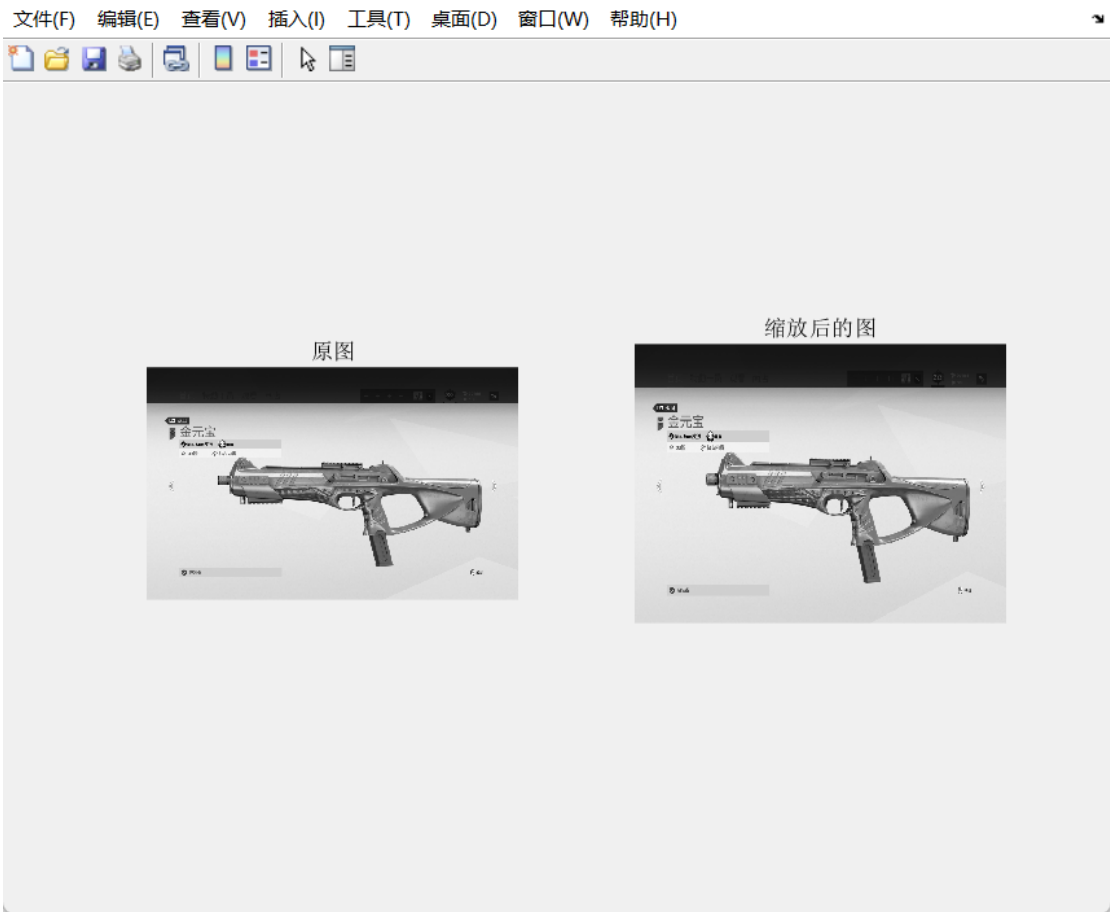


数字图像处理与技术作业

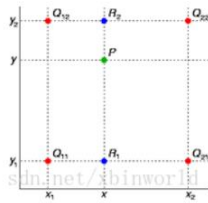
1、图像放大缩小 `resize.m`(代码实现的文件名)

先是计算行列的变化比例；再遍历新的图像，每个点都找到原图对应的点的像素值。
对于不在整数坐标的点，使用的是最临近插值方法，会选择最近遍历新的图像。



双线性插值

在数学上，双线性插值是有两个变量的插值函数的线性插值扩展，其核心思想是在两个方向分别进行一次线性插值[1]。见下图：



假如我们想得到未知函数 f 在点 $P = (x, y)$ 的值，假设我们已知函数 f 在 $Q_{11} = (x_1, y_1)$ 、 $Q_{12} = (x_1, y_2)$ 、 $Q_{21} = (x_2, y_1)$ 以及 $Q_{22} = (x_2, y_2)$ 四个点的值。最常见的情况， f 就是一个像素点的像素值。首先在 x 方向进行线性插值，得到

$$f(R_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}) \quad \text{where } R_1 = (x, y_1),$$

$$f(R_2) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}) \quad \text{where } R_2 = (x, y_2).$$

然后在 y 方向进行线性插值，得到

$$f(P) \approx \frac{y_2 - y}{y_2 - y_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2).$$

综合起来就是双线性插值最后的结果：

$$f(x, y) \approx \frac{f(Q_{11})}{(x_2 - x_1)(y_2 - y_1)} (x_2 - x)(y_2 - y) + \frac{f(Q_{21})}{(x_2 - x_1)(y_2 - y_1)} (x - x_1)(y_2 - y) \\ + \frac{f(Q_{12})}{(x_2 - x_1)(y_2 - y_1)} (x_2 - x)(y - y_1) + \frac{f(Q_{22})}{(x_2 - x_1)(y_2 - y_1)} (x - x_1)(y - y_1).$$

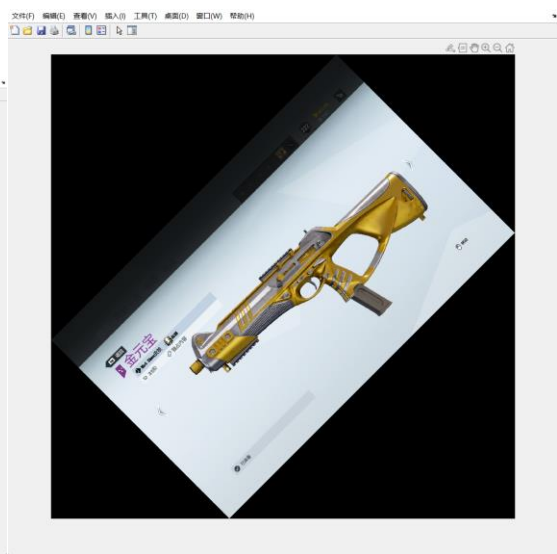
2、 图像旋转 `rotate_gray.m` `rotate_rgb.m`

先判断旋转角度，对于锐角可以直接旋转，大于 90° 的角度需要先旋转 90° ，再判断角度，如果还是大于 90° ，则需再次旋转 90° ，直到角度变为锐角。

首先计算旋转后的图像大小。如下图所示，这是图像旋转了 30° 后的图，用蓝色框示意原图位置。原图旋转 α 度，相反的，旋转后的图像只要旋转 $-\alpha$ 度就可以回到原图，但是起点的纵坐标相差了一段距离，图中红线表示，在旋转后的图上寻找对应原图的点时，需要先上移红色线段距离，再旋转 $-\alpha$ 度。红色线段长度为 $\text{column} * \sin(\alpha)$ 。

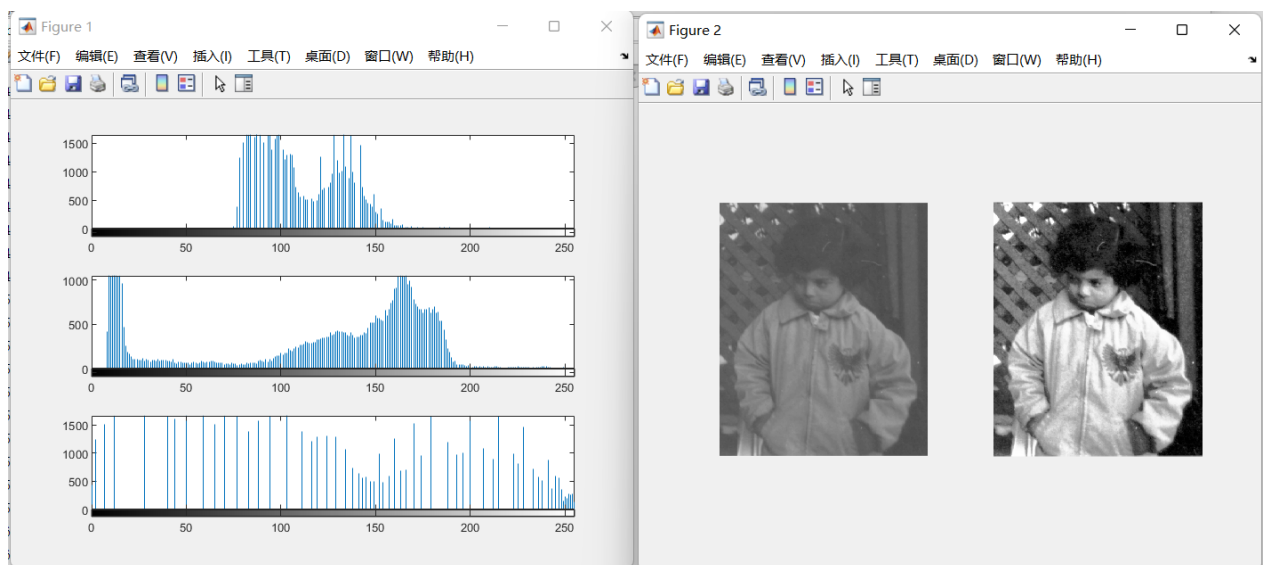
彩色图像和灰度图像一样地旋转方法，只是需要分别旋转 RGB 三个分量，最后再组合起来就可以了。彩色图像的程序我只编写了锐角的旋转，钝角的旋转与灰度图像相似，就没有写。

下面是我的一些测试图旋转后的效果。有灰色图像顺时针锐角 30° ，钝角 120° ，直角 90° ，大于 180° 的 210° 角；彩色图像旋转 45°



4、直方图规定化 `histogram_spec.m`

1. 对要规定化的原始图像进行直方图均衡化，得出 S_k
2. 对给定直方图进行直方图均衡化，得出 v_k
3. 遍历对每个 S 遍历 v ，找到最接近的 v ， v 所表示的映射关系即为 s 的映射关系。
4. 用修改过对应关系后的 s 生成图像 $g(x,y)$

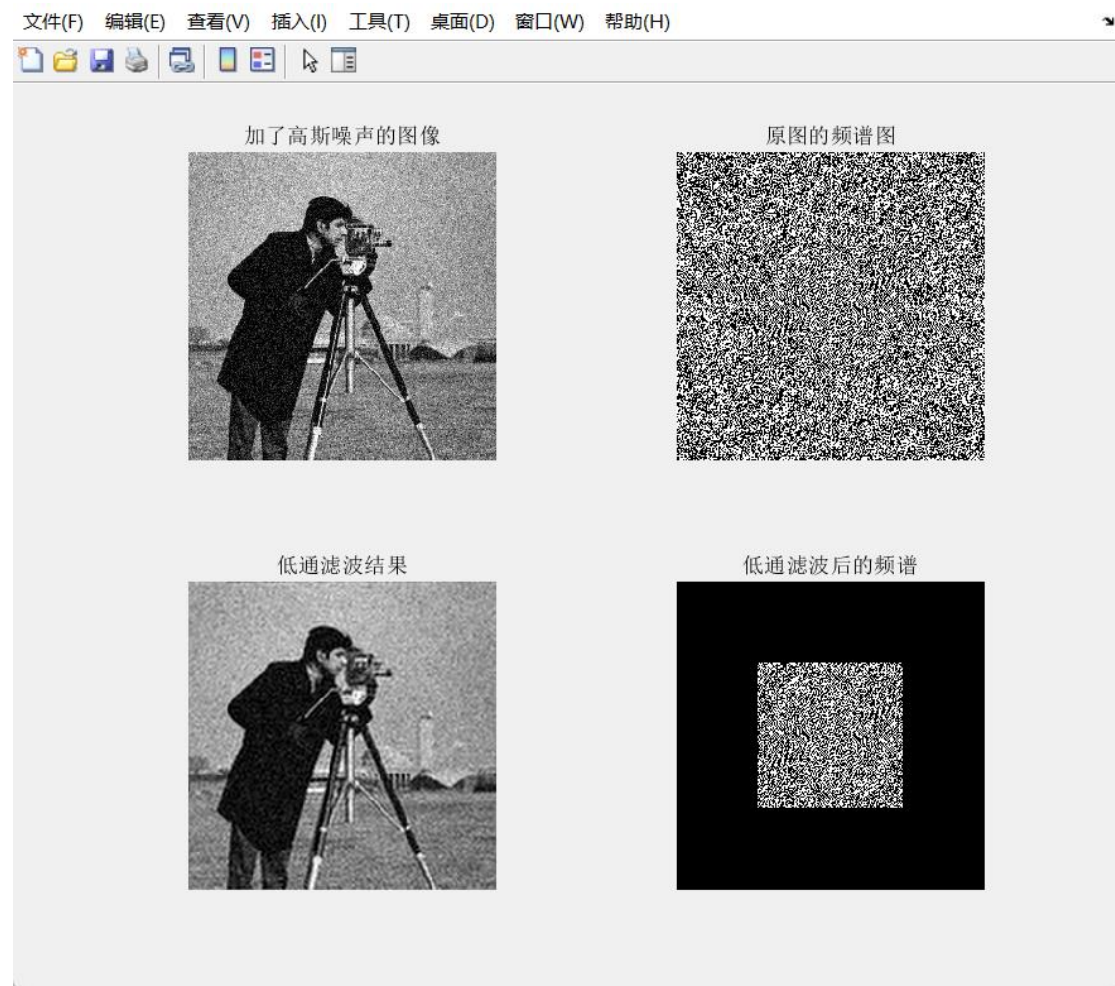


17、基于频域滤波的基本步骤 `fftFliter.m`

1. 将输入图片进行傅里叶变换，得到其频谱图
2. 保留低频的频谱分量，高频的删去，置为 0
3. 再将滤波后的频谱进行傅里叶反变换，得到输出图像

对于二维图像频谱图，其低频信号是图像中心，越靠近中心的频谱，对还原的图像影响越大，越边缘的频谱影响越小。

下面是我的结果图



20. 请写出 Canny 算子检测边缘的详细步骤。

1. 高斯滤波

滤波的主要目的是降噪，高斯滤波主要使图像变得平滑，同时也有可能增大了边缘的宽度。

2. 计算梯度值和梯度方向

边缘是灰度值变化较大的像素点的集合。一道黑边一道白边中间就是边缘，它的灰度值变化是最大的，在图像中，用梯度来表示灰度值的变化程度和方向。它可以通过点乘一个 Sobel 或其它算子得到不同方向的梯度值

综合梯度通过以下公式计算梯度值和梯度方向：

$$G(m, n) = \sqrt{g_x(m, n)^2 + g_y(m, n)^2}$$

$$\theta = \arctan \frac{g_y(m, n)}{g_x(m, n)}$$

3. 非极大值抑制

沿着梯度方向对幅值进行非极大值抑制，而非边缘方向。在高斯滤波过程中，边缘有可能被放大了。这个步骤使用一个规则来过滤不是边缘的点，使边缘的宽度尽可能为 1 个像素点：如果一个像素点属于边缘，那么这个像素点在梯度方向上的梯度值是最大的。否则不是边缘，将灰度值设为 0。

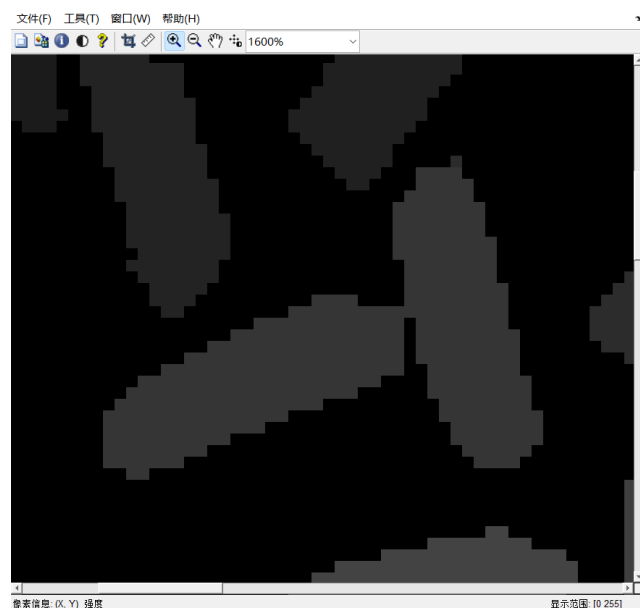
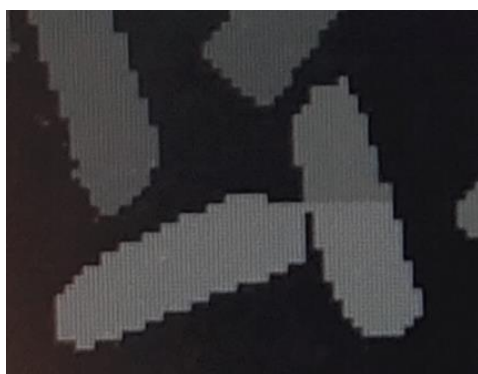
4. 用双阈值算法检测和连接边缘

它设置两个阈值，分别为 TH 和 TL。其中大于 TH 的都被检测为边缘，而低于 TL 的都被检测为非边缘。对于中间的像素点，如果与确定为边缘的像素点邻接，则判定为边缘；否则为非边缘。

21、区域标记 `zonemarker.m`

我采用的是逐行扫描的方法，每一行扫描时，遇到第一个像素 1 的点时给一个标记。再继续扫描，如果扫描的点像素还是 1，判断这个点的上面点是否为 0，不为 0 表示这一行 1 像素的点是和上一行连通，这时将标记改为上一行的像素值；为 0 则表示不连通。直到扫描到像素为 0 的点，代表这行连续的 1 像素结束了，如果标记不为 0，代表与上一行相连，需要将这行连续的 1 像素点修改为上一行的像素。

这样做完一遍后会有同一个区域有两个颜色的情况，如下左图所示，这和老师上课说的“山”的例子一样扫描完第二竖、第三竖的像素与第一竖、横的像素不一样，老师说的集合方法 MATLAB 我没弄好，我就再换了个方法。我再扫描一遍图像，但方向是从下往上扫描，和第一次一样的方法，判断扫描的行是否和下面一行连通，连通则改为下一行的像素，否则保持不变。



下图是我编写的程序结果：



22、边界跟踪 `boundary_following.m`

对图像按行扫描，找到第一个值为 1 的点，以这个点为起点，记为 P_0 ，再定义 8 个搜索方向

按逆时针方向顺序依次判断当前点的 8 个邻接点是否为 1，找到第一个为 1 的点，这个点为边界的下一点。再从这个点开始寻找下一个边界点

直到找到的下个点的坐标是 P_1 的坐标，并且它前一个点坐标是 P_0 的坐标为止，表示已经循环了一圈，最后再顺序输出边界坐标。

下左图是原图，下右图是边缘图：

