

## Document

16302010059 张健

标签（空格分隔）：多媒体技术

---

### 一、获取图片RGB值

描述：

- 给定一张图片
- 获取点(x,y)，以及(x,y)周围8个点的R、G、B值

要求：

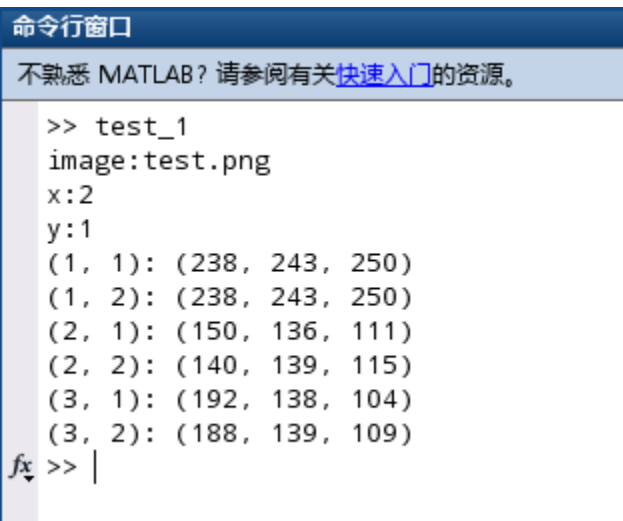
- 允许用户指定图片的路径、坐标。做好边界检查

难点：边界检查

实现：通过max()和min()限定边界

- 当取点超过下边界，则取1
- 当取点超过上边界，取数组长度
- [x] 代码：

```
for m = max(px - 1, 1) : min(px + 1, x)
    for n = max(py - 1, 1) : min(py + 1, y)
```



```
命令窗口
不熟悉 MATLAB? 请参阅有关快速入门的资源。

>> test_1
image:test.png
x:2
y:1
(1, 1): (238, 243, 250)
(1, 2): (238, 243, 250)
(2, 1): (150, 136, 111)
(2, 2): (140, 139, 115)
(3, 1): (192, 138, 104)
(3, 2): (188, 139, 109)
>> |
```

输出图示：

## 二、实现rgb2gray函数

描述：

- 使用matlab实现rgb2gray的功能
- 计算出转换后灰度图片的方差

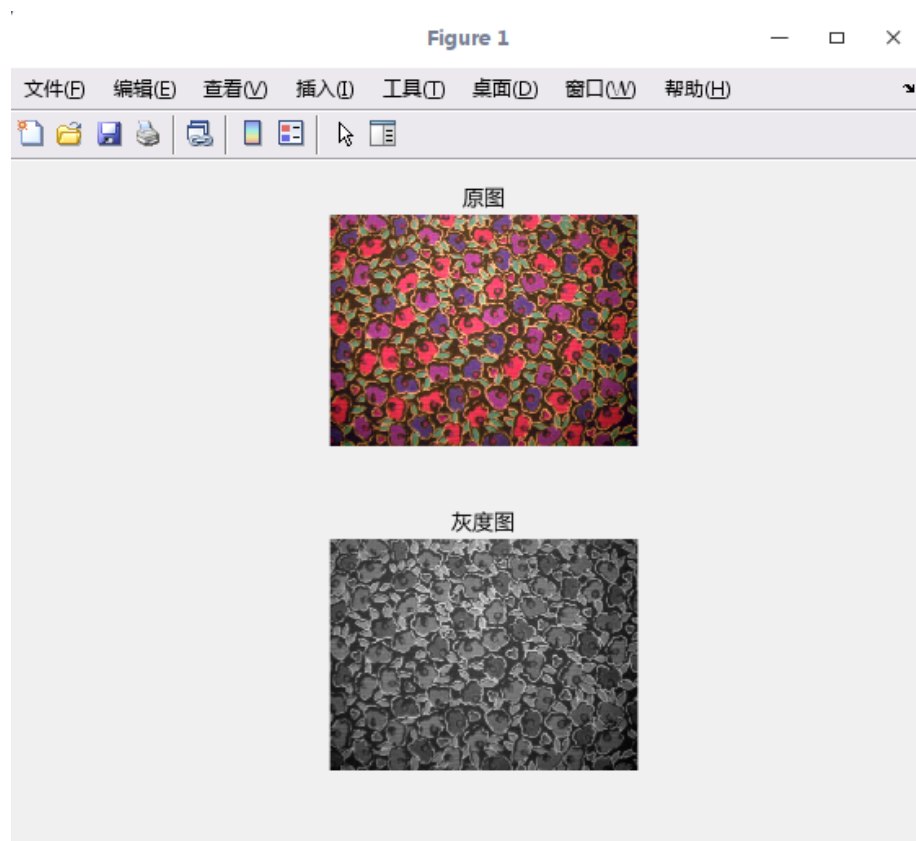
要求：

- 可以使用matlab自带图片，也可自选图片
- 自选图请放在项目当前目录
- 输出的文件保存到当前目录即可

实现：

- 灰度值计算公式： $\text{Gray} = 0.29900 * R + 0.58700 * G + 0.11400 * B$
- 将图片所有点经过灰度值计算公式转换成灰度值
- [x] 代码：

```
grayI = 0.29900 * I(:, :, 1) + 0.58700 * I(:, :, 2) + 0.11400 * I(:, :, 3);
```



输出图示：

### 三、灰度图对比度增强

描述：

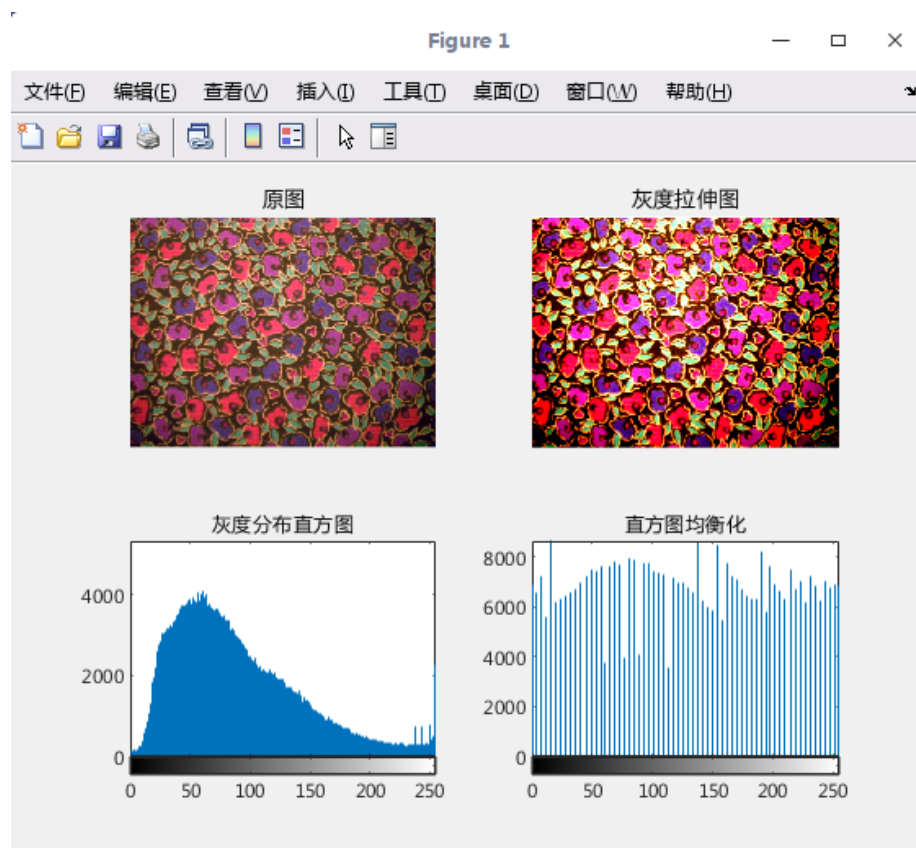
- 分别使用灰度拉伸和直方图均衡化，增强给定图的对比度

要求：

- 需给出增强前后对比图，以及灰度值分布图
- 通过结果，谈谈你对灰度拉伸和直方图均衡化的理解

实现：

- 调用内置函数`imadjust`实现灰度拉伸
- 调用内置函数`histeq`实现直方图均衡化
- 调用内置函数`imhist`得到灰度分布直方图



输出图示：

理解：灰度拉伸：

- 即改变图片对比度，但不破坏原有像素的排序

直方图均衡化：

- 把一副图像的灰度值进行拉伸，使得灰度值可以分布在更多的灰度级上
- 对图像进行非线性拉伸,重新分配图像像素值,使一定灰度范围内象元值的数量大致相等
- 均衡化的结果会使得直方图分布更均匀（本来就是这个目的），以此来提高局部对比度

#### 四、图像去噪

描述：

- 对给定图片加椒盐噪声
- 至少选择两个你了解的去噪算法去除椒盐噪声

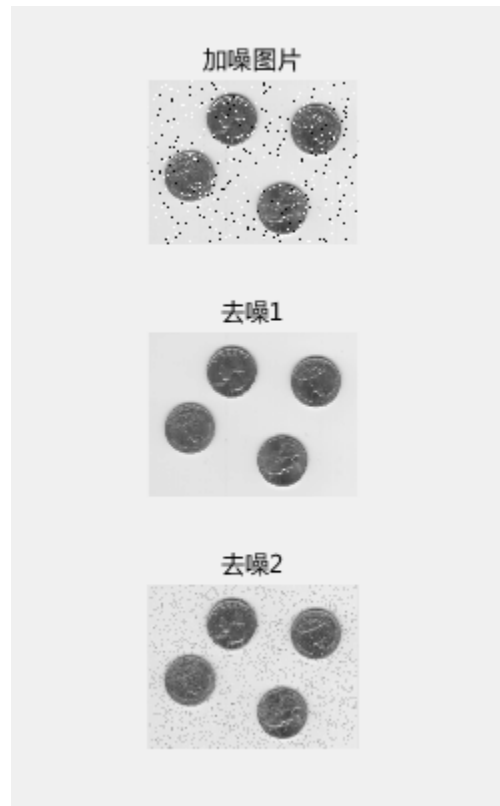
要求：

- 给出去噪后的结果对比图
- 通过算法分析造成这种差异的原因

实现：

- 调用内置函数`imnoise`给图片加噪，叠加密度为0.04的椒盐噪声
- 去噪方式一：
  - 取以(x, y)为中心的 $3 \times 3$ （边界检查）个点的中位数作为(x, y)的值
  - 即用几个像素的灰度平均值来代替一个像素的灰度
- 去噪方式二：
  - 取以(x, y)为中心的 $3 \times 3$ （边界检查）个点的平均数作为(x, y)的值
- [x] 代码：

```
J2 = imnoise(I, 'salt & pepper', 0.04);
filter1 = medfilt2(J2, [3 3]);
h = fspecial('average', [3 3]);
filter2 = imfilter(J2, h);
```



输出图示：

分析去噪原理及误差原因：

方式一：中值法—中值滤波

- 首先确定一个以某个像素为中心点的邻域
- 将邻域中各像素的灰度值进行排序
- 取中间值作为中心像素灰度的新值
- 中值滤波的输出像素是由邻域图像的中间值决定的
- 因而中值滤波对极限像素值(与周围像素灰度值差别较大的像素)远不如平均值那么敏感
- 从而可以消除孤立的噪声点,可以使图像产生较少的模糊

方式二：平均值法

- 将一个像素及其邻域中所有像素的平均值赋给输出图像中相应的像素
- 即用几个像素的灰度平均值来代替一个像素的灰度
- 其主要的优点是算法简单、计算速度快
- 但其代价是会造成图像一定程度的模糊