

数字图像分析第一次实验

实验原理和设计思路

1.文本图像的二值形态学处理

1.1长字符提取

1.2空洞填充

1.3边界清除

2.图像分割

3.粒度测定

4.纹理分割

代码说明

运行平台

提交目录

数字图像分析第一次实验

实验原理和设计思路

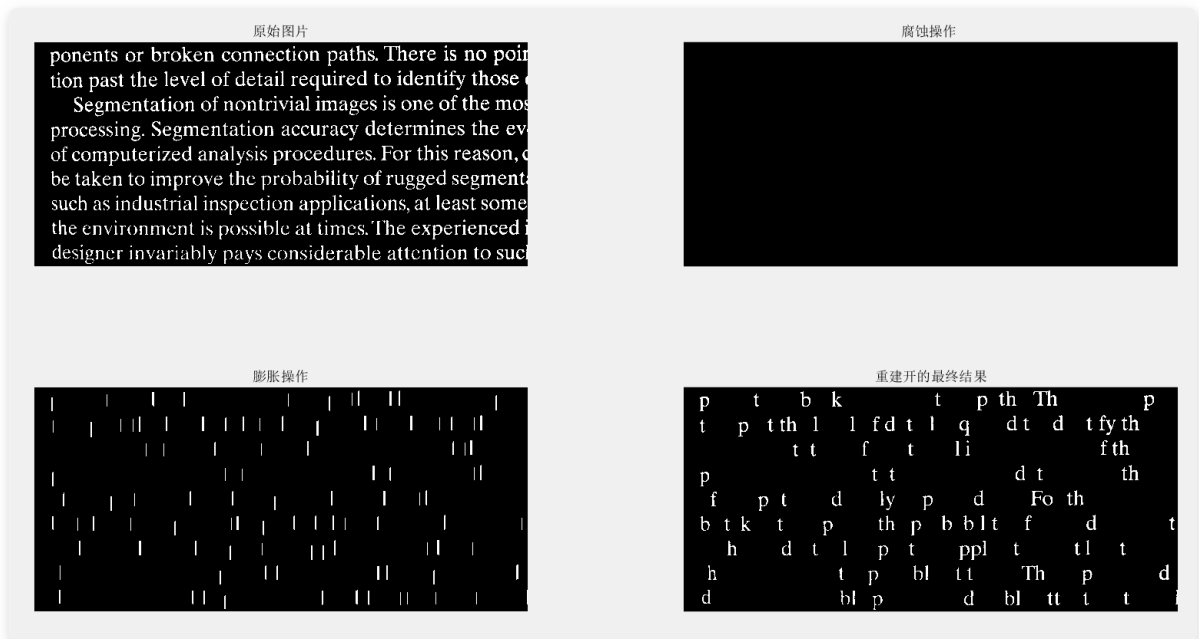
1.文本图像的二值形态学处理

1.1长字符提取

重建开操作: $O_R^{(n)}(F) = R_F^D[(F \ominus nB)]$

首先使用竖直方向存在连续长度为k(实验中k取为51)的模板E, 对原图像进行腐蚀, 随后再利用模板E对腐蚀后的图像进行膨胀, 从而找到竖直方向的长字符的竖直方向部分,最后使用 51×9 模板进行n次膨胀重建恢复字符, 直到无法继续膨胀。实验中 $n=21$ 即重建结束。

该功能见long_character.m文件。实验运行结果如下所示, 其中最终结果保留为output/1_长字符提取.tif文件



长字符提取程序输出

1.2孔洞填充

孔洞填充:令 $I(x, y)$ 代表一幅二值图像，并假定一个标记后的图像 F ，除了图像边界位置为 $1-I$ 之外，其他位置均为0，即有：

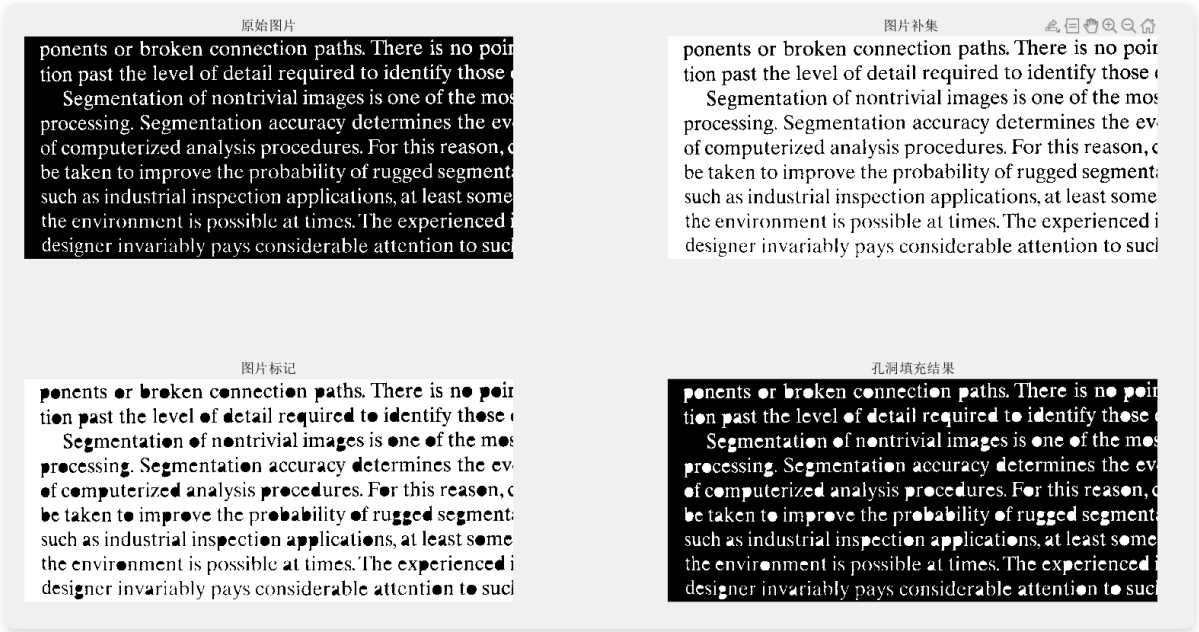
$$F(x, y) = \begin{cases} 1 - I(x, y) & (x, y) \text{ 在 } I \text{ 的边界上} \\ 0 & \text{其他} \end{cases}$$

则

$$H = [R_{I^c}^D(F)]^c \quad (1)$$

是一幅等于 I 且所有孔洞都被填充的二值图像。

该功能见fill_character.m文件。实验运行结果如下所示，其中最终结果保留为output/1_孔洞填充.tif文件



孔洞填充程序输出

1.3边界清除

边界清除:令 $I(x, y)$ 代表一幅二值图像，并假定一个标记后的图像 F ，除了图像边界位置为 I 之外，其他位置均为0，即有：

$$F(x, y) = \begin{cases} I(x, y) & (x, y) \text{ 在 } I \text{ 的边界上} \\ 0 & \text{其他} \end{cases}$$

边界清除首先计算形态学重建 $R_I^D(F)$ （简单地提取出边界的字符），最后从原图中减去边界字符，即

$$X = I - R_I^D(F) \quad (2)$$

是一幅等于 I 去除边界后的二值图像。

该功能见boundary_delete_character.m文件。实验运行结果如下所示，其中最终结果保留为output/1_边界清除.tif文件

边界



边界清除



ponents or broken connection paths. There is no position past the level of detail required to identify those
Segmentation of nontrivial images is one of the most difficult tasks in image processing. Segmentation accuracy determines the effectiveness of computerized analysis procedures. For this reason, great care must be taken to improve the probability of rugged segmentation. In some applications, such as industrial inspection applications, at least some degree of automation is possible at times. The experienced designer invariably pays considerable attention to success

边界清除程序输出

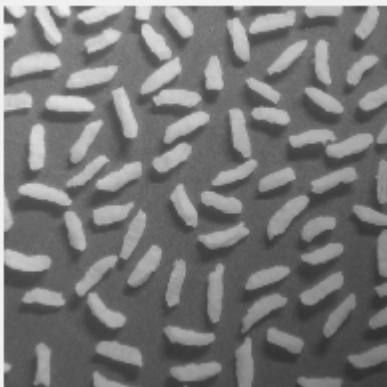
2.图像分割

顶帽变换: $T_{hat}(f) = f - (f \circ b)$

实验中采用Otsu最佳阈值处理方法得到，调用matlab自带的graythresh函数,采用半径为40的圆盘结构进行开操作，可见，经过顶帽变换后，采用Otsu最佳阈值处理方法，图片中所有的米粒都可以正确分割出来。

可运行main2.m文件查看，该主要功能见T_hat.m文件。T_hat.m运行结果如下所示。

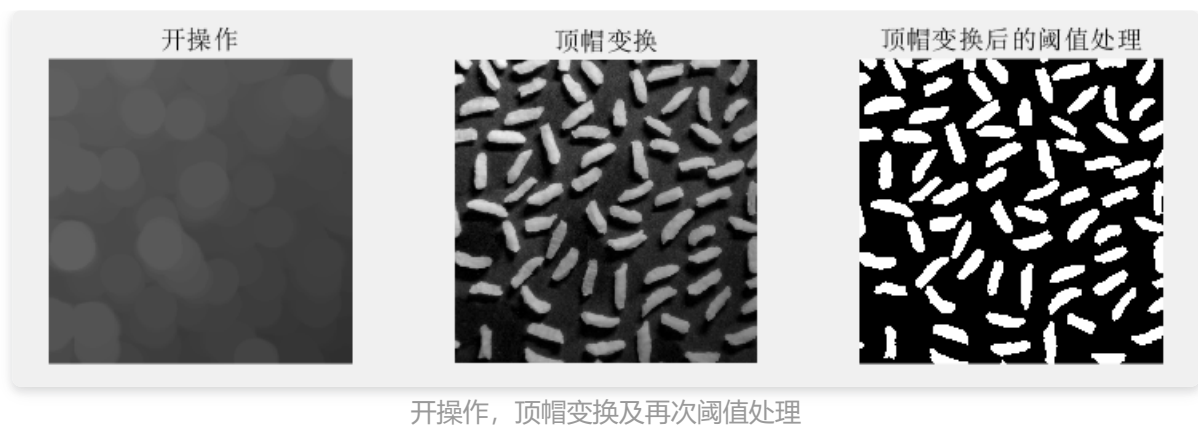
原始图像



对原始图像进行阈值处理



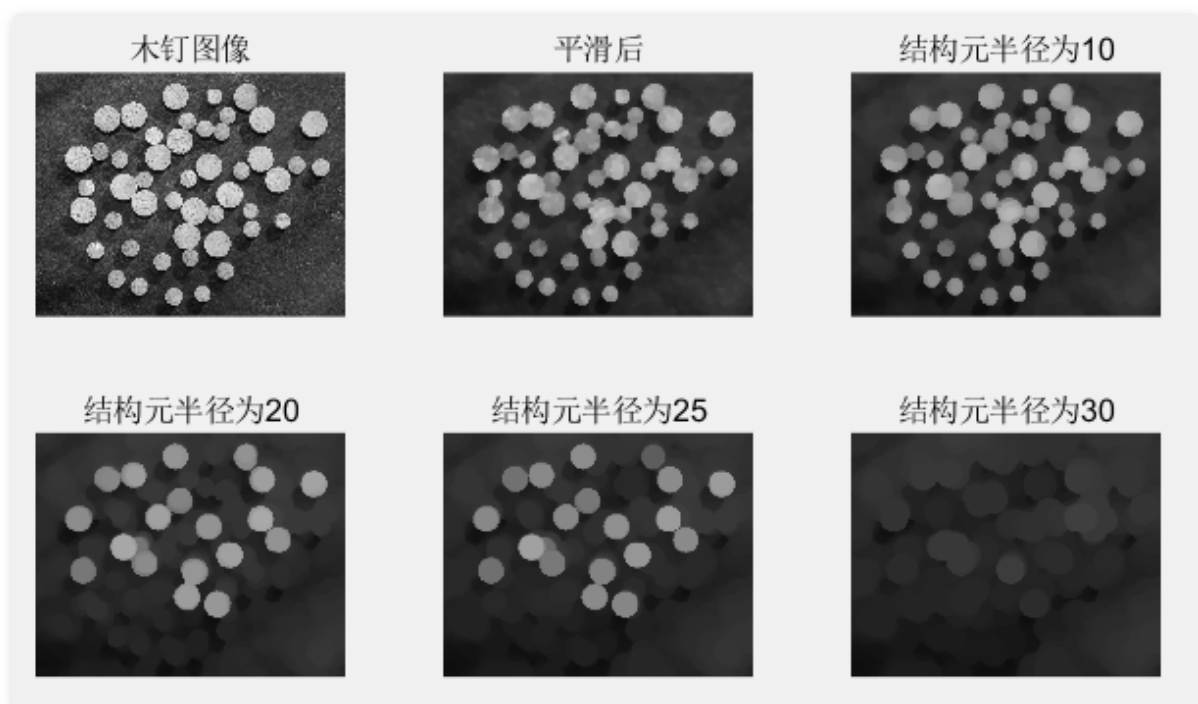
原始图像与阈值处理



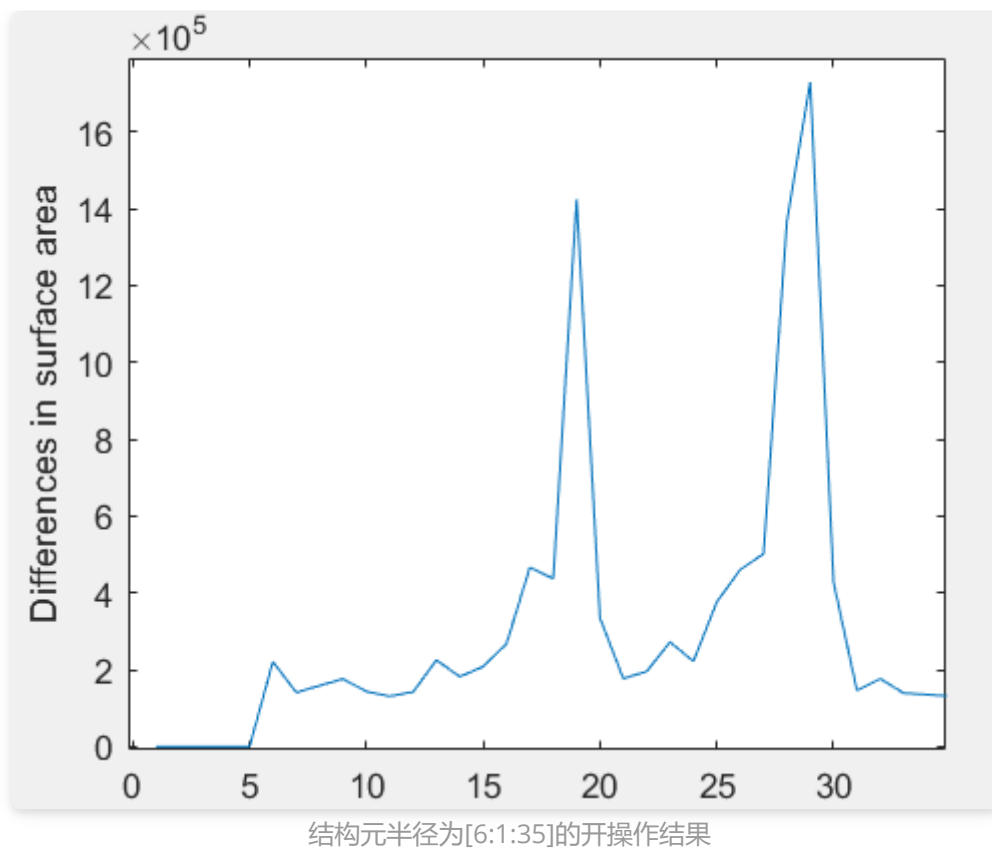
3.粒度测定

使用逐渐增大的结构元对图像执行开操作。某个特殊尺寸的开操作会对包含类似尺寸的颗粒输入图像的区域产生最大的效果。对于每一次开操作，计算出开操作中像素值的和，它会随着结构元的增大而减小。该过程会得到一个阵列，每个元素对应于阵列中该位置的结构元素的大小的开操作中的像素之和。从阵列中相邻元素的差，其峰值则反映了颗粒大小的分布。实验结果如下所示：

可运行main3.m文件查看，该主要功能见Granularity_detection.m文件。运行结果如下所示。



木钉图片，平滑图像，以及使用结构元半径为10,20,25,30的开操作结果

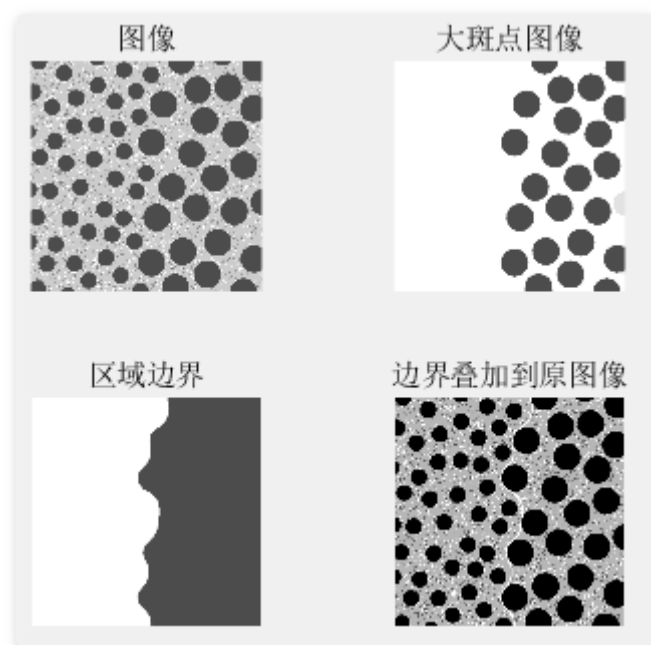


结构元半径为[6:1:35]的开操作结果

4.纹理分割

使用半径为30个像素的圆盘形结构元对输入图像进行闭操作的结点，删除了背景上的小斑点，然后使用半径为60个像素的圆盘形结构元对该图像进行开操作，删除大斑点之间的亮间隔区域，通过形态学梯度找到边界区域，最后将边界叠加到原图像中。

可运行main4.m文件查看，该主要功能见my_Gradient.m文件。运行结果如下所示。



纹理分割

代码说明

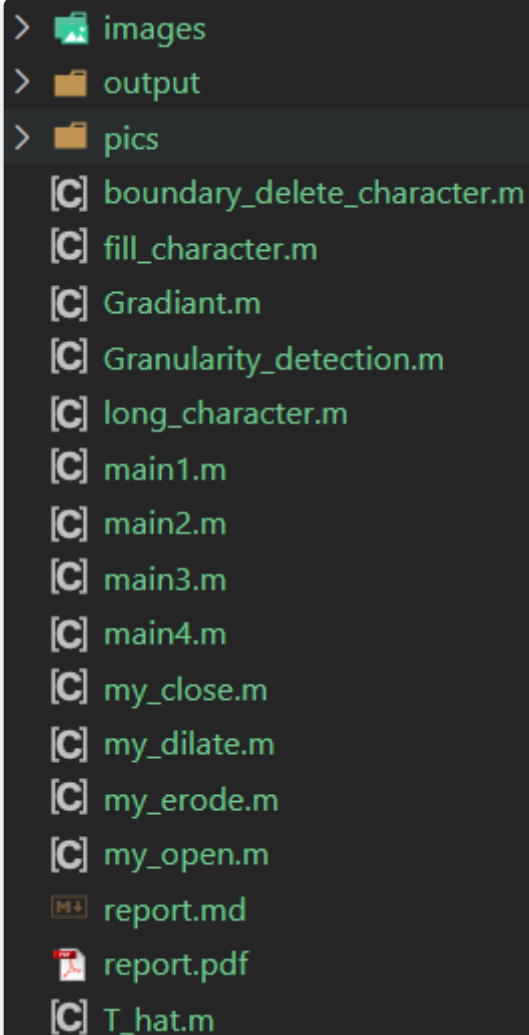
运行平台

Windows + Matlab(9.8.0.1323502 (R2020a))

提交目录

提交目录如下,现对主要文件做简要说明:

- output存储部分生成的最终图片,方便查看实验结果
- main1,mian2,mian3,main4分别是四个小问题的运行主程序
- my_close,my_open,my_erode,my_dilate为实现的腐蚀,膨胀,开启和闭合运算的函数
- boundary_delelte_character,fill_character,long_character为实验一的三个核心函数
- T_hat为实验二的核心函数
- Granularity_detection为实验三的核心函数
- Gradient为实验四的核心函数
- report.md和report.pdf为报告的md和pdf两个版本,可供自由查看



```
> images
> output
> pics
[C] boundary_delete_character.m
[C] fill_character.m
[C] Gradient.m
[C] Granularity_detection.m
[C] long_character.m
[C] main1.m
[C] main2.m
[C] main3.m
[C] main4.m
[C] my_close.m
[C] my_dilate.m
[C] my_erode.m
[C] my_open.m
[M] report.md
[PDF] report.pdf
[C] T_hat.m
```

提交目录