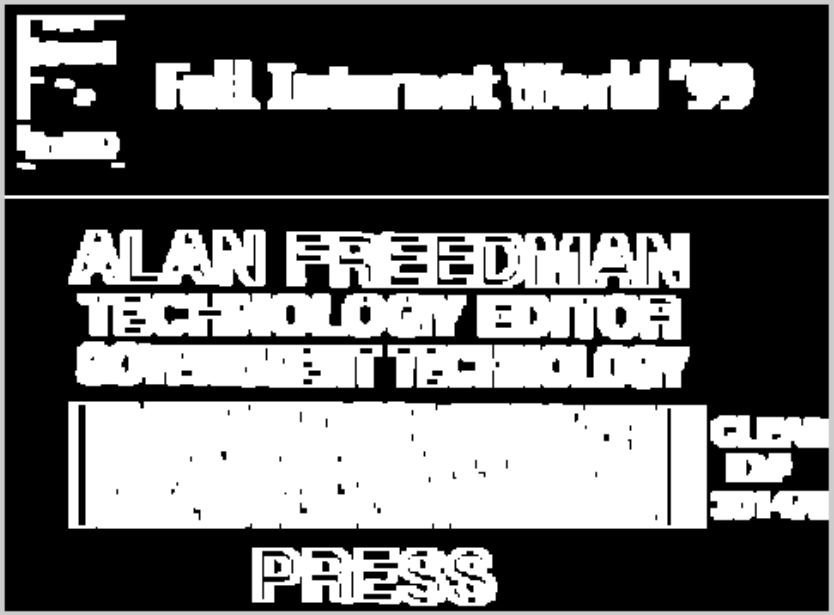
利用重构去除与边界相连的连通域

1. 把marker设定为原图像的边界值，掩膜设定为原图像。
2. 进行重构运算$h_{k+1} = h_k \oplus B \cap g$

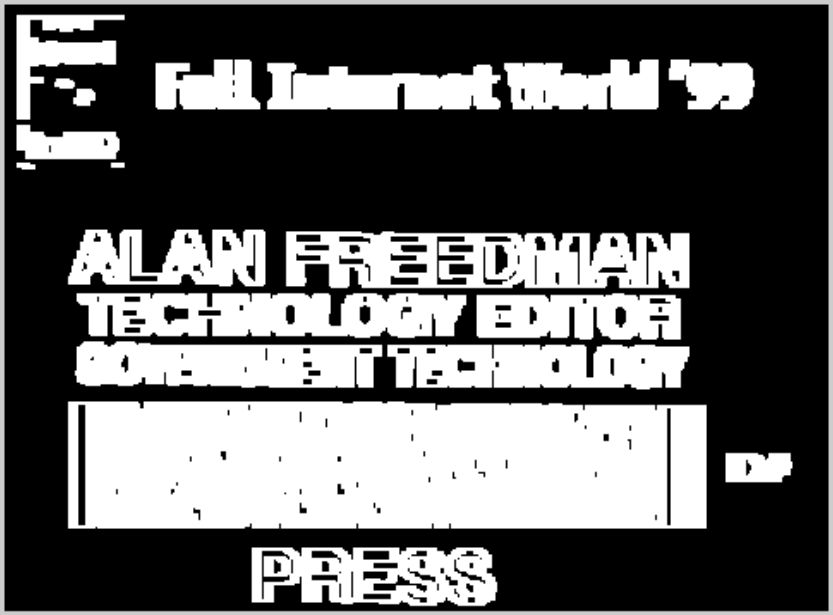
   while h_k ! = h_{k+1}

   进行运算$h_{k+1} = h_k \oplus B \cap g$

   end
3. 用原图像减去重构的结果



去掉与边界相连的连通域




Internet
World '99
Fall Internet World '99
ALAN FREEDMAN
TECHNOLOGY EDITOR
GOVERNMENT TECHNOLOGY
CLEAR
ID#
301479
PRESS

流程

1. 将图像转化为二值化的图像
2. 定义结构元素(记得定义结构体原点)、边缘算子
3. 对图像的边界进行扩展

   原图像大小为row*col ,结构元素大小为m*n，则扩展后大小为(row+2(m-1))*(col*2(n-1))，扩展值为0。
4. 提取边缘
5. 根据图像特点进行腐蚀和膨胀，最后去除边界上扩展的像素值。
6. 去除与边界相连的连通域
7. 建立查找表，去除小面积连通域
8. 投影定位、截取图像

建立查找表

1. 函数[L,num]=bwlabel(src);用于对二值图像src中的连通域进行标号。L是各连通域的编号，num是连通域的数量。
2. 计算各连通域的面积。

for 1:num

统计各个连通域的面积s(i)

end

1	1	1	0	0	0	0	0
1	1	1	0	1	1	0	0
1	1	1	0	1	1	0	0
1	1	1	0	0	0	1	0
1	1	1	0	0	0	1	0
1	1	1	0	0	0	1	0
1	1	1	0	0	1	1	0
1	1	1	0	0	0	0	0

1	1	1	0	0	0	0	0
1	1	1	0	2	2	0	0
1	1	1	0	2	2	0	0
1	1	1	0	0	0	3	0
1	1	1	0	0	0	3	0
1	1	1	0	0	0	3	0
1	1	1	0	0	3	3	0
1	1	1	0	0	0	0	0

3. 找到把最大连通域面积的一半作为阈值，去除小于阈值的连通域。

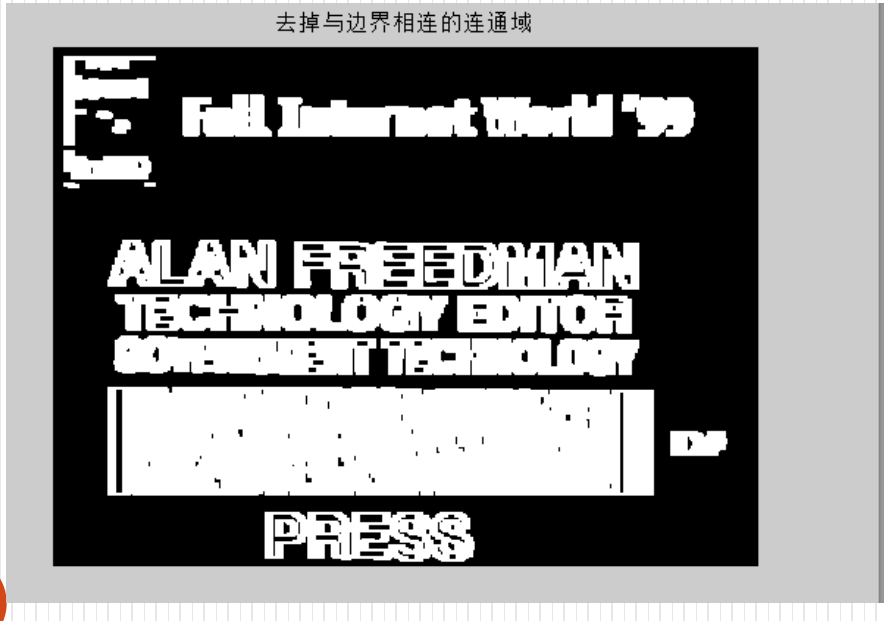
for 1:num

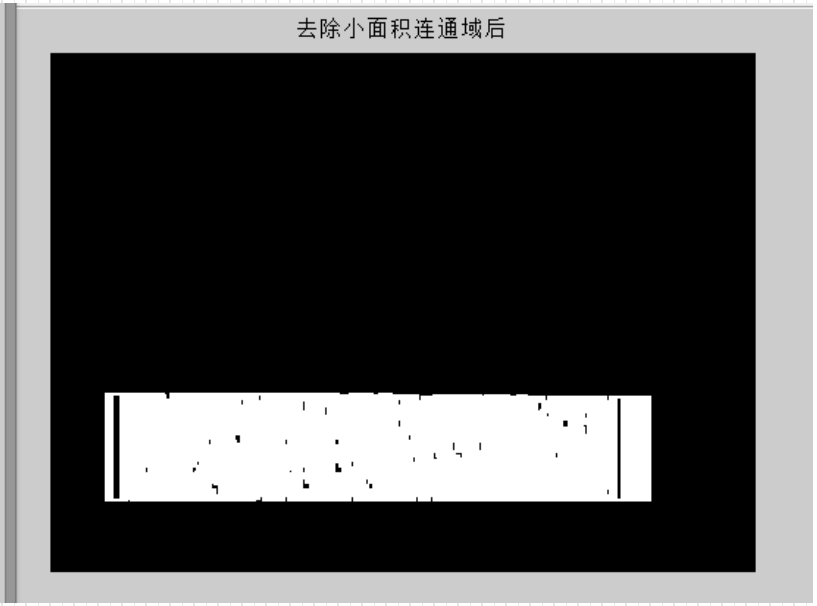
if s(i) < ½ * 最大值

把L==i的点均去除（赋值为0）

end

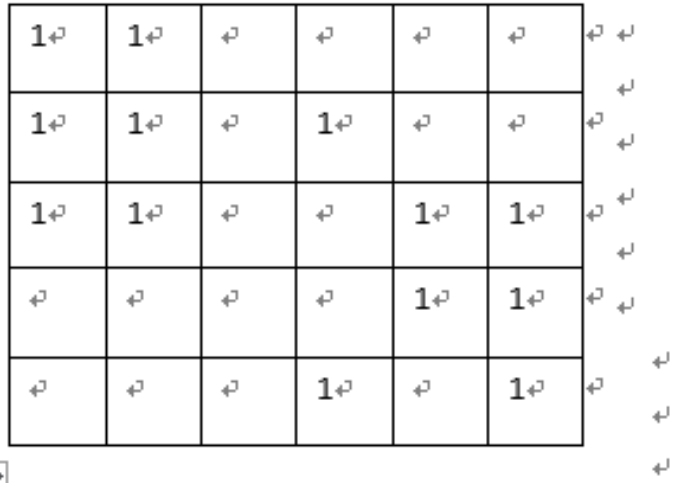
end

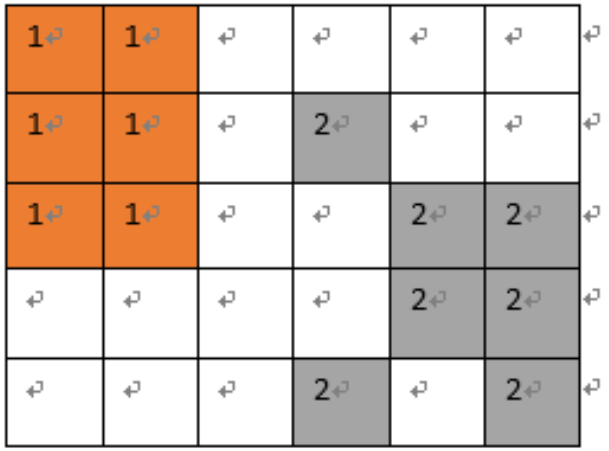




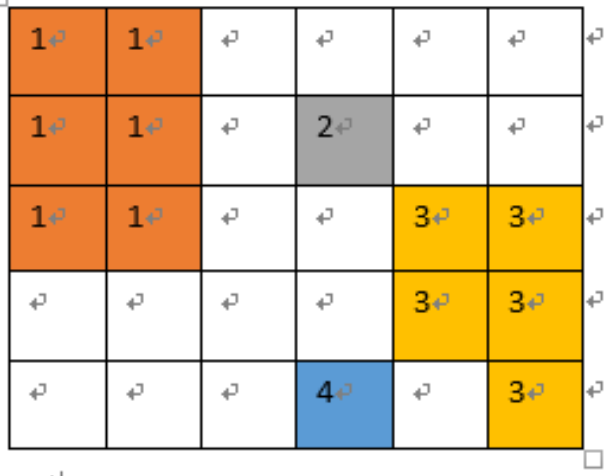
P.S. 连通域分为4连接和8连接

连通分量的获取和划分与选择的连接方式有关，如下图





8 连接



4 连接

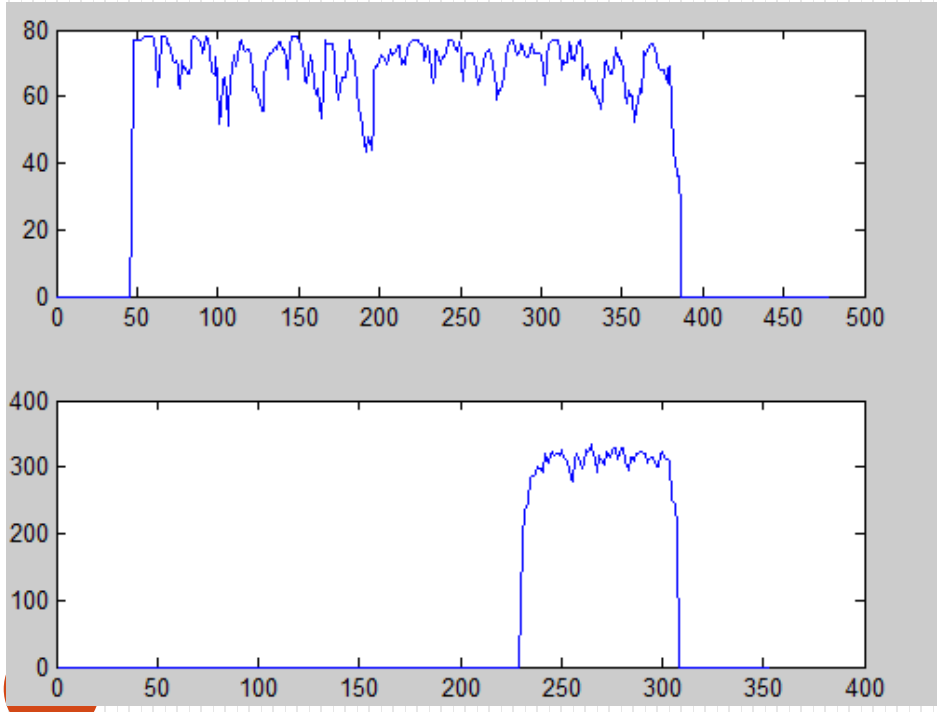
投影定位

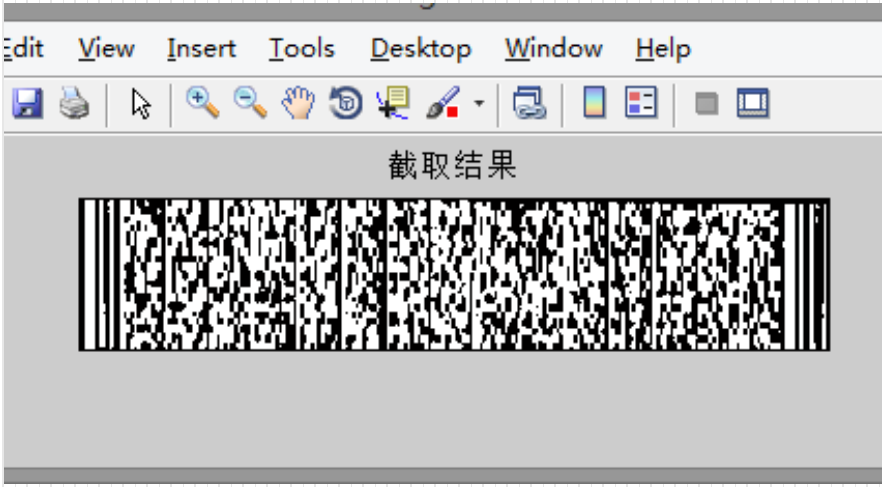
1. 统计每一行的前景像素的数量作为水平方向上的投影$S_y(y)$。
2. 统计每一列的前景像素的数量作为垂直方向上的投影$S_x(x)$ 。
3. 利用公式$S_k = 0.7S_k + 0.3S_{k+1}$ 进行平滑。
4. 把各方向上的投影值的平均值作为阈值。
5. 找到各方向上大于阈值的区间
6. 截取图像

ALAN FREEDMAN
TECHNOLOGY EDITOR
GOVERNMENT TECHNOLOGY

CLEAR ID# 301479

PRESS





迷人风景只有站在一定高度才能领略，望诸君努力，谢谢！