**《计算机视觉（1）》实验报告**

实验六 使用区域生长的分割

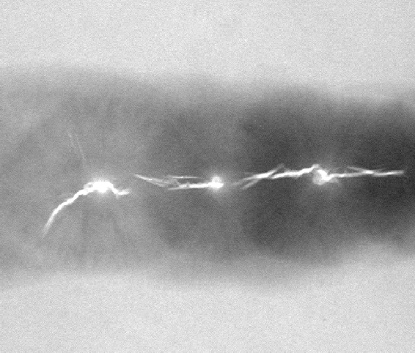
|  |  |  |
| --- | --- | --- |
| **实验小组成员**  **（学号+班级+姓名）** | **分工及主要完成任务** | **成绩** |
|  |  |  |
|  |  |  |
|  |  |  |

山东大学

2021年3月

完成《数字图像处理》P494页例10.23的编程实验，编程语言可以选择Matlab，C，C++，OpenCV，Python等。设计方案可参照教科书中的分析，也可以自行设计新的方案。

参照例10.23中“使用区域生长的分割”算法，对如下的有缺陷焊缝的X射线图像得到图10.51中(b)~(i)中的实验结果。



原始图像的电子版图像在Images文件夹中。实验报告写在如下空白处，页数不限。

实验六 区域生长分割算法

1. 实验目的

图1显示了一条焊缝（水平深色区域）的8比特X射线图像，该图像含有几条裂缝和孔隙（水平横穿图像中心的明亮区域）。我们希望用区域生长的方法将有缺陷的焊接区域分立出来。这些分割的特性可以应用与检测、历史研究的数据库，以及控制自动焊接系统等其他很多方面。在研究区域生长算法的过程中，本实验也同时要求利用形态学腐蚀、单阈值处理、多阈值处理方法对图像进行处理，以说明区域生长的处理效果。

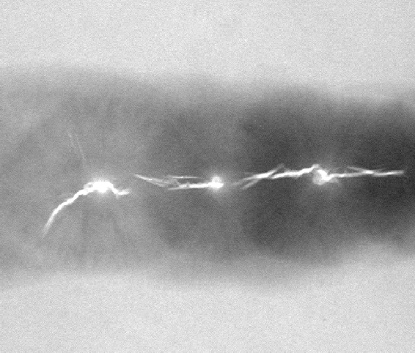


图1：焊缝（水平深色区域）的8比特X射线图像

1. 实验原理

区域生长是一种根据事先定义的准则将像素或子区域聚合成更大区域的过程，基本的方法是对目标区域要先指定一个种子点作为生长的起点，从该（组）种子点开始，将与种子性质相似的相邻像素附加到生长区域的每个种子上。直到没有满足条件的像素被包括进来为止。这样一个区域的生长就完成了。

**区域生长算法**：

令表示一个输入图像阵列：表示一个种子阵列，阵列中种子点位置处为1，其他位置处为0；表示在每个位置的属性。假设阵列和的尺寸相同。基于8连接的算法步骤如下，

1. 在中寻找所有连通分量，把每个连通分量腐蚀为一个像素；把所有找到的像素标记为1，把中其他像素标记为0.
2. 在坐标形成图像：如果输入图像在该坐标满足给定的属性，则令，否则令。
3. 令是这样的图像：把中为8连通种子点的所有1值点，添加到中的每个种子点集合。
4. 用不同区域标记标出中每个连通分量，即得到区域生长的分割图像。

**算法关键问题**：

1. 种子选择

通常我们基于问题的性质选择一组或多组开始点。当鲜艳之事不可用时，这一过程是在每个像素处计算相同特性的集合，最终在生长过程中，根据这些特性集合把刑诉分配给各个区域。如果这些计算的结果显示了一值，则那些特性靠近这些族重心的像素可以作为种子使用。

1. “生长”准则

相似性：“生长”像素点和种子像素需满足如灰度图像的差值；彩色图像的颜色等关于像素与像素间关系描述的相似关系。

连通性：“生长”像素点和种子像素徐具有连通属性。

1. 终止规则

当不再有像素满足假如某个区域的“生长”准则时，区域生长就会停止。

1. 实验软件

Python、OpenCV、PyQt5

1. 实验步骤与结果
2. 打开图片

打开原图像，利用plot.hist(img.ravel(),256,(0,255)，density=True)生成原图像的直方图，如图2所示。

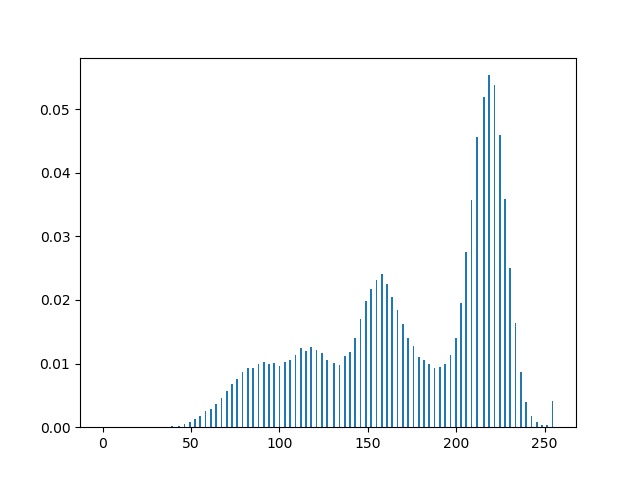


图2：原图像直方图

1. 生成种子

我们知道我们的目标区域要比X射线图像中其他部分明亮一些，因此可以再高百分比处设置阈值对原始图像进行阈值处理来提取种子点。我们采用图像中灰度值的99.5%的阈值对图像进行阈值处理，此时阈值为254。因此阈值处理效果等价于提取图像中所有像素为255的像素点。这里我们利用cv2.threshold()函数对原图像进行阈值处理。得到种子图像如图3所示。

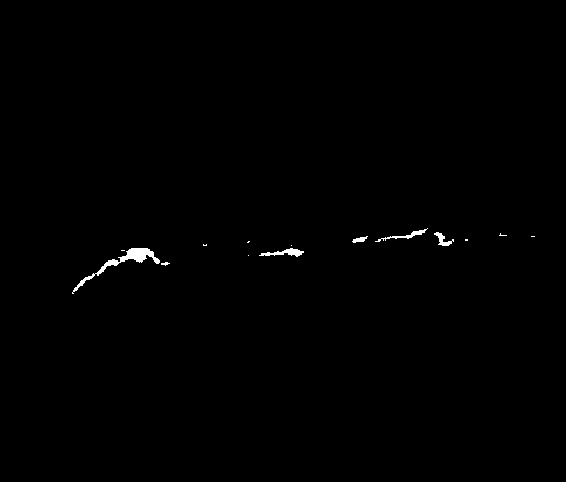


图3：种子图

考虑到运算量和运算时间，我们可以进一步缩减种子点集合的规模。因此我们考虑对种子图像提取连通分量，并将每个连通分量用形态学腐蚀为一个单点，最终得到一个元素数等于连通分量个数（19）的种子点集合。

其中提取连通分量算法涉及到图像的膨胀和交运算，这里分别利用cv2.dilate()和cv2.bitwise\_and()函数实现。最终种子腐蚀图像如图4所示。

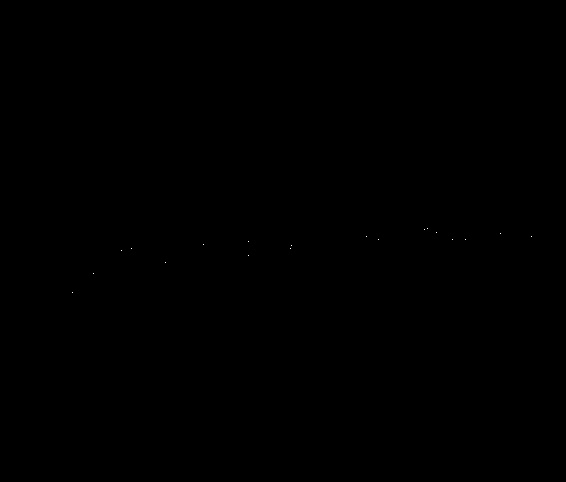


图4：种子腐蚀图（放大）

1. 阈值处理（用于对照）

我们先将原图像的灰度图像进行黑白反转如图5所示。利用相同的方法得到反转后图像的直方图，如图6所示。

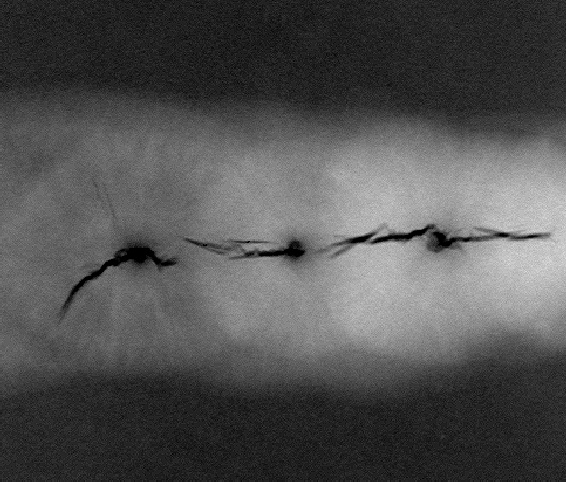


图5：黑白反转图

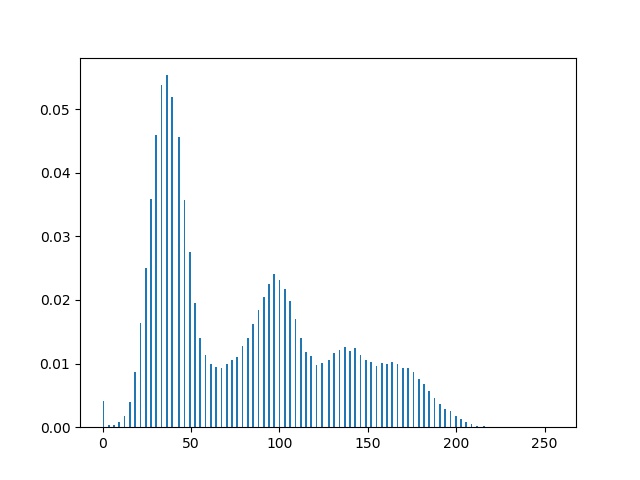


图6：黑白反转图直方图

我们需要一个阈值以便于建立相似性属性，观察到直方图有三个主要模式（波峰），因此我们首先考虑利用双阈值处理图像。这里我们利用自编写的双阈值处理函数对图像进行处理（阈值处理原理不在此赘述）。最终得到阈值分别为，双阈值处理图像如图7所示。

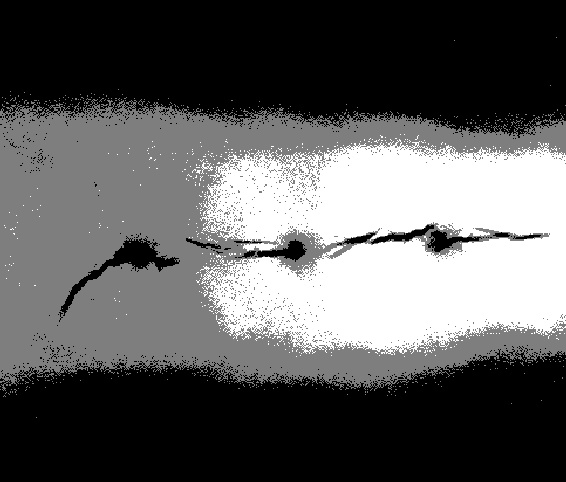


图7：基于双阈值处理的图像分割图

上图显示了双阈值处理不能解决缺陷的分割问题，尽管阈值处在主要波谷中。因此我们进一步使用单阈值处理图像，我们尝试分别仅使用和的单阈值对图像进行处理，这里我们使用cv2.threshold()实现。最终得到效果如图8所示。

通过下图可看出时效果明显较好，故采取作为最终单阈值。

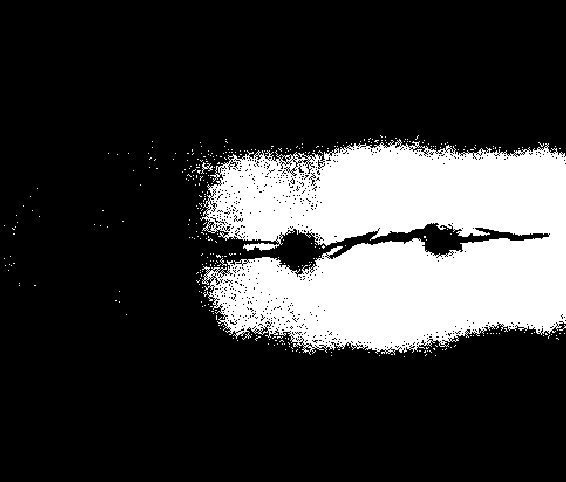
 

图8：基于单阈值处理的图像分割图（左：T=68/右：T=126）

1. 区域生长

上述阈值处理主要是为了与最终的区域生长效果进行对照。在“生成种子”步骤中，我们已经获取到了目标区域中每个连通分量内的种子点，接下来我们需要确定“生长准侧”和“终止规则”。本实验中我们使用灰度差作为一种相似性度量，则应用于每一个位置(x,y)的属性是：

其中是一个基于灰度差的指定的阈值，由于本实验中，种子点像素为255，因此图5的黑白反转图实质上也可看作原图像与种子图像的差值绝对值图，图6直方图则反映了差值图像的性质。再结合阈值处理过程，我们通过实践得出了分割效果更好的阈值，这为我们区域生长算法中阈值的选取起到了指引作用，因此我们选取相同的阈值实现区域生长算法，图像分割效果如图9所示。



图9：基于区域生长的图像分割图

进一步尝试小幅度调整阈值后，我们观察到实验结果几乎无差别，因此在此不对阈值选取进行更详细的分析讨论。

1. 难点分析
2. 种子点选取

法一：在种子图中将连通分量腐蚀单点的过程中，由于腐蚀本身原理，在有限次循环迭代后，图像所有像素点的灰度值将变为0（黑色）。因此为达到将连通分量腐蚀为单点的目的，我们可以记录腐蚀为全黑时的循环次数，可以认为其上一次循环中，连通分量内仅剩的一个非零灰度点即为腐蚀的目标点。

法二：相较于腐蚀方法，我们也尝试了直接从每个连通分量中提取任意一个像素的方法。此方法虽无原理支撑，但比起法一更加快捷。

通过实验证明，我们发现法二的效果不弱于法一，甚至基于法二的区域生长图像分割效果略优于基于法一的分割效果。因此我们在本实验中更倾向于法二。

1. 区域生长阈值选取

实验前期对于区域生长的阈值选取没有头绪，后发觉图5（黑白反转图）也可理解为原图像与种子像素点灰度值的差值图像，于是进一步理解了实验中对差值图像进行阈值处理的目的，一方面是为了对比不同方法分割图像的效果，另一方面也是为区域生长的阈值选取提供理论支撑。因此我们最终很快确定了区域生长的灰度差阈值为左右。

1. GUI封装

使用GUI界面对程序进行封装，如图10所示。运行所有功能后生成图象的界面如图11所示。

根据实验要求，GUI界面中可以呈现原图像、原图像直方图、种子图、种子腐蚀图、差值（黑白反转）图、差值图直方图、单阈值处理图、双阈值处理图、区域生长图。

GUI下方四个按钮分别可以实现“实验步骤及结果”中的四个主要步骤的对应功能。按钮右侧输入框中可设置区域生长算法的阈值。



图10：GUI界面

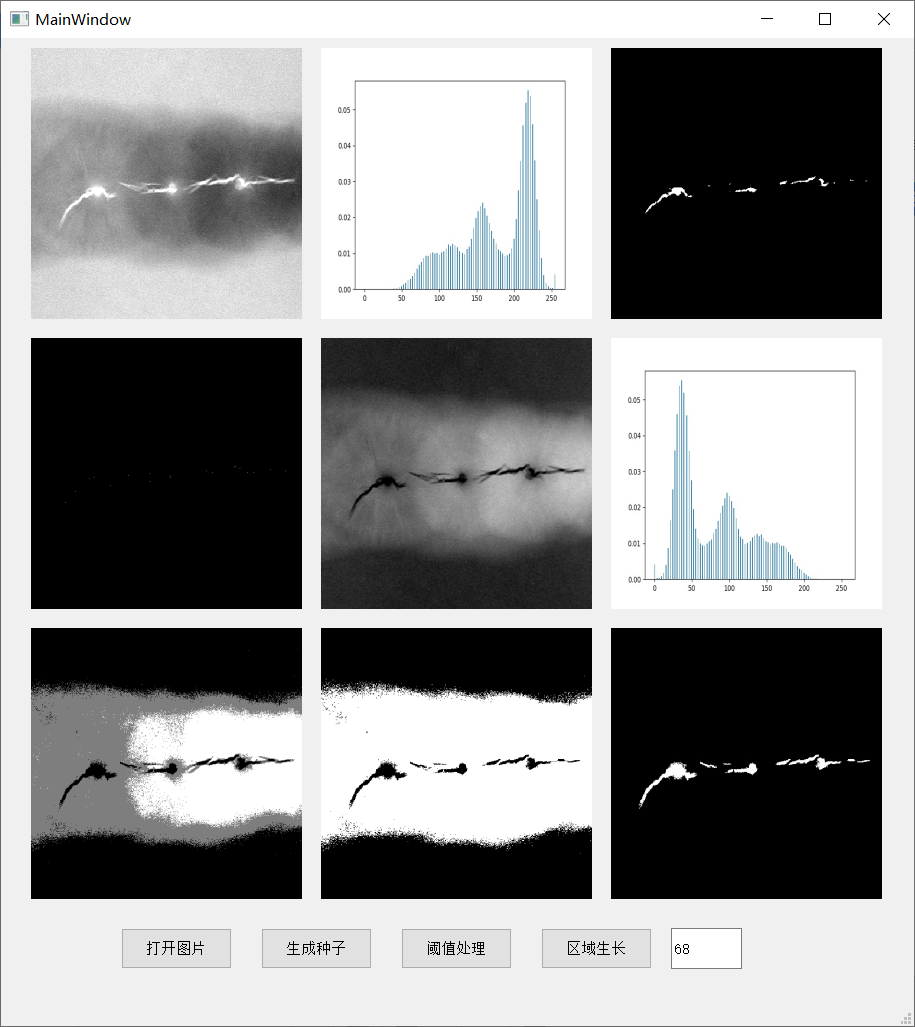


图11：GUI运行后界面

1. 反思感悟

通过本实验，对图像形态学处理原理、阈值处理方法、区域生长分割算法等有了更进一步了解，进一步提升了图像处理相关的代码编程能力。