

数字图像处理作业 综合作业 2
去反光与工件提取
项目报告

晏筱雯

自 42

2014011459

2016.11.25

目录

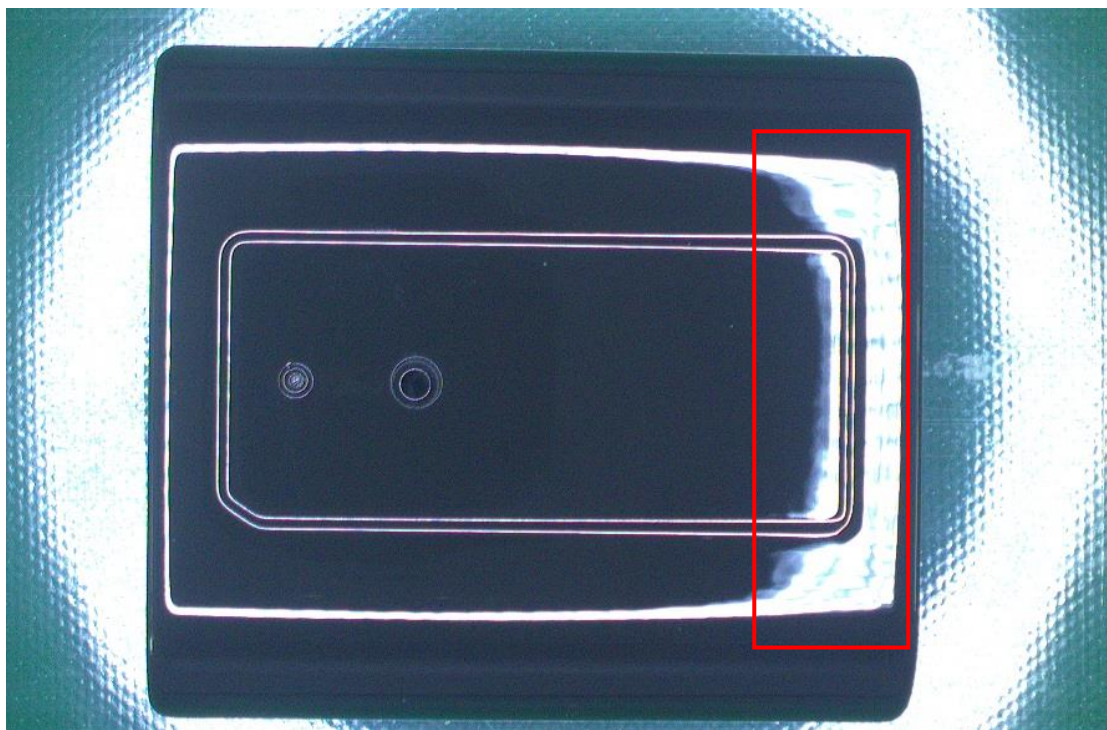
- 1 作业要求 3
- 2 方法原理 3
- 3 作业思路和算法设计 4
 - (1) 寻找反光区 4
 - (2) 去反光..... 5
 - (3) 提取轮廓..... 6
 - (4) 分离工件..... 8
 - (5) 基于 TV 模型修复的图像去反光..... 8
- 4 结果讨论 9
 - (1) 寻找反光区 9
 - (2) 去反光..... 9
 - (3) 提取轮廓..... 11
 - (4) 分离工件..... 11
 - (5) 基于 TV 模型修复的图像去反光..... 12
- 5 收获与总结 13
 - (1) 遇到的问题及解决办法..... 13
 - (2) 收获..... 14

1 作业要求

对下图工件中反光较为严重的区域（红框区域）实现去反光，可自己进行文献调研选择适当的图像处理方法，并用压缩包中的十张图片测试算法鲁棒性。（提示：可以使用同态滤波）。

选做：将工件与背景分离（加 10 分）。

注：为满足部分同学设计算法对图片清晰度的要求，作业中给的图片较大，有需要的同学可自己进行压缩。



2 方法原理

这次作业要求对一张反光严重的照片去反光，根据助教的提示，可以利用同态滤波在频域中对于图像照度分量和反射分量的处理来达到去反光的目的是。而对于分离工件，可以根据联通区、膨胀腐蚀等方法 and 原理找到工件的轮廓，从而完成分离。

另外，由于一开始对于图像同态滤波的效果不是很好，我也在网上调研了基于 TV 模型去反

光的方法，并进行了初步实践，虽然效果不是很好，但是对于我也有一定的启发。

3 作业思路和算法设计

由以上的分析，这次作业我的基本思路是先找到图像的反光部分，对反光部分同态滤波，再与原图合成得到去反光之后的图像，最后分离工件。

(1) 寻找反光区

HSV(Hue, Saturation, Value)是根据颜色的直观特性由 A. R. Smith 在 1978 年创建的一种颜色空间，也称六角锥体模型(Hex cone Model)。其中，明度 (Value) 表示颜色明亮的程度，对于光源色，明度值与发光体的光亮度有关；对于物体色，此值和物体的透射比或反射比有关，通常取值范围为 0% (黑) 到 100% (白)。反光是一幅图像中亮度最高的部分，根据这一特性，我们可以根据图像的 hsv 模型来制作掩模，掩模的特点是反光区的灰度值为 0 而其他部分的灰度值为 1。

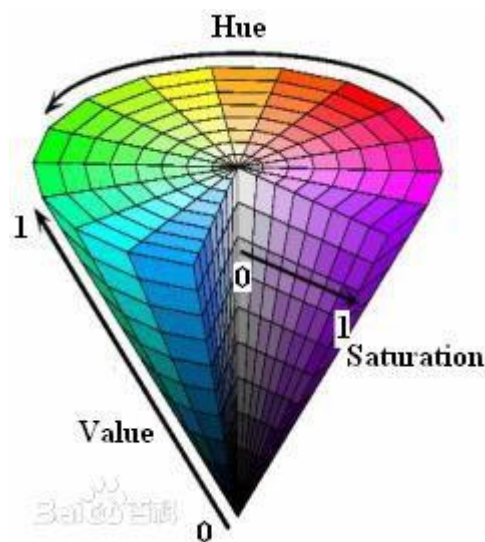


图 1 hsv 模型¹

¹ 图片来源：

http://baike.baidu.com/link?url=UTAiRqfKryUhWPueZOjixQmYHeHard_JrGXsfh1BqXI7aR2YRhUx

(2) 去反光

一幅图像 $f(x, y)$ 可以看成是照射分量 $i(x, y)$ 和反射分量 $r(x, y)$ 的乘积:

$$f(x, y) = i(x, y) \cdot r(x, y) \quad \cdots \cdots \textcircled{1}$$

这次的作业中我们就需要对图像的反射分量进行处理, 从而去除改善反光的影响, 揭示反光区域的颜色、纹理细节。

通过①式我们知道, 由于照射分量和反射分量是乘积的关系, 因此不能通过直接对图像 $f(x, y)$ 直接处理, 而必须先作一步对数变换:

$$\ln f(x, y) = \ln(i(x, y) \cdot r(x, y)) = \ln(i(x, y)) + \ln(r(x, y))$$

对上式两边取傅里叶变换得:

$$F(x, y) = I(\ln(i(x, y))) + I(\ln(r(x, y)))$$

图像的照射分量通常由慢的空间变化来表征, 而反射分量往往引起突变, 特别是在不同物体的连接部分。这些特性导致图像取对数后的傅里叶变换的低频成分与照射相联系, 而高频成分与反射相联系。

使用同态滤波器可以更好地控制照射分量和反射分量。这种控制器需要指定一个滤波器函数 $filter(\gamma_1, \gamma_2)$, 它可用不同的可控方法影响傅里叶变换的低频和高频。如果 γ_1 和 γ_2 选定, 而 $\gamma_1 < 1$ 且 $\gamma_2 > 2$, 那么滤波器函数趋近于增强低频(照射)的贡献, 而衰减高频反射的贡献。

最终结果是同时进行动态范围的压缩和对比度的减弱。

通过调整 γ_1 和 γ_2 , 在频域处理完图像之后, 通过以上步骤的逆变换, 即反傅里叶变换、求指数之后就可以得到经过同态滤波之后的图像。

以上的步骤用流程图表示如下:

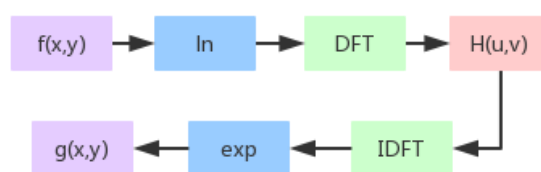


图 2 同态滤波流程图

由于原图像中反光成分太多，反光区域中保留的颜色纹理信息较少，所以根据第一步制作的掩模，对于原图中没有反光的部分，依然保留原图信息，而反光区则用同态滤波的结果来覆盖。

(3) 提取轮廓

①连通区

四连通或八连通是图像处理里的基本概念：8 连通，是说一个像素，如果和其他像素在上、下、左、右、左上角、左下角、右上角或右下角连接着，则认为他们是联通的；4 连通是指，如果像素的位置在其他像素相邻的上、下、左或右，则认为他们是连接着的，联通的，在左上角、左下角、右上角或右下角连接，则不认为他们连通。

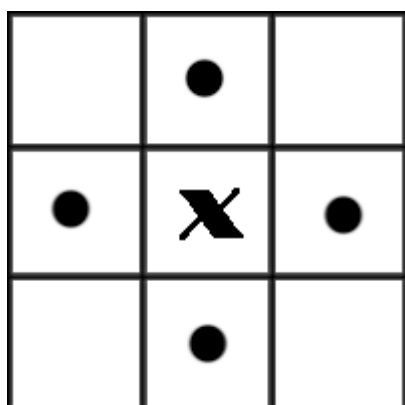


图 3 4 连通区

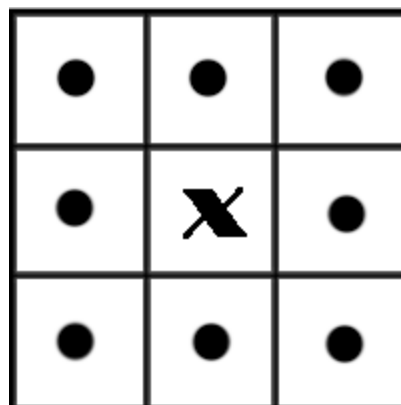


图 4 8 连通区²

工件的外轮廓可以看作是一个联通区，因此我们可以利用 MATLAB 中自带的 bwlabel 函数来

² 图片来源：<http://www.cnblogs.com/nsnow/p/4544115.html>

标记图像的连通区从而找出外轮廓。

②基于数学形态学的膨胀腐蚀³

在数学形态学中，“结构元素”相当于信号处理中的“滤波窗口”，假设 $B(x)$ 是一个结构元素，那么对于工作空间 E 中的每一点 x ，腐蚀和膨胀定义为：

$$\text{腐蚀: } X = E \ominus B = \{x: B(x) \subset E\}$$

$$\text{膨胀: } Y = E \oplus B = \{y: B(y) \cap E \neq \emptyset\}$$

简单的腐蚀是消除物体的所有边界点的一种过程，其结果使剩下的物体沿其周边比原物体小一个像素的面积。简单膨胀是将于某物体接触的所有背景点合并到该物体中的过程，其结果是使物体的面积增大了相应数量的点。

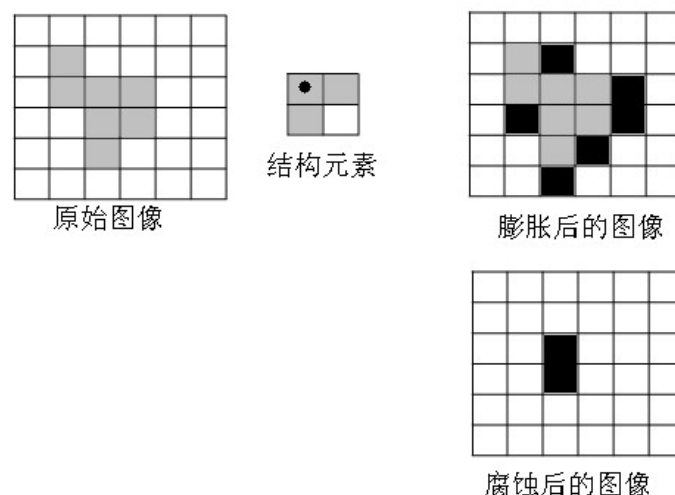


图 5 膨胀腐蚀⁴

在利用 bwlablel 提取边缘的过程中，得到的边缘可能有很多噪声，因此我们可以用腐蚀去掉边缘中残留的噪声，再用膨胀使边缘更光滑。

³ 资料来源：

http://wenku.baidu.com/link?url=t8N3XA0jRCZ1C64OtKa8v3SPGUCtICTeFNZYuNVCQ8KEw2SHdyqpPLX_rXtTQ5qPHx3yDDTmX2_MrQwlh0LUhoRGeRvytAhuTl3s9mLTBbu

⁴ 图片来源：<https://zhidao.baidu.com/question/745091163435016332.html>

(4) 分离工件

有了第三步中提取的工件外轮廓,就只需判断去反光之后的点是否在外轮廓围出的区域中即可。MATLAB 中自带了函数 $\text{inpolygon}(x,y,x1,y1)$, 其中 $x1$ 和 $y1$ 分别是轮廓的横纵坐标的集合, 而 (x,y) 是待判断的点的横纵坐标。将不在外轮廓中的点的像素值置为 0, 即可成功地分离工件。

(5) 基于 TV 模型修复的图像去反光

在使用同态滤波的过程中, 我发想结果并不太尽如人意, 于是调研了一些别的方法并进行尝试, 基于 TV 模型的图像修复就是其中一种。

对于待修复区域的建模如下图所示, D 表示待修补区域, E 是处于信息完好区域中包围 D 的一个图像邻域。定义 u^0 为图像的初始值, u 表示修复后的图像值, TV 模型的修复过程就是 E 中能量向 D 内传播的过程。

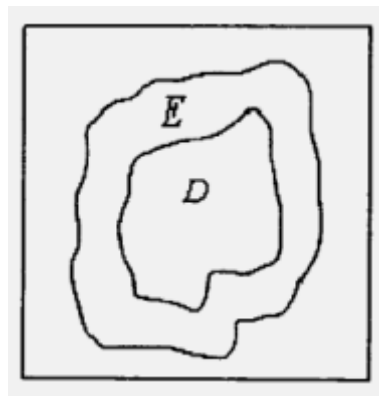


图 6 TV 模型

简单的算法如下:

- ①读入图像和掩模信息。
- ②对图像掩模区域中的每一个像素, 执行第③、④步。
- ③若像素位于待修补区域内部, 则进行修复。

④计算缺损像素值的估计值，并保存到新的图像中。

⑤以新图像代替旧图像并转第②步执行，直到迭代次数完成。

4 结果讨论

(1) 寻找反光区

制作出的掩模如下，黑色的即为反光区：

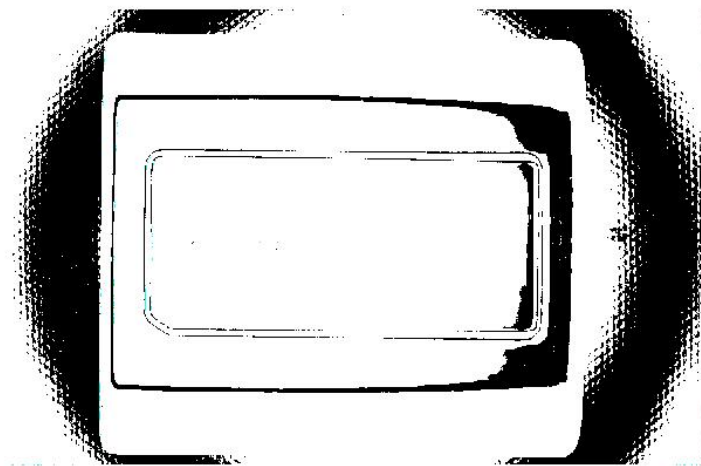


图 7 掩模

(2) 去反光

在经过反复地调整同态滤波器的参数之后，终于得到了如下的结果：

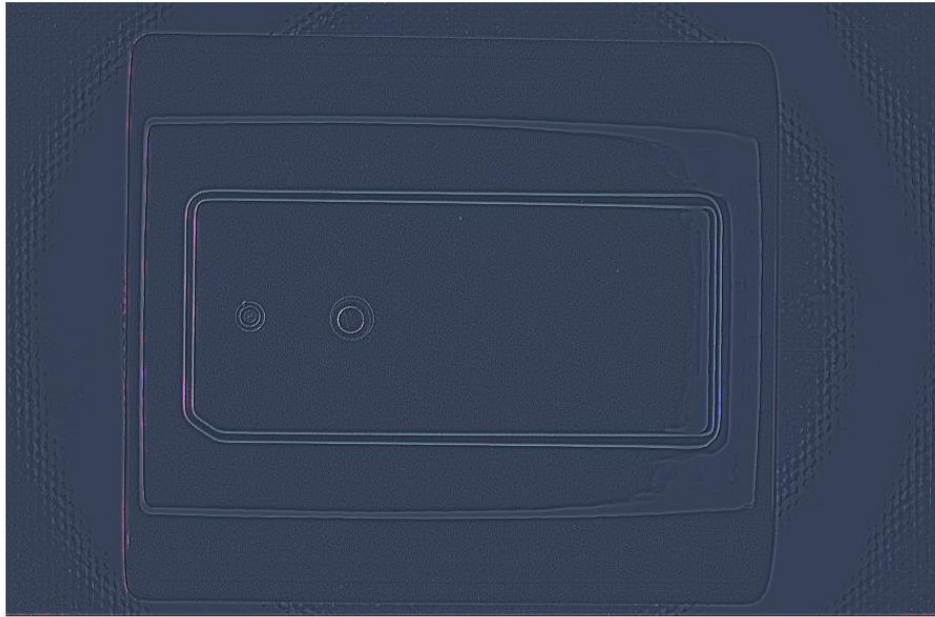


图 8 同态滤波结果

如图 8 所示，同态滤波之后，图像的反光区域已经基本没有了。

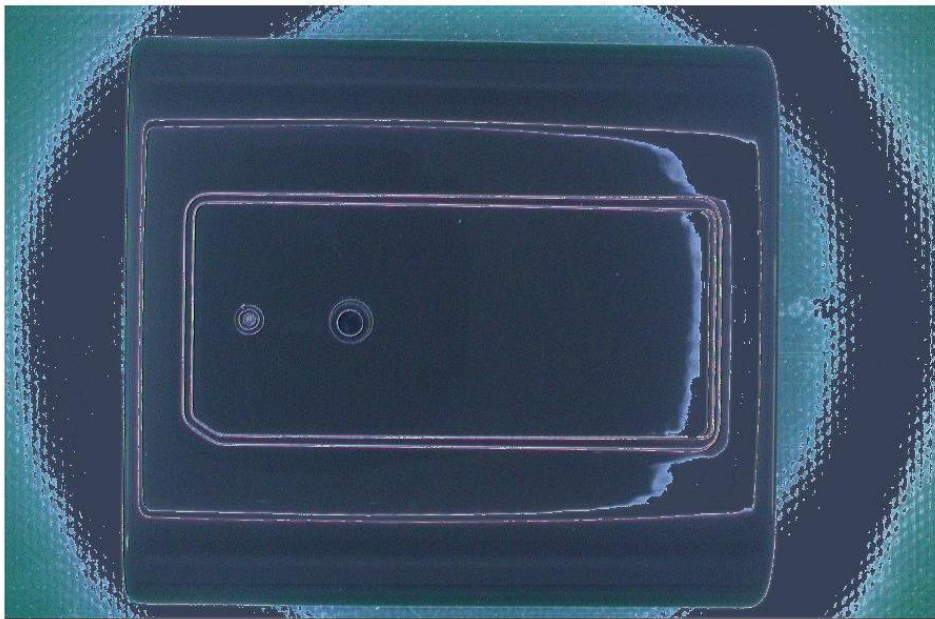


图 9 与原图合成之后的结果

如图 9 所示，把经过同态滤波的图和原图合成之后，基本得到了想要的结果，但是在反光的边缘部分有着颜色的突变。

(3) 提取轮廓

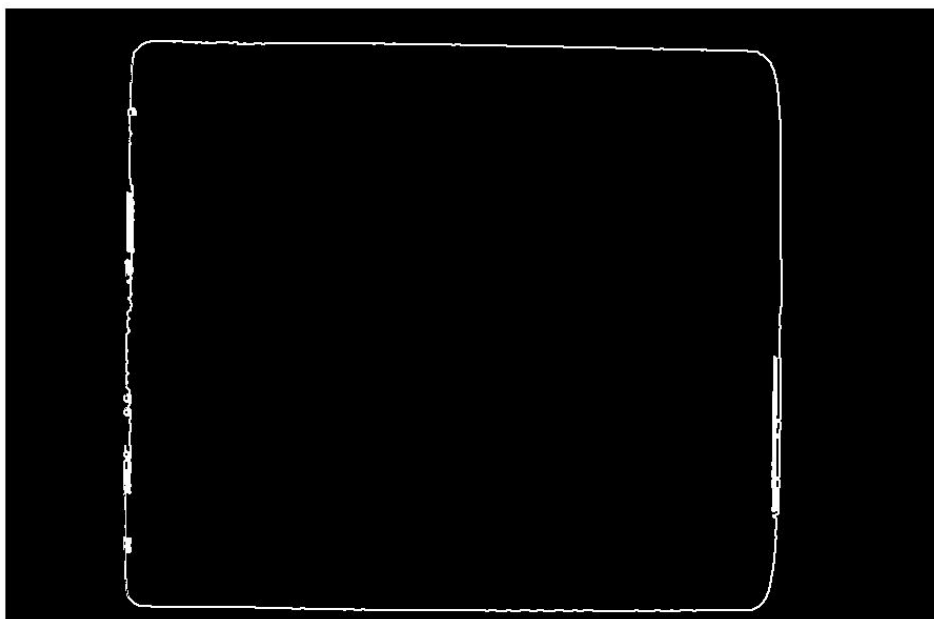


图 10 工件轮廓

如图 10 所示，工件轮廓还比较清晰，但是由于图像噪声的存在，即使经过了膨胀和腐蚀，边缘依然不平滑。

(4) 分离工件

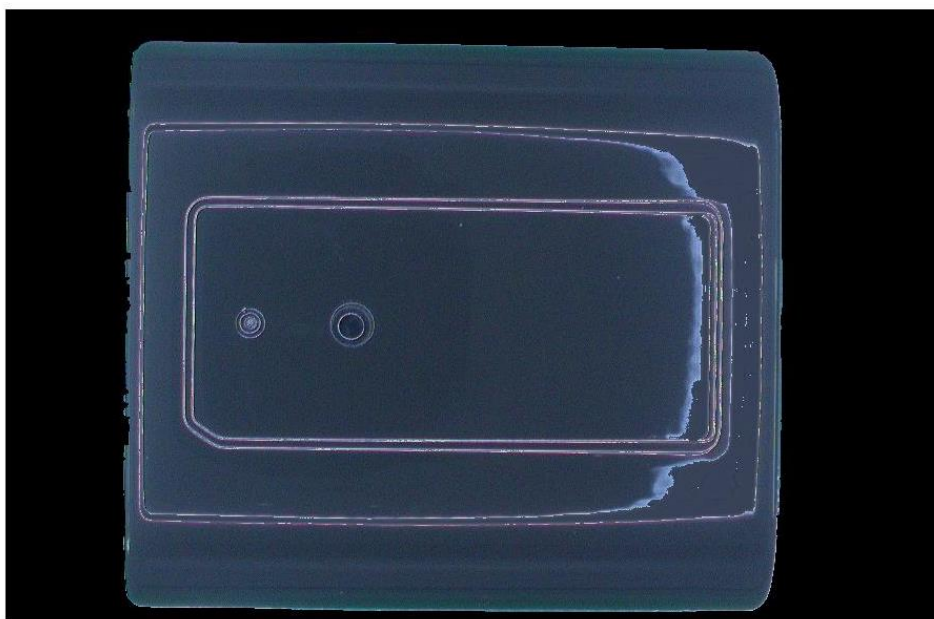


图 11 分离工件结果

分离工件的结果可从轮廓提取的结果判断，工件基本被分离出来，但是轮廓不平滑。

另外从整个程序运行的过程来看，`inpolygon` 这个函数运行得非常慢，在矩阵运算中就更慢了，因此导致整个程序也运行得非常慢。

(5) 基于 TV 模型修复的图像去反光

设置迭代次数为 300，去反光的結果如下：

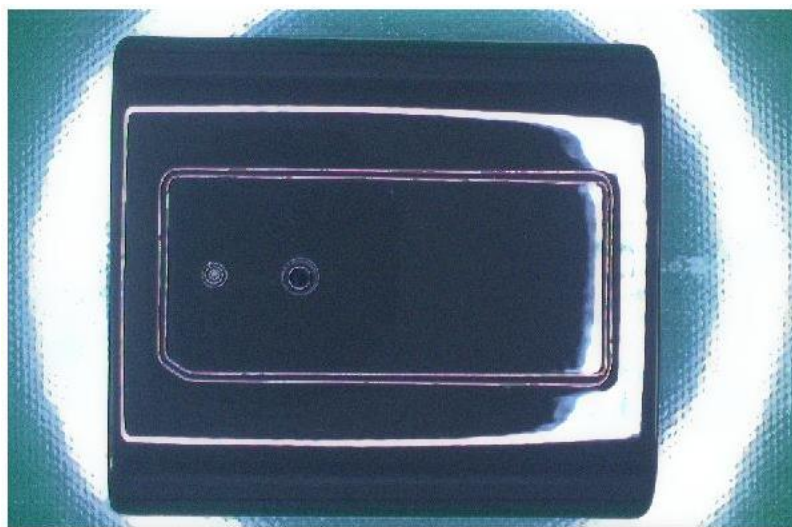


图 12 基于 TV 模型修复的去反光迭代 300 次结果

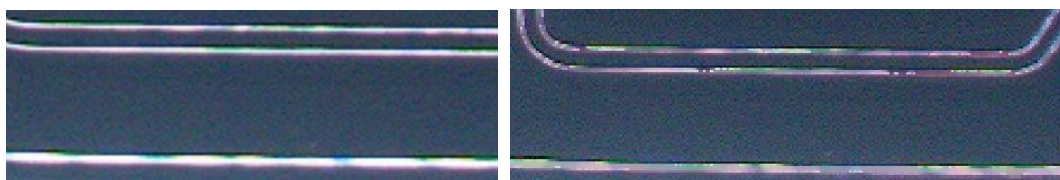


图 13 与原图（左图）的细节对比

从图 12 和 13 中我们可以看出，经过 300 次迭代修复之后，反光区域得到了非常细微的改善，这表明图像有待进一步的迭代修复。但是，由于 MATLAB 对于矩阵的运算比较慢，因此 300 次迭代已经耗时较长，如果设置更大的迭代次数，程序会执行得非常慢，而且可能会崩溃。

另外对基于 TV 模型的图像修复的算法进行分析之后我们发现，这种修复模式比较适合于去除反光区域比较小而多的图像，而对于本次作业中给出的面积较大的反光区域则不太适

合。

5 收获与总结

(1) 遇到的问题及解决办法

①在通过 `bwlabel` 这一函数寻找连通区的过程中，发现有很多小的连通区如下图：

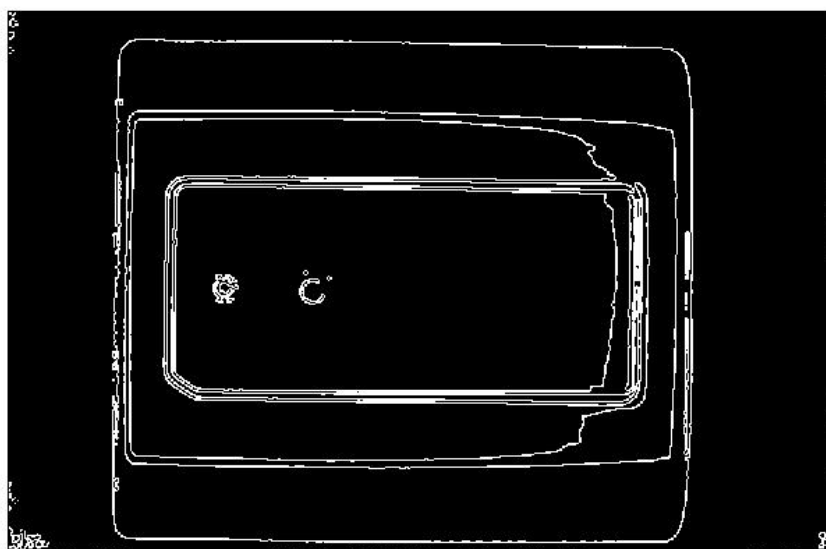


图 12 有小连通区的轮廓

为了去除这些小的连通区，可以通过设置连通区长度的阈值进行“过滤”。在多次调试参数之后，我发现这一阈值设置在 500 对于去除小的连通区比较有效，最终得到的结果亦如 4 中分析的那样得到了比较合适的轮廓。

②在上文中我提到过 `inpolygon` 这个函数的运行速度非常慢，由于是系统自带的函数我也没有什么办法，但是 `inpolygon` 是运行在矩阵运算中的，因此可以考虑减小矩阵的大小，从而提高运算速度。从第 3 步提取的工件轮廓中我们可以得到轮廓的四个角的坐标值，亦即轮廓上所有点的横纵坐标的最大、最小值，这四个值框出的区域相对于整张图片来说，缩小了 `inpolygon` 判断的区域大小，从而提高了运算速度。但是这一改善也只是微小的，最终程序运行得还是非常缓慢。

(2) 收获

通过这次的大作业,尤其是在用 TV 模型去反光的过程中,我提高了自己对于新方法的调研、理解能力和对于新问题的发现、解决能力,收获非常大。对于 MATLAB 和数字图像的了解也更加深入了。

参考文献:

[1] 刘鹏程. 图像修复技术去除肉品图像反光的应用研究, 2012-06