**《计算机视觉（1）》实验报告**

实验二 使用直方图统计的局部增强

|  |  |  |
| --- | --- | --- |
| **实验小组成员**  **（学号+班级+姓名）** | **分工及主要完成任务** | **成绩** |
| 201800820045  18数据科学 许函嘉 | 全部独立完成 |  |
|  |  |  |
|  |  |  |

山东大学

2021年3月

完成《数字图像处理》P87页例3.12的编程实验，编程语言可以选择Matlab，C，C++，OpenCV，Python等。如图显示了一根绕在支架上的钨丝的SEM(扫描电子显微镜)图像。图像中央的钨丝及其支架很清楚并很容易分析。在该图像的右侧即图像的暗侧，有另一根钨丝的结构，但几乎不能察觉到，其大小和特征当然也不容易辨认。通过对比度操作进行局部增强是解决这种图像中包含部分隐含特征问题的理想方法。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。

实验完成主要任务：

1）参照教材给出的基于局部直方图统计矩（均值及方差）的亮度及对比度增强方案，编写实现图像全局直方图、局部直方图统计程序、基于直方图的全局均值及方差计算程序、局部方差及方差计算程序、选择性局部图像增强程序；

2）通过实验，探讨增强关键参数选择方法，包括参数𝐸，𝐾0，𝐾1，𝐾2以及局部区域𝑆𝑥y大小，分析改变这些参数对增强图像质量的影响规律。

原始图像

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。

实验二报告

1. 实验目的

如图显示了一根绕在支架上的钨丝的SEM(扫描电子显微镜)图像。图像中央的钨丝及其支架很清楚并很容易分析。在该图像的右侧即图像的暗侧，有另一根钨丝的结构，但几乎不能察觉到，其大小和特征当然也不容易辨认。通过对比度操作进行局部增强是解决这种图像中包含部分隐含特征问题的理想方法。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。

1. 实验任务

1）参照教材给出的基于局部直方图统计矩（均值及方差）的亮度及对比度增强方案，编写实现图像全局直方图、局部直方图统计程序、基于直方图的全局均值及方差计算程序、局部方差及方差计算程序、选择性局部图像增强程序；

2）通过实验，探讨增强关键参数选择方法，包括参数𝐸，𝐾0，𝐾1，𝐾2以及局部区域𝑆𝑥y大小，分析改变这些参数对增强图像质量的影响规律。

1. 需求分析

1）基本模块需求

包括了基本的图像打开存储，操作界面的搭建，图像像素级别处理，日志记录等。于是需要相关模块，包括pyQt5，opencv等。

2）基本算法需求

根据实验要求，算法主要内容在于对图像的像素进行处理、对像素进行直方图统计，因此需要熟练运用嵌套循环对图像数组进行操作，了解直方图的原理和制作方法。

3）基本UI需求

根据实验所要求的功能需求，最基本的需要设计全局直方图、局部直方图统计结果显示界面（主要包括直方图和数值结果）、选择性局部图像增强界面（主要包括参数设置和图像结果显示）。在实验要求基础上，我额外设计了全局图像均衡化的功能和界面。

1. 实验过程
2. UI设计
3. UI效果

在实验一的基础上，利用TabWidget组件，补充了实验二的相关界面和组件，并进一步优化了实验一的界面。总体效果如下：



1. 组件说明（TabWidget中自左而右、自上而下）

**全局直方图**：按钮。单击后对打开的图片进行全局直方图的绘制以及均值和方差的计算。

**全局均衡增强**：按钮。单击后对打开的图片进行全局均衡化，并进行全局直方图的绘制以及均值和方差的计算。

**局部图像增强**：按钮。设置完E、Sxy、K0、K1、K2参数后，单击对图片进行局部图像增强，并进行全局直方图的绘制以及均值和方差的计算。

**局部直方图**：按钮。设置完“中心点”、“窗口尺寸”参数后，单击实现对选中窗口的图像显示，并计算该窗口中图像的直方图、均值方差。

**参数设置（左）**：输入框。用户输入“局部图像增强”功能的相关参数。

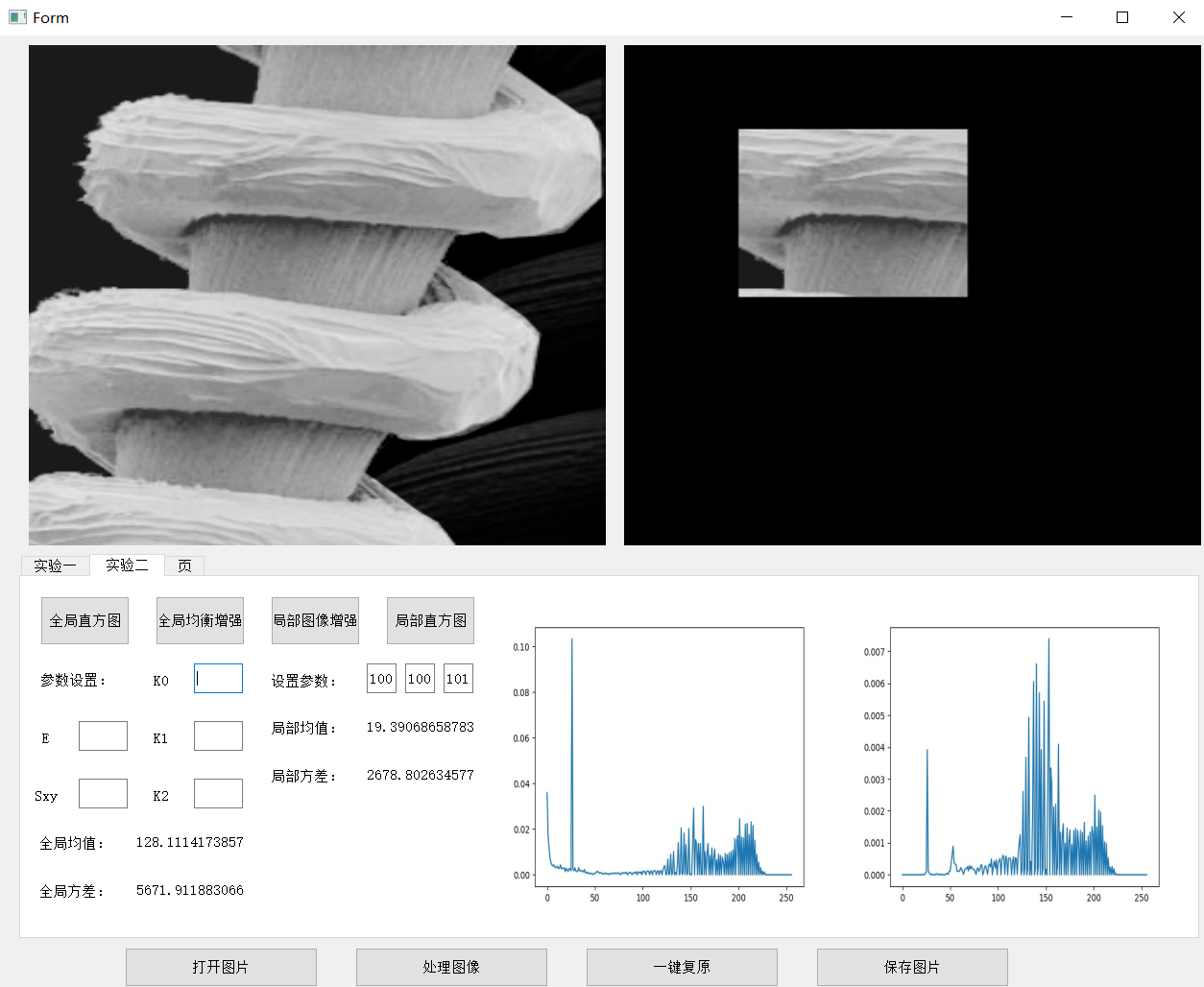
**参数设置（右）**：输入框。用户输入“局部直方图”功能的相关参数。

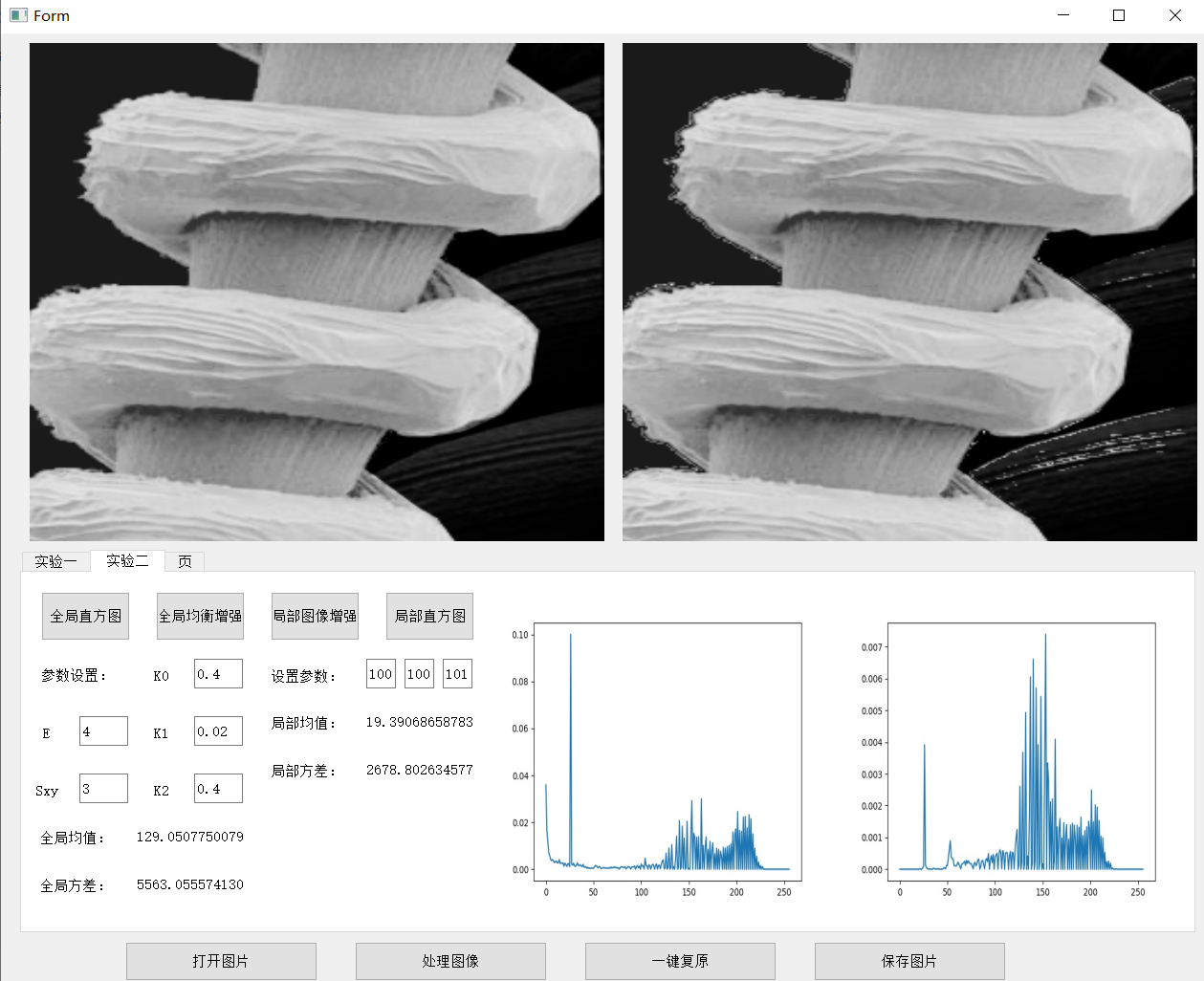
**全局均值、方差**：显示全局图像的图像灰度的均值方差。

**局部均值、方差**：显示窗口图像的图像灰度的均值方差。

**全局直方图**：显示全局图像的图像灰度的直方图。

**局部直方图**：显示窗口图像的图像灰度的直方图。





1. 问题和思路
2. 关于TabWidget的使用

在实验一后对UI界面进行了大程度的修改，主要体现在使用了TabWidget组件，使得实验集成UI界面更加美观。

1. 关于参数设置

依旧采取输入框设计，使参数设置的自由度更大。

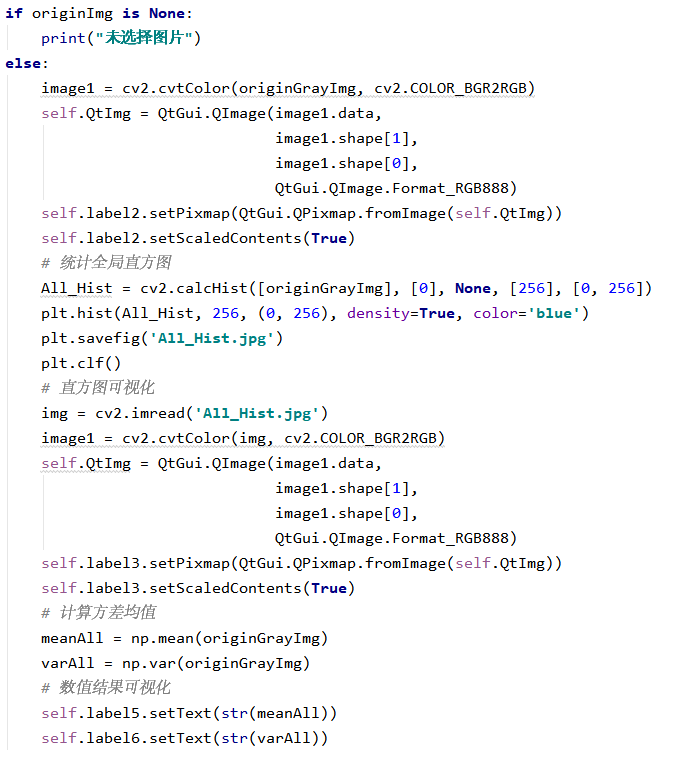
1. 关于直方图及统计结果的可视化

针对全局和局部分别设置了两个直方图显示窗口和均值方差显示列表。方便用户使用时对比全局和局部的信息差异，更直观。

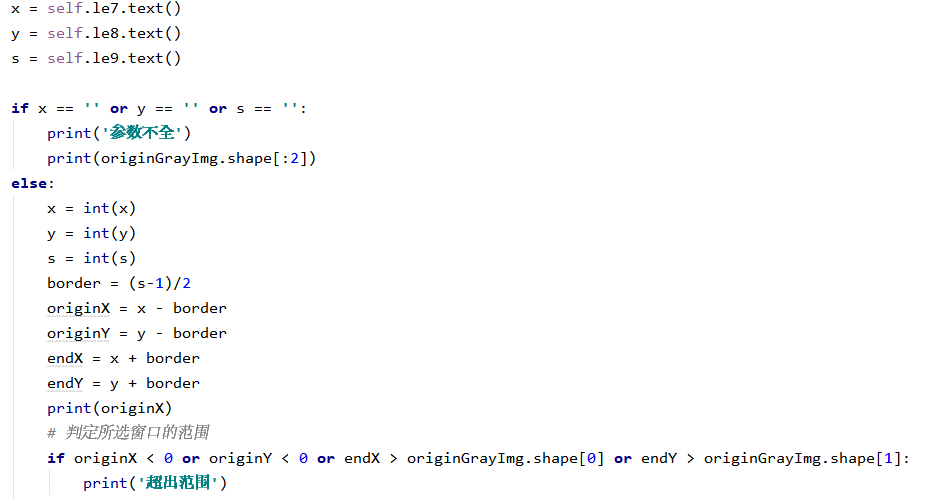
1. 关于处理后图像的显示

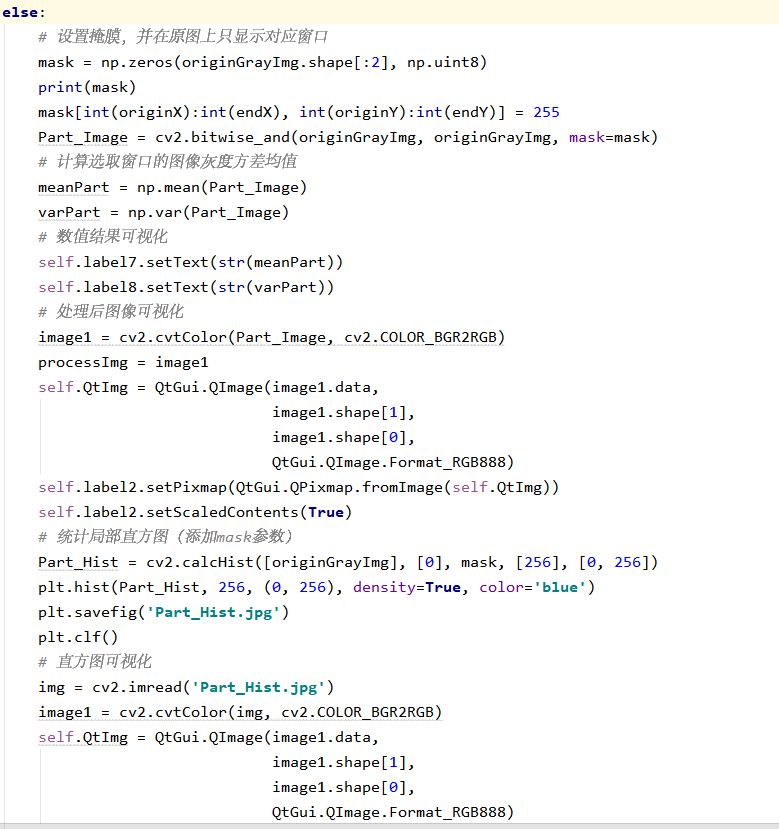
依然保留两个图像显示窗口，分别呈现原始图像和处理后图像。本实验中，处理后图像包括了“全局均衡处理图”（要求之外）、“局部图像增强图”、“局部图像窗口图”三种，将分别在点击相应功能按钮后显示。

1. 算法设计（核心代码）
2. 全局直方图和基于直方图的全局均值及方差计算程序

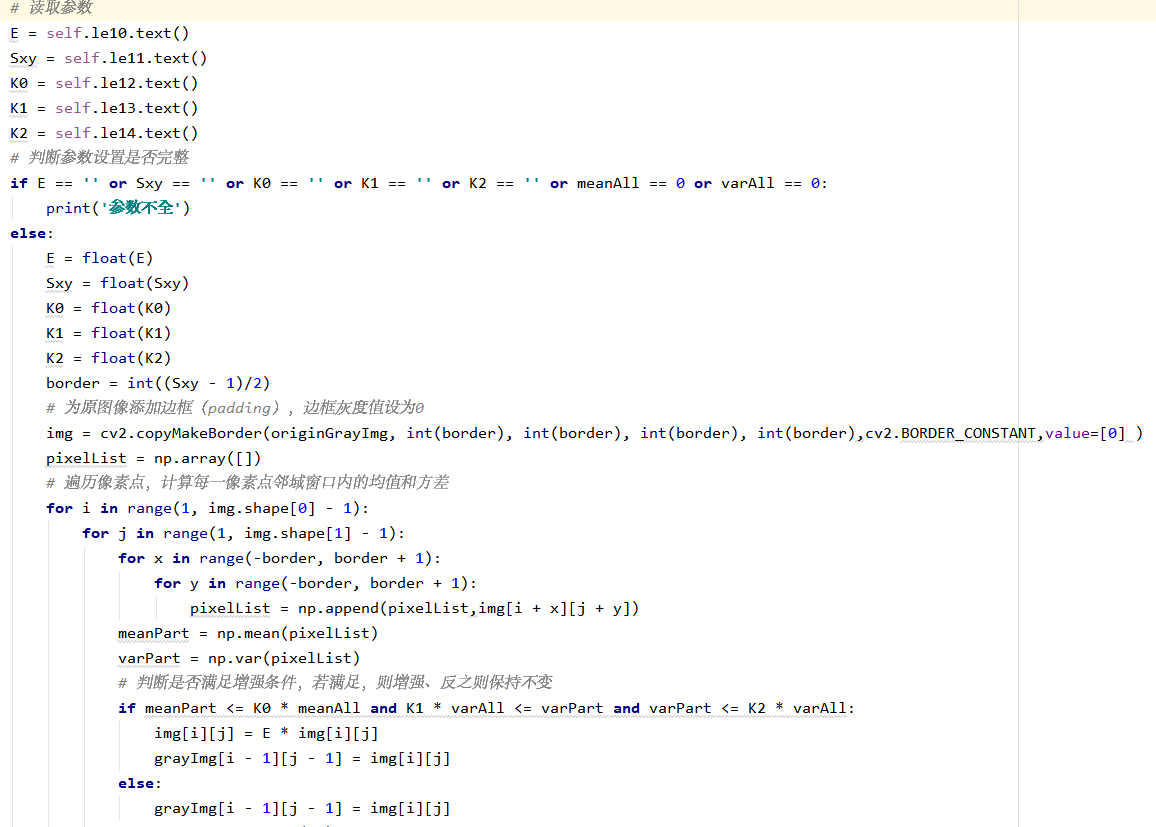


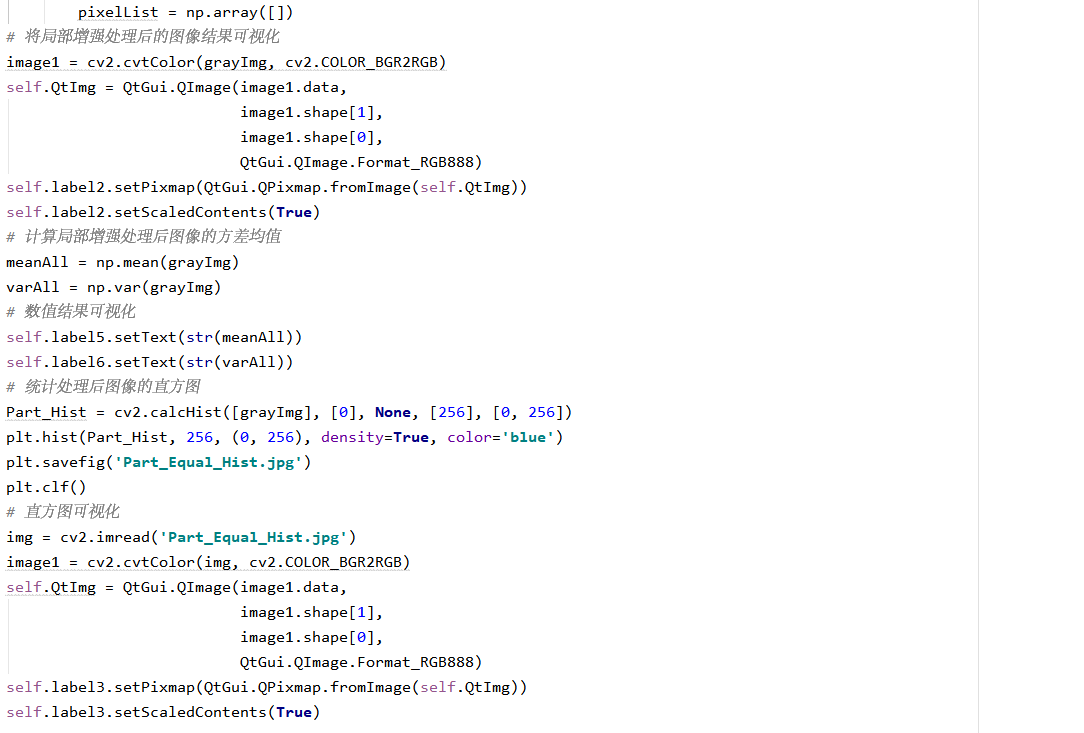
1. 局部直方图和基于直方图的局部均值及方差计算程序



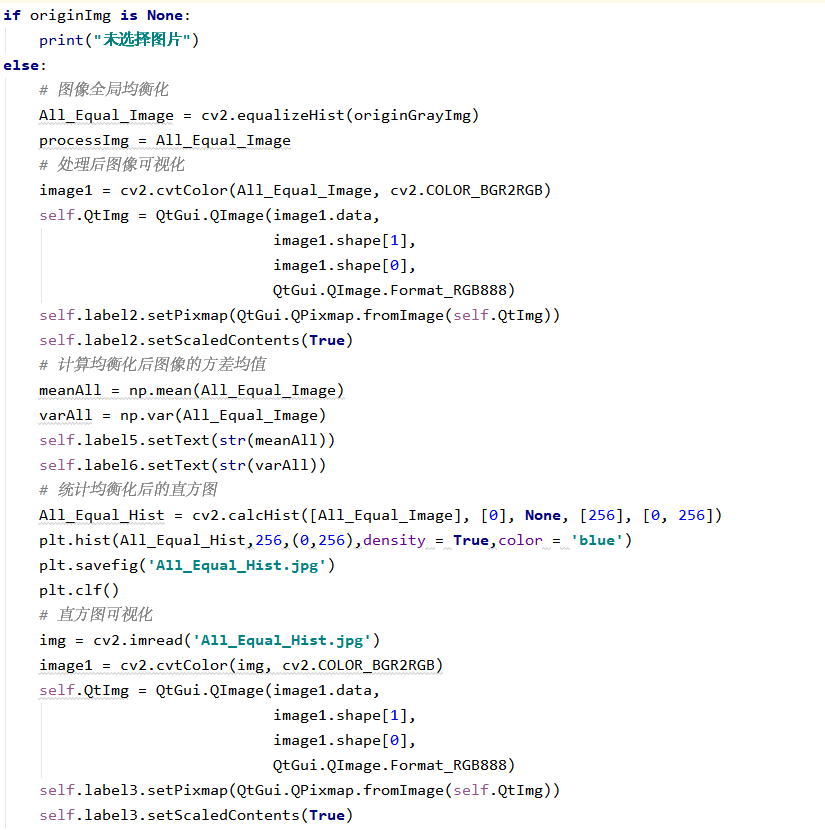


1. 选择性局部图像增强程序

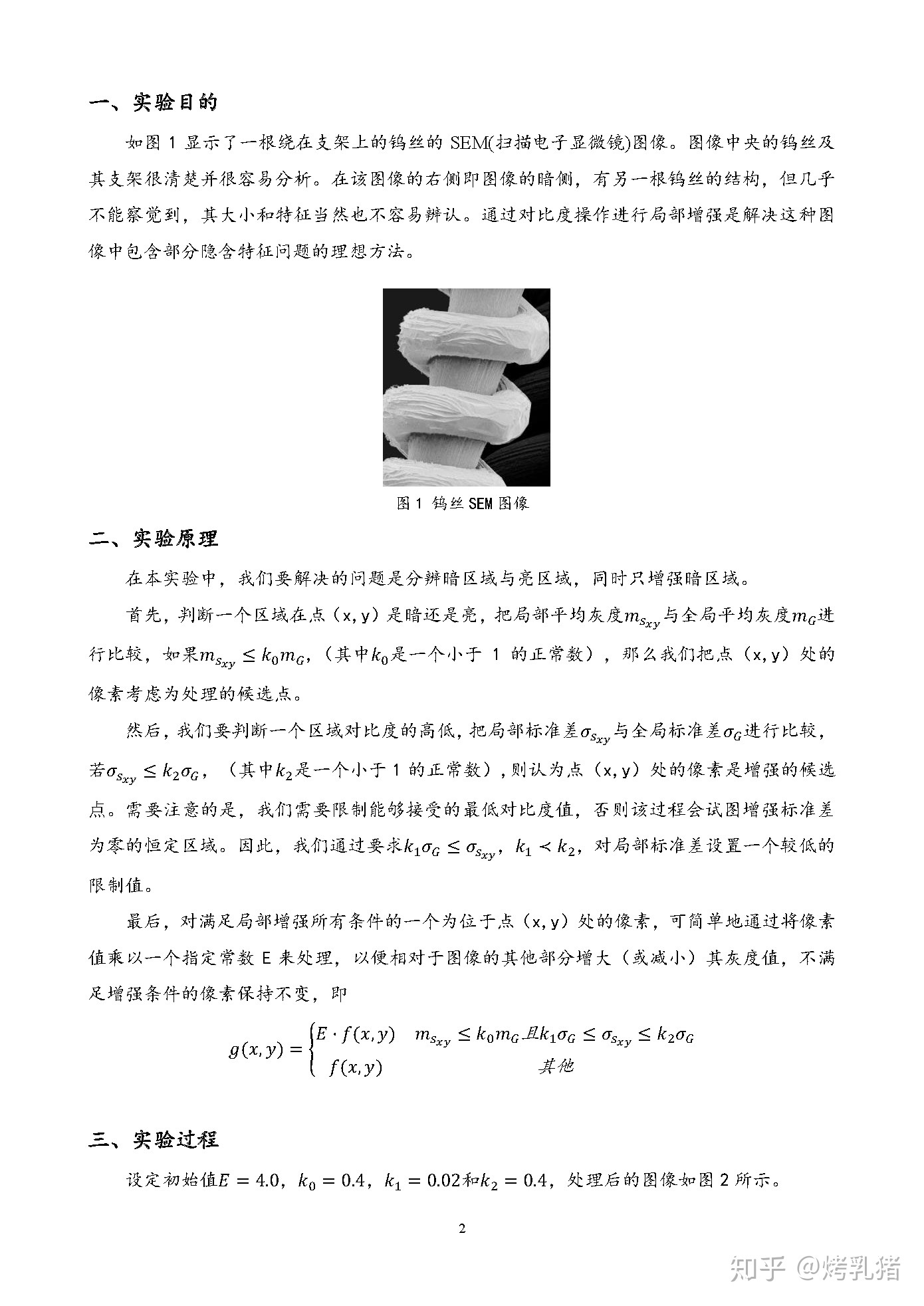




1. 全局均衡化处理

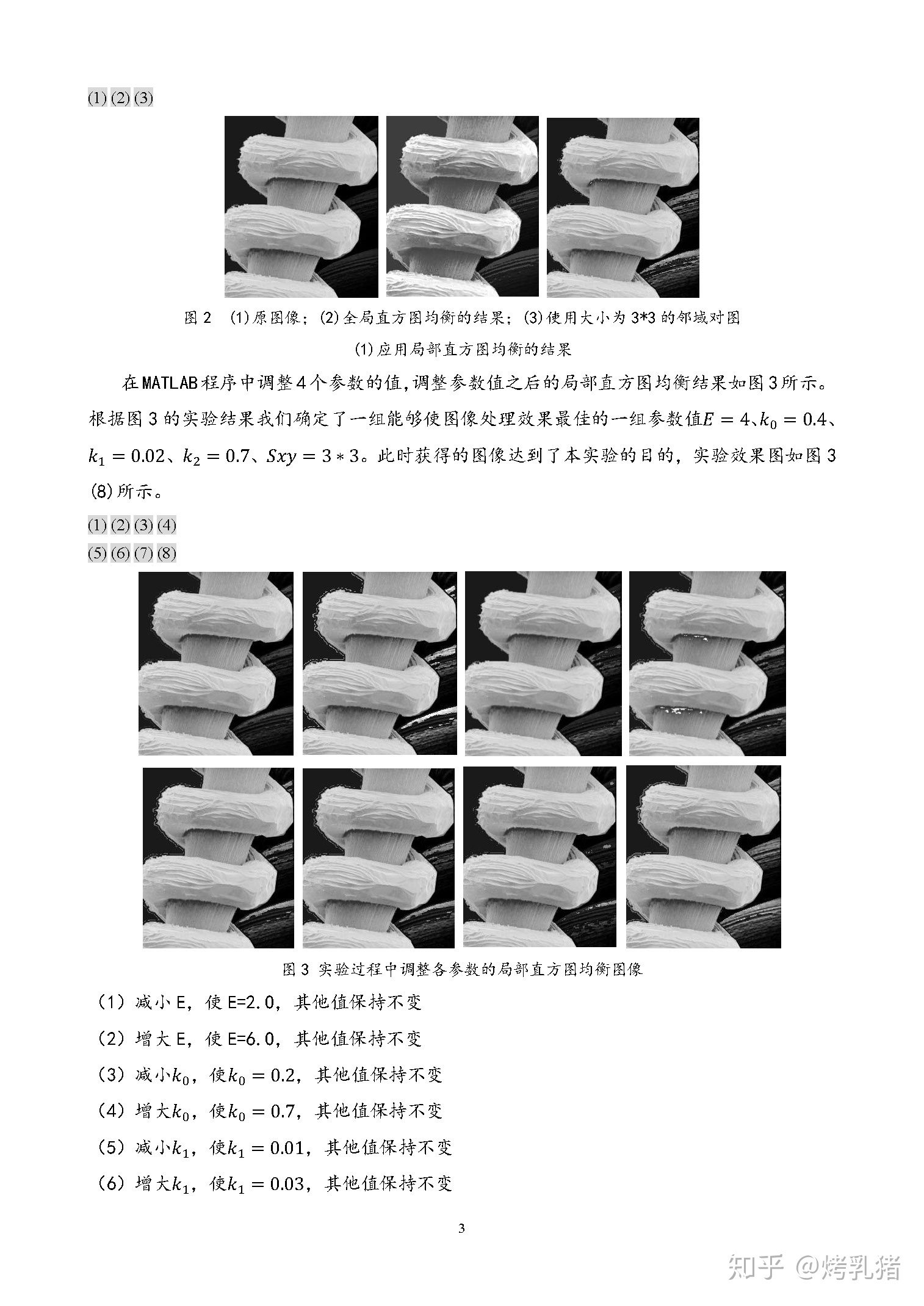


1. 参数选择方法研究
2. 实验原理（如书所写）

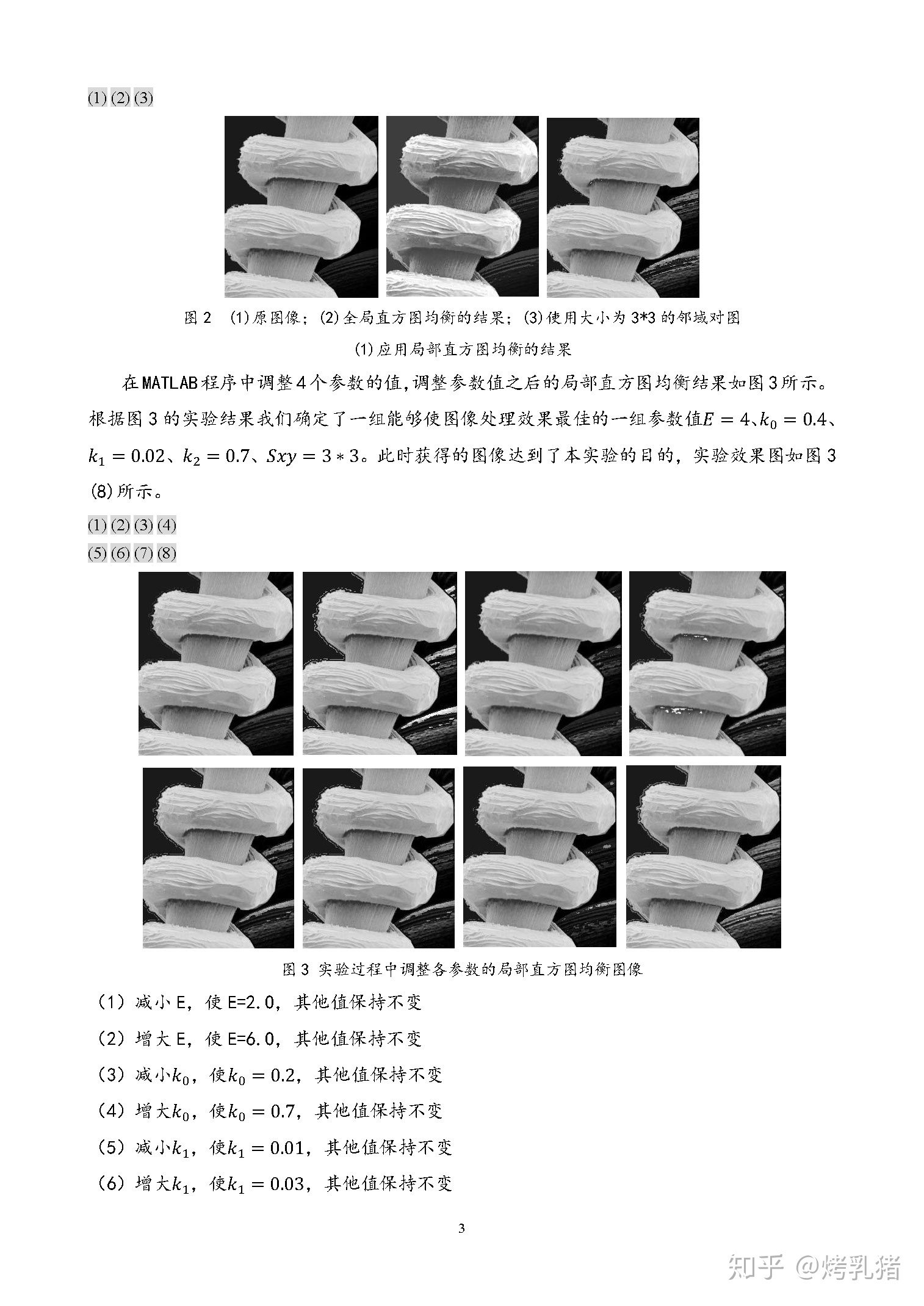


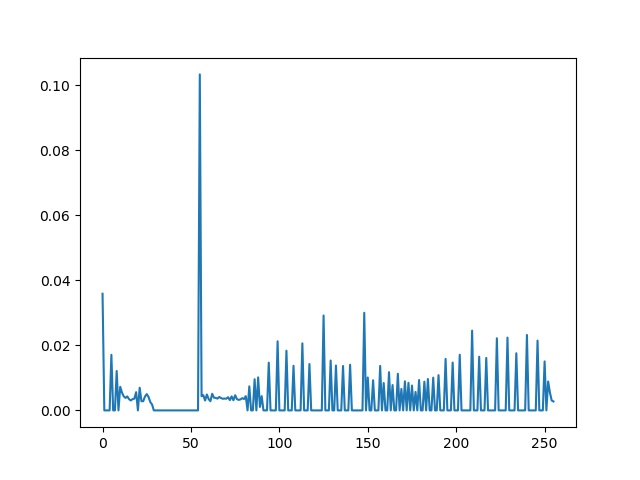
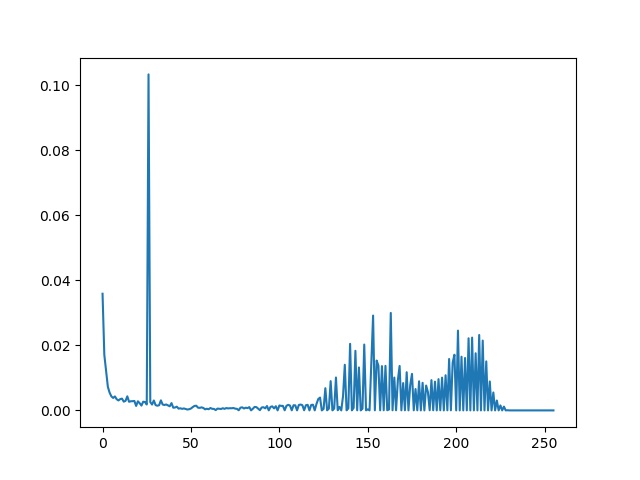
1. 实验结果

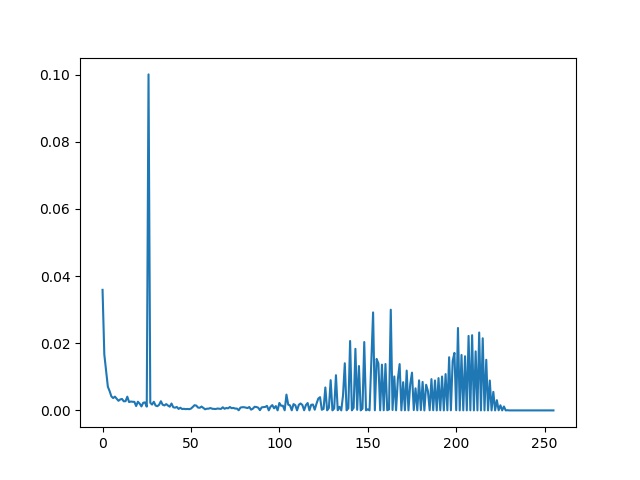
根据书本提示，设定初始值为E = 4，Sxy = 3，K0 = 0.4，K1 = 0.02，K2 = 0.4；得到结果为：



从左到右分别为：原始图像、全局均衡化图像、局部增强处理图像；

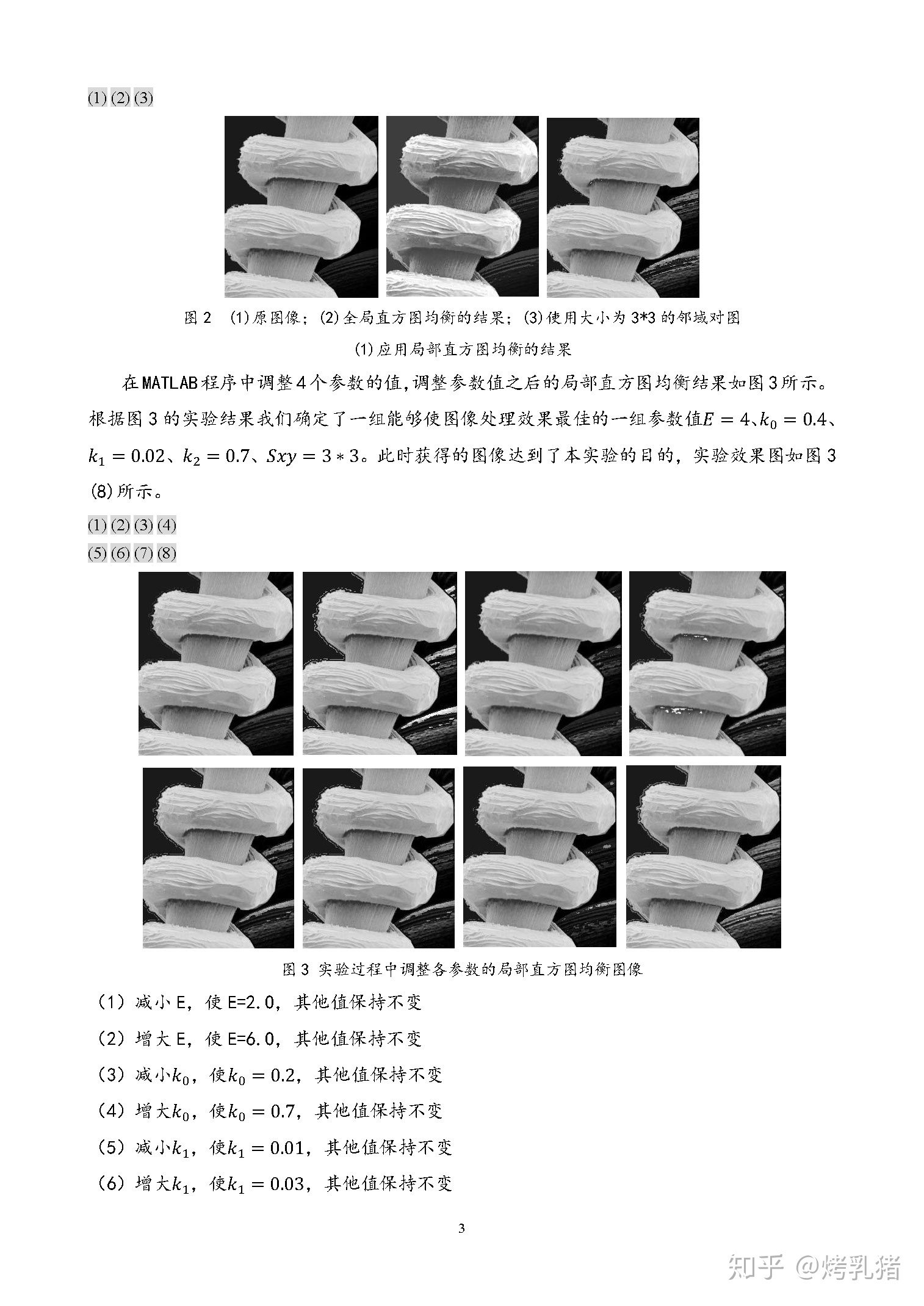


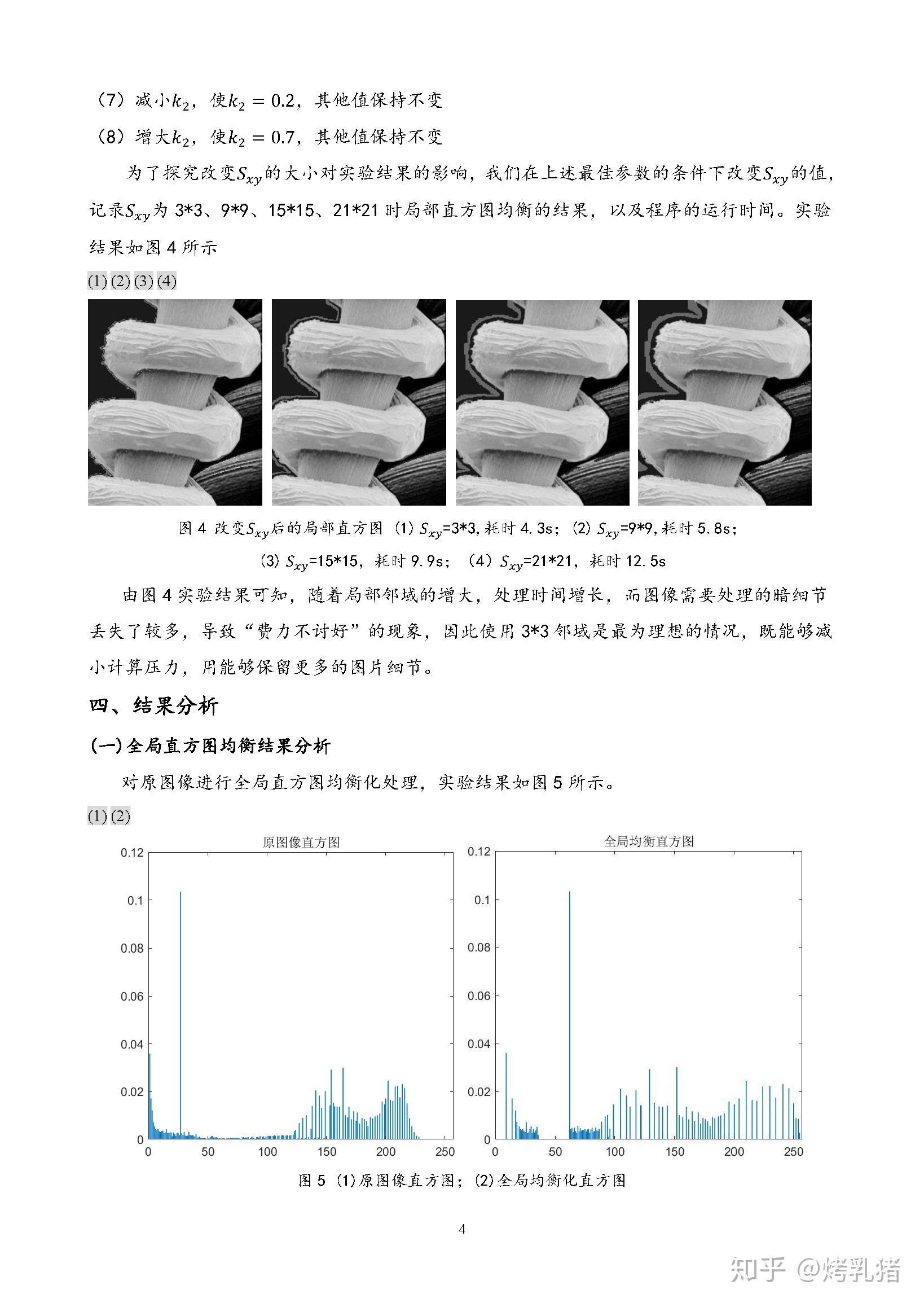




1. 参数调整

由实验结果可以看出，局部增强处理比起全局均衡处理有更好的效果，符合预期。为了进一步讨论参数对于增强效果的影响规律，我采取了控制变量法，通过调整某个参数，观察处理结果的质量，来判断参数对处理结果的影响作用。由于时间和能力有限，本次实验只测试了十二组参数组合，实验结果如下：





1. E = 2，其余默认；
2. E = 6，其余默认；
3. K0 = 0.2，其余默认；
4. K0 = 0.6，其余默认；
5. K1 = 0.01，其余默认；
6. K1 = 0.03，其余默认；
7. K2 = 0.2，其余默认；

（8） K2 = 0.6，其余默认；

（9） Sxy = 3\*3

（10）Sxy = 5\*5

（11）Sxy = 7\*7

（12）Sxy = 9\*9

1. 规律分析
2. E

该值主要决定将按出灰度值扩大的倍数。由3中结果可以看出，E越大，暗处处理后更明亮，反之则效果不明显。可以通过比较图像中暗处灰度均值和亮处灰度均值来确定该参数。

1. Sxy

该值为局部增强过程中单次处理区域的尺寸大小。由3中结果看出，该值的变化对图像处理的效果影响并不显著，此外在实验过程中，当增大该值时，有明显的计算时间增长现象。从理论上来说，我们也希望单次处理的区域越小越好，这样可以使得局部增强的处理更细微，效果会更好。因此在处理实验所给图片（尺寸）时，考虑3\*3的局部直方图区域大小。

1. K0

该值主要用于区分变化区域和不变化区域。由3中结果可以看出，K0较小时，增亮的原暗处区域就相对较少，K0较大时，又使得一些原亮区域增强，也不符合我们的实际需求。因此K0对于局部直方图处理过程有较大影响作用，需要谨慎选择。

1. K1

该值主要作为局部区域方差最小值。由3中结果可以看出，K1较小时，图像处理过度，一些细节更加模糊，图像的背景也发生了一些变化，K1较大时，图像处理不足，一些暗处区域变化不明显，因此K1对选择待处理区域有较大作用。

1. K2

该值主要作为局部区域的方差最大值。由3中结果可以看出，K2减小时，会损失局部灰度值方差较大的区域，K2增大时，会增加灰度值方差较大的区域。也就是说，在确定为灰度均值较小的暗处区域基础上，适当增大K1可以更多的将这些暗处区域作为待变化区域，因此选择相对较大的K2可以提高处理效果。

1. 问题及思路
2. 关于直方图的统计

Opencv模块中提供了获取图像像素直方图结果的函数cv.calcHist()，在实验过程中，我一度没有明白该函数的输入参数和输出参数的类型，将输出值误以为是图像。明白输出值为数组后，即可利用matplotlib模块中的matplotlib.plot()函数根据输出数组绘制直方图。也可利用matplotlib模块中的matplotlib.hist()函数，将原图像数组扁平化处理（image.ravel()）后绘制直方图，在将数值转换为概率密度时，在函数中添加‘density = True’参数即可。

（2）关于灰度均值、方差的计算

在实验开始时由于对概念理解不清，误以为直接计算直方图函数输出数组的均值和方差即为所求。后在局部直方图处理过程中发觉异常。应先读取图像每个像素点的灰度值，再进行相关计算。

（3）关于全局图像均衡化

可以利用opencv中的cv.equalizeHist()函数实现。注意与cv.calcHist()函数不同，此函数输出结果为图像。

（4）关于局部直方图统计中掩膜的设计

实验中该部分主要遇到的问题是，用户输入参数值与图像不匹配问题。因此在处理前，需要先对用户输入的参数值进行计算，判断输入参数构成的掩膜尺寸是否合理、可行。另外也在该过程中掌握了局部图像显示的处理操作。

（5）关于边缘点的处理

在局部增强过程中，对于原图像的边缘像素点，为使其具有相同尺寸的邻域，在原图像进行了边缘填充的操作，可以利用opencv中的cv2.copyMakeBorder()实现。实验中我尝试了两种填充方法，一种为边缘0填充，另一种为边缘复制填充，两种方式在函数中的参数分别体现为cv2.BORDER\_CONSTANT,value=[] 和 cv2.BORDER\_REPLICATE。经过对比，最终选择了边缘0填充方法。

但边缘填充的方法会导致边缘像素邻域的均值方差会受到填充值的影响，可能会产生一些效果影响，因此进一步将方法更改为：对边缘像素点特殊处理，只对处于邻域内且处于图像内的像素点计算均值方差。

最终选用了第二种方法。

（6）关于图像像素点的查询与修改

本实验进一步掌握了像素级别的操作，也对图像的构成有更清晰的认识。可以通过遍历循环的嵌套对图像的每一个像素点进行查询和处理。

（7）关于参数设置和结果可视化

为满足实验需求，本实验依旧支持用户在可视化界面上输入自定义参数，参数全部可调，同时也不需要用户对输入参数进行合理性判断和分析。系统将自动计算并分析可行性。另外本实验所有输出结果均可在UI界面上可视化显示。

1. 总结

通过本次实验，我对图像构成有了更深刻的了解，也对图像像素级别的处理有了初步的认识。掌握了像素的查询和修改、图像直方图及图像灰度统计、全局图像均衡处理、选择性局部图像增强处理及其参数含义。为接下来的课程内容奠定良好的基础。