

# 数字图像处理和模式识别综合实验报告

课题名称: 手机屏幕划痕检测

院    系: 人工智能与自动化学院  
班    级: 自动化 1704  
姓    名: 韩 晗、李佼佼  
学    号: U201714253、U201714267  
指导老师: 马 杰、郑定富

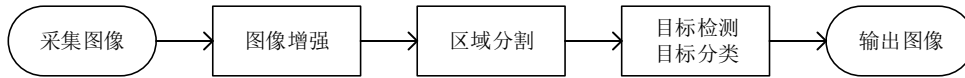
自动化学院  
2019 年 2 月 5 日

# 目 录

一、总体方案设计 .....	3 -
二、关键技术 .....	4 -
三、图像采集系统设计 .....	4 -
四、源程序设计和运行结果 .....	4 -
五、实验结果评价 .....	6 -
六、建议和体会 .....	7 -
七、参考文献 .....	7 -

## 一、总体方案设计

### 总体设计：



采集图像后对图像进行增强操作以提高信噪比，将增强过后的图像分割成手机屏幕内部和手机屏幕外部两部分，后续的目标检测与分类只在手机屏幕内部进行操作，得到结果图后输出图像。各部分详细操作如下。

### 采集图像：

1. 找到使待处理图像信噪比最大的打光角度。手机划痕检测对灯光要求较高，所以实验的第一步就是通过变换不同的打光角度，以找到能够使图像信噪比最大的灯光位置。

### 图像增强：

2. 灰度图中值滤波去噪。采取对待处理图像后，将其转为灰度图进行去噪。在多次实验中我们发现图像的噪声主要来自于打光不均匀所带来的椒盐噪声，具体表现为黑色屏幕的背景上会随机出现灰度值跳变的白色小点（主要是灰尘与光源的反射），所以我们使用卷积核大小为  $3 \times 3$  的中值滤波来进行第一次图像去噪。
3. 梯度计算。对经过第一次去噪的灰度图求取梯度，检测图像的边缘。经过多次实验，我们发现在 prewitt、sobel、canny 三个算子中 canny 算子综合表现更优异，所以我们最终决定使用 canny 算子对灰度图进行梯度计算。

### 区域分割：

4. hough 变换检测手机边缘，使用 Hough 变换检测出图像中所有高于阈值的直线，并利用手机边缘的直线相互垂直或平行这一特性去除掉有噪声或裂纹构成的直线，然后对剩余的直线进行排序分类并从每一类中选区一个，直线的类数即手机边缘直线数。
5. 采用基于几何特征的矩形区域检测方法，去除手机边框对划痕检测的干扰。根据第四步传入的直线信息来确定有效边界点（直线之间的交点或者直线与边界的交点），并根据边界点构成直线的斜率来向屏幕中央适当的缩进，将范围之外的图像信息全部去掉。

### 目标检测与分类：

6. 对梯度图进行数学形态学处理。在梯度图上对手机边界内部区域进行数学形态学运算，主要为膨胀运算，将检测出的相近而不连续的裂痕连到一起，保证裂痕的连续性。
7. 对第六步计算出的图形进行连通域标记并计算其面积。标记连通区域的同时计算联通区域最小外接矩形的面积与连通区域自身的面积，将连通域面积小于设定阈值的判定为噪声。计算其余连通域面积与最小外接矩形面积的比值记为占有率（假设头发是较分散的分布在屏幕上方，且与划痕无交叉），若占有率小于设定阈值，则认为该划痕疑似为发丝，进行下一步分类。否则判定其为手机划痕。
8. 计算第七步中不确定连通域的平均灰度值（基于图像中发丝亮度明显大于划痕亮度这一物理现象），大于设定阈值时判定其为发丝，否则判定为手机划痕。

### 输出图像：

9. 将第八步得出的二值图叠加到原图像并输出结果。

## 二、关键技术

1. 连通域标记。种子生长法，选中图像上白色点作为种子，后根据堆栈先进后出的特性，由此点向四周发散直至将该连通域生长完毕。之后选择下一个种子再次进行生长。
2. Canny 算子。将图像高斯平滑后求取梯度，通过对梯度图进行非极大值抑制细化边缘，再根据双阈值与邻域追踪方法连接边缘。
3. Hough 变换。利用直角坐标系和映射坐标系的点和线的对应性，即直角坐标系的一条直线对应极坐标系的一个点，将图像信息映射到极坐标系中，极坐标系中累计的值越多相应直角坐标系的直线就越多，不断地优化阈值使得保留边界直线的同时去掉噪声裂纹构成的直线，一条直线的信息由  $r$  和  $\theta$  来表征，再依此进行排序聚类并从每一类中选取一条作为结果直线。

## 三、图像采集系统设计

因为手机屏幕划痕检测对光照要求较高，所以我们设计的采集系统不是连续的动态的返回结果，需要人为操作按钮才可以采集图片并返回结果图。采集系统呈现给用户的界面见 Figure 1。

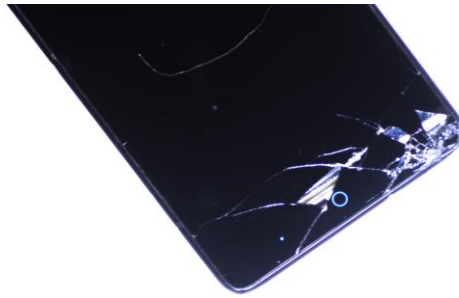


Figure 1: 左为用户交互界面，右为原图像  
操作时点击“开始采集”即可采集图像并呈现结果。

## 四、源程序设计和运行结果

编程时也可以大致分为以下三个模块（不包括用户交互界面的编写）：

### 图像增强：

1. 中值滤波去噪：输入为采集到的图像的灰度图。第一次编写该函数时使用  $3 \times 3$  大小卷积模块以步长为 1 卷积增广图，但是在实验中发现该实现方法进行了大量的重复运算，计算量很大，运算时间达不到要求。于是在查阅资料后改用了基于直方图的快速中值滤波方法<sup>[1]</sup>。 $3 \times 3$  的模块滑动时 63D0 取窗口内所有元素建立直方图，之后从左至右计算累加直方图中每个灰度级像素点个数，记为  $\text{count}$ ，直到  $\text{count}$  大于 5.5，此时灰度值即为中值。模块在同一行每次滑动时更新  $\text{count}$  即可在几乎不损失性能的情况下改善滤波速度。运算结果见 Figure 2。原图像见 Figure 1。

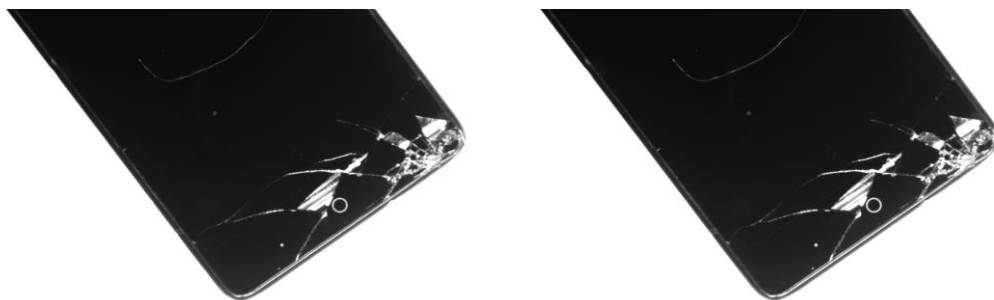


Figure 2: 左为自己实现结果图，右为 OpenCV 结果图

2. Canny 算子计算梯度：输入为均值滤波后的灰度图。该函数也编写了两个。第一个是在对 OpenCV 中 Mat 类型不够了解时编写的。该函数将 Mat 转换为二维矩阵参与运算，并且实现邻域追踪时十分粗暴，所以计算时间长且表现比较差，出现了双边缘甚至边缘丢失。在对 Mat 类有了一定了解后，自己写了第二个 Canny 算子的函数，将邻域追踪方法使用了队列实现，减小了运算时长也改善了其表现。运算结果见 Figure 3。

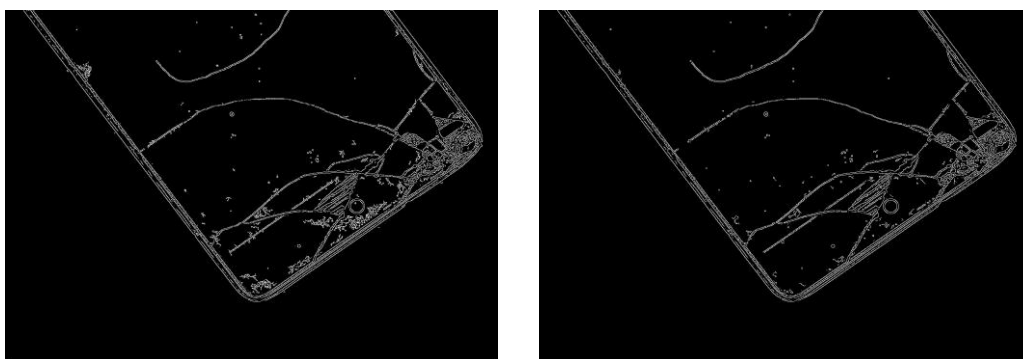


Figure 3: 左为自己实现结果图，右为 OpenCV 结果图

3. 数学形态学处理：主要为图像的膨胀运算。第一次实现时使用  $3 \times 3$  的卷积模板在图像上以步长为 1 滑动处理。效果较好，但是计算时间较长。受快速均值滤波的启发，第二次实现时也使用直方图进行滑动处理，提高了计算速度。运算结果见 Figure 4。

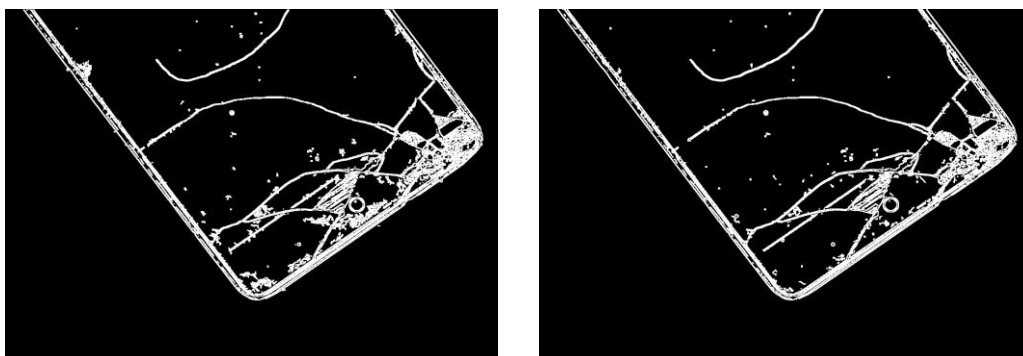


Figure 4: 左为自己实现结果图，右为 OpenCV 结果图

#### 区域分割：

4. Hough 变换：先采用映射到直角坐标系的方法发现斜率可能会趋于无穷大，后改为映射到极坐标系的方法，一条直线由  $r$  和  $\theta$  来表征，且不断地优化阈值使得保留边界直

线的同时去掉噪声裂纹构成的直线，然后对直线先根据  $\theta$  再根据  $r$  排序来对直线进行分类，并在每一类中选取一条出来，过程中经过试错才发现了可能会有  $\theta$  值为 355 和 1 的直线是一类但是排序不会在一起 所以修改了选取直线的判别条件。

5. 采用基于几何特征的矩形区域检测方法，去除手机边框对划痕检测的干扰：根据上一步传入的直线数来进行不同的操作，根据不同的直线数确定相应的有效边界点并以此来对图像进行缩进其中判断直线的交点是否有效依据它的位置是否在图像范围内，直线与边界的交点是否有效则根据它的一定邻域范围是否有有效像素点。运算结果见 Figure 5

### 连通域标记与分类：

6. 连通域标记并计算面积进行分类：输入为经过区域分割后的二值图，白色为待检测目标，黑色为背景。基于堆栈的先进后出特性遍历图像，是白色点则入栈并向其四邻域发散，非白色点则继续遍历图像。在遍历同时计算连通域面积与其外接矩形面积及二者的比值作为占有率，将连通域面积小于阈值的判定为噪声，大于阈值再判定其占有率，若小于阈值则为疑似发丝，大于则判定为划痕。对于疑似发丝的目标计算其平均灰度值，若大于阈值则判定为发丝，若小于阈值则判定为划痕。运算结果见 Figure 5。

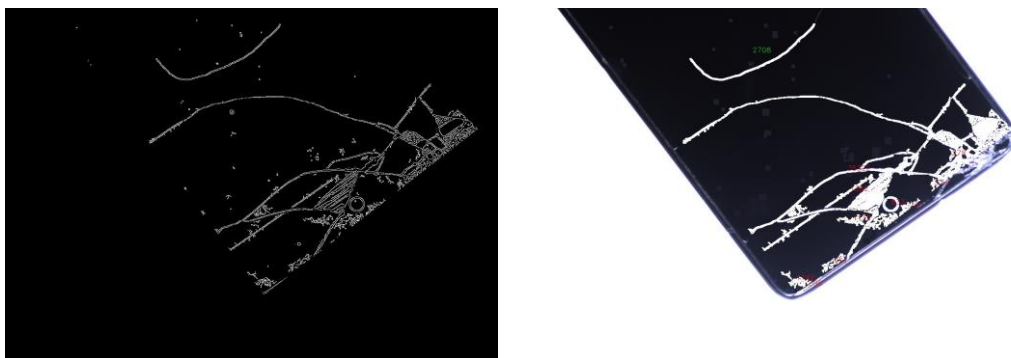


Figure 5: 左为区域分割结果图，右为连通域标记结果图（绿色面积标记的为头发）

## 五、实验结果及其评价

实验结果：选择三张结果图作为结果展示，见 Figure 6。

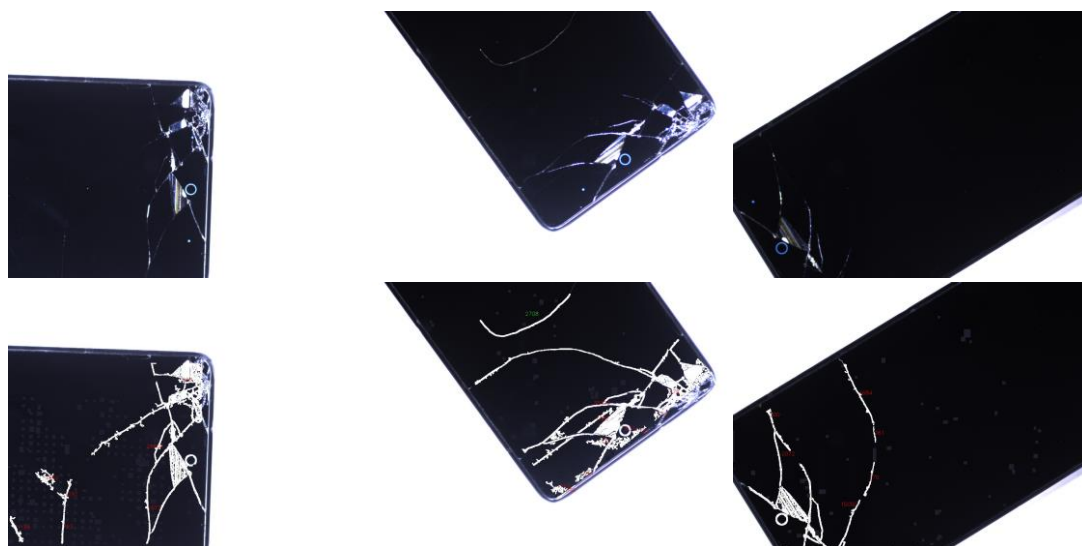


Figure 6: 结果图展示，上方为原图，下方为结果图

**评价：**处理后的实验图像基本可以检测出所有的划痕，能够正确计算出划痕的面积，在一定程度上也能够区分头发（绿色面积标记）与划痕（红色面积标记）。但是我们的实验也依然有所欠缺，比如检测划痕有可能不连续，见 Figure 6 左 1 与左 3，同时噪声影响并未完全去除，见 Figure 6 左 1。分析产生这些现象的原因如下：

1. 光源。本次实验时因为光源的限制，我们只使用了一个线光源进行打光，实验后期甚至只能使用圆光源进行打光，导致成像效果不好，噪声严重，见 Figure 6 左 1。
2. 滤波造成的图像边缘信息丢失。为了去除光源造成的噪声影响，我们对图像进行了两次滤波操作，该过程中很有可能丢失划痕的边缘信息，导致划痕不连续，见 Figure 6 左 3。

根据这些问题，我们也设计了我们程序的优化方向，主要如下：

1. 光源上的优化。
2. 滤波的优化。进行中值滤波时会丢失部分划痕边缘信息，实验后期我们想到使用自适应滤波方法来处理图像，但是因为时间的限制，程序并未编写完成。
3. 区域分割的优化。我们是基于手机的几何特征进行的识别，检测手机的直线进而分割图像，但是该方法对形状不规则的手机并不适用。所以此分割方法依然有可优化的空间。
4. 速度上的优化。整个处理过程需要 1.3s 左右，时间主要耗费在中值滤波和 hough 变换上。尽管中值滤波已经采用了基于直方图的快速算法，但是运算时间依然较长，后来我们基于这篇论文<sup>[2]</sup>想实现时间复杂度更小的算法，但是也因期末时间的限制未能如愿。

## 六、体会和建议

**体会：**实验过程中能够将书本<sup>[3]</sup>上或者论文中学习到的算法复现是件很开心的事情，专注于算法速度的提升也是件很纯粹并且很有趣的事情。总的说来实验的过程很有挑战性但也极其有趣味性。并且通过此过程我们更加熟悉了 VS 和 OpenCV 的使用，对图像处理有了更深刻的认识，在实现课程上学习到的算法时也加深了我们对算法的理解。

**建议：**实验总体是充满乐趣的，整个过程并没有感觉像其它实验那样枯燥，但是实验中也存在一些小小的问题，比如硬件的不足，实验中使用的线光源或多或少都有问题，导致成像有很大噪声，以至于我们后期只能使用圆光源进行实验。

## 七、参考文献

- [1] Huang T, Yang G, Tang G. A fast two-dimensional median filtering algorithm[J]. IEEE Transactions on Acoustics, Speech, and Signal Processing, 1979, 27(1): 13-18.
- [2] Perreault S, Hébert P. Median filtering in constant time[J]. IEEE transactions on image processing, 2007, 16(9): 2389-2394.
- [3] 章毓晋. 图象工程/上册/图象处理和分析[M]. 清华大学出版社有限公司, 1999.

指导教师的评语及评分	
综合评分	