HW4：图像复原和彩色图像处理

15331416 赵寒旭

1. 习题

1.1 彩色空间

1. （12 分）

Ans：彩色图像 RGB 通道图，从 0-1 对应亮度由黑到白。单个通道图体现为标量矩阵，显示效果为灰度图，每个像素点的灰度值由原图对应通道分量决定。

$$\mathrm{Gray} = 0.299 \times \mathrm{R} + 0.587 \times \mathrm{G} + 0.114 \times \mathrm{B}$$

某一位置上，某一通道的灰度值越大，其对应的色彩分量在最终的彩色图像体现上所占的比重就越大。

（a）B 通道图。海绵宝宝眼睛是蓝色，在蓝色通道上应有最大值。

（b）G 通道图。海绵宝宝的主体黄色由 G 和 R 分量共同体现，易得 R 通道图为 d，在 b, c 中确定 G 通道图：b 图中领带位置值很小，接近黑色，判断为 G 通道图。

（c）灰度图。红色领带位置和蓝色眼珠位置均为灰色。

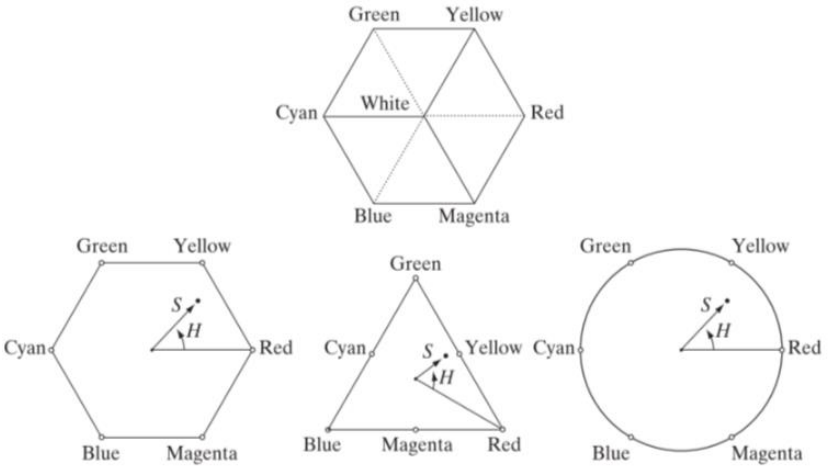
（d）R 通道图。领带为红色，领带位置在红色通道 R 上应有最大值。

分析 b 和 c 相似的原因：

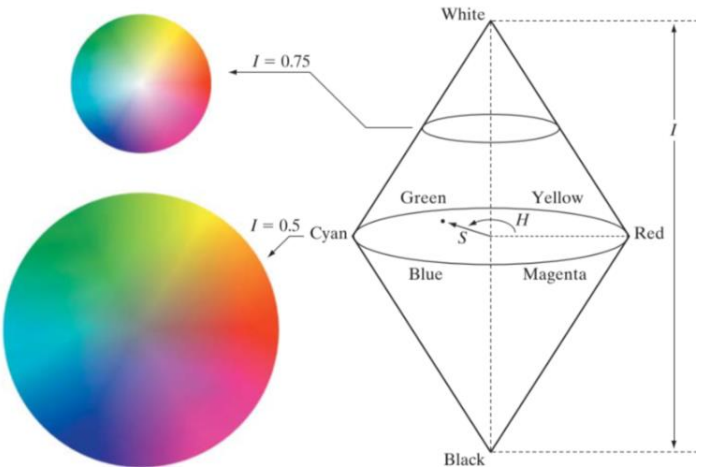
$\mathrm{Gray} = 0.299 \times \mathrm{R} + 0.587 \times \mathrm{G} + 0.114 \times \mathrm{B}$

由公式可知，G 通道分量权值最大，生成灰度图会相对更接近 G 通道图。

2. （8 分）



Ans：H 通道+60°，黄色到绿色，蓝色到洋红，红色到黄色，白色仍为白色。转换后为(b)。

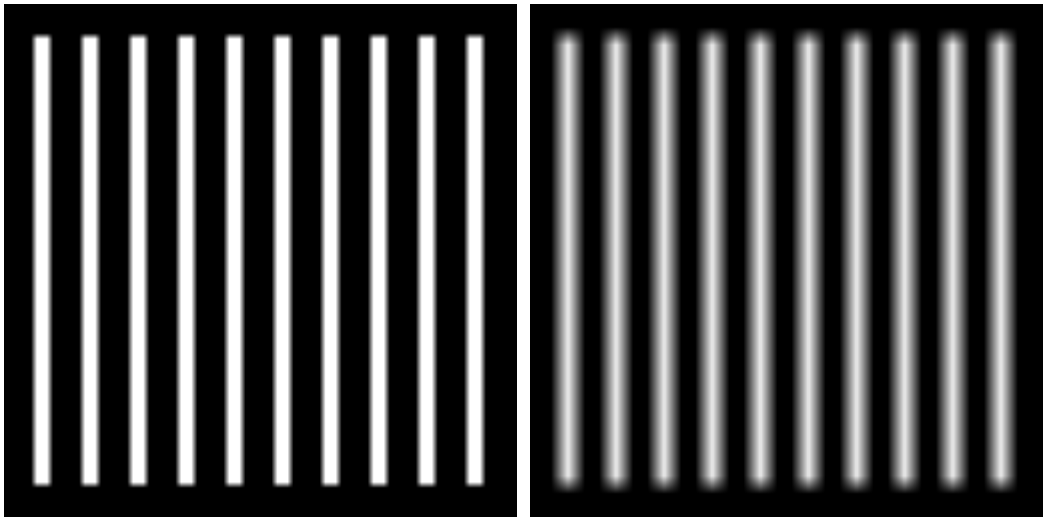


2. 编程题

2.2 图像滤波（10 分）

目标 图 4 中的白条宽 8 像素、高 224 像素，白条之间间隔 16 个像素。其他细节请参考你的输入图片。

1. （2 分）用 3 × 3 和 9 × 9 的算术均值滤波器(arithmetic mean filter)对输入图像做滤波，并把相应的两个结果贴在你的报告里。同时，你需要在报告中简要描述这两个滤波结果的特点，比如白条的宽/高/颜色等。（答案中只取[r g b] = [1 1 1]为白条）



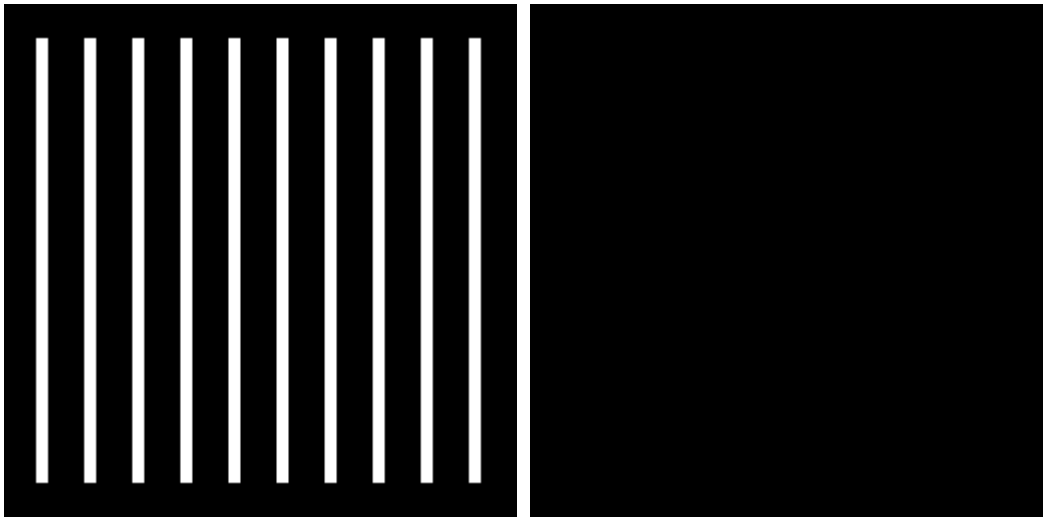
左图为 3 × 3 结果：

白条宽 6 像素，长 220 像素，白条之间间隔 18 个像素，边界模糊，边角为直角。

右图为 9 × 9 结果：

此时无绝对白条，边界模糊，边角近似弧形。

2. （4 分）用 3 × 3 和 9 × 9 的调和均值滤波器(harmonic mean filters)对输入图像做滤波。把相应的两个结果贴在你的报告里，同时简要描述每个滤波结果的特点。



左图为 3 × 3 结果：

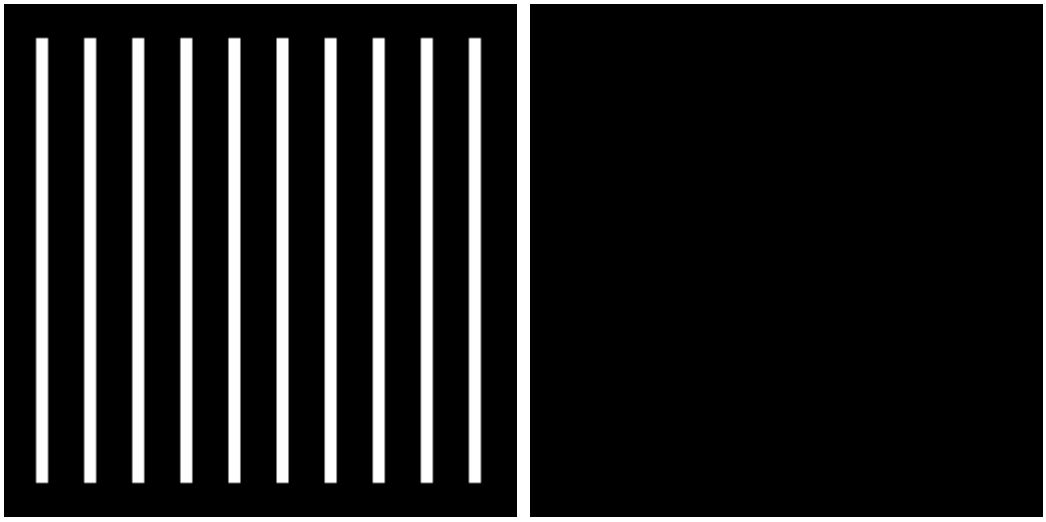
白条宽 6 像素，长 222 像素，白条之间间隔 18 个像素，边界清除，边角为直角。

右图为 9 × 9 结果：

白条消失。

3. （4 分）用 3 × 3 和 9 × 9 的谐波均值滤波器(contra-harmonic mean filters)对输入图像做

滤波（Q=-1.5）。把相应的两个结果贴在你的报告里，同时简要描述每个滤波结果的特点。



左图为 3 × 3 结果：

白条宽 6 像素，长 222 像素，白条之间间隔 18 个像素，边界清除，边角为直角。

右图为 9 × 9 结果：

白条消失。

2.3 图像去噪（40 分）

1. （5 分）实现一个噪声生成器。

见 mynoise.m

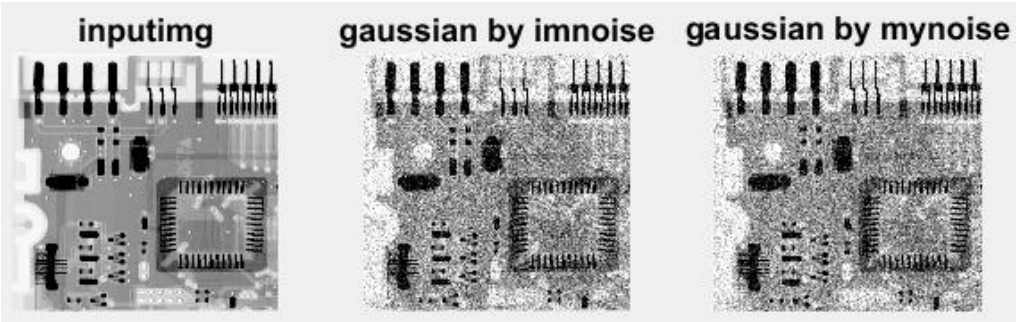
```
function output_img = mynoise( input_img, type, a, b)
% 输入图像加噪声后输出
%    高斯噪声 a：均值，b:方差
%    椒盐噪声 a: 椒(0)噪声概率，b：盐(1)噪声概率
```

可以指定高斯噪声的噪声均值和标准差和椒盐噪声中两个噪声成分各自的概率。

2. （10 分）高斯噪声

1）对输入图片添加均值为 0，标准差为 40 的高斯噪声。

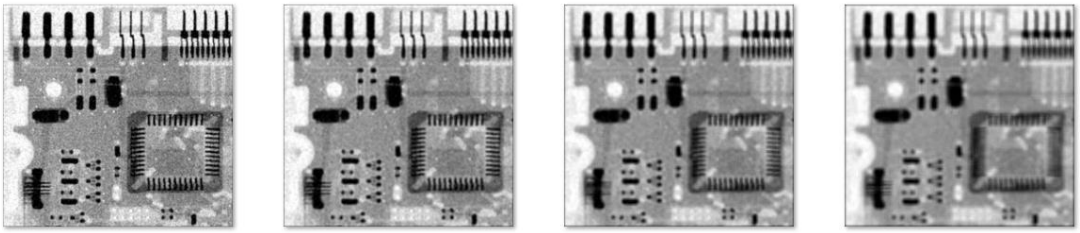
2）用算术均值滤波、几何均值滤波和中值滤波分别对图片去噪



选择主观判断较为清晰的去噪结果展示，滤波类型和滤波器尺寸如图所示。(均取 5x5)

3）分析各个滤波效果的优劣，并说明理由。

（1）算术均值滤波



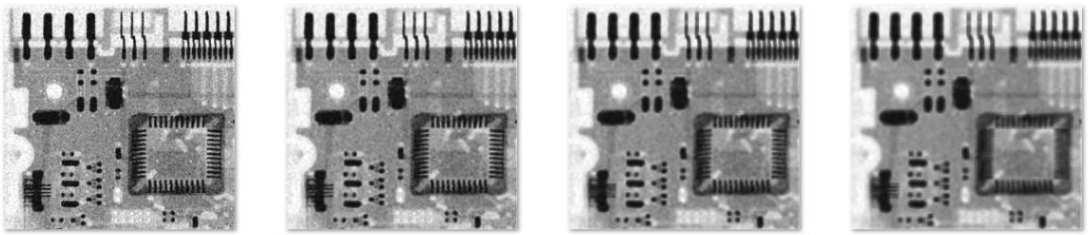
arithmetic_3x3.png　arithmetic_5x5.png　arithmetic_7x7.png　arithmetic_9x9.png

每个复原的像素由滤波器尺寸确定的子图象窗口中所有像素取均值得到。

优点：主要起到平滑模糊的效果，算数均值滤波平滑一幅图像中的局部变化，对于高斯噪声或者均匀随机噪声的处理比较合适。

缺点：滤波器尺寸小时去噪效果欠佳，滤波器尺寸增大会造成较大的细节模糊，不利于图像恢复。

（2）几何均值滤波

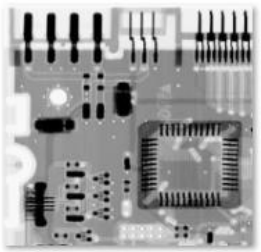


geometric_3x3.png　geometric_5x5.png　geometric_7x7.png　geometric_9x9.png

每个复原的像素由滤波器尺寸确定的子图象窗口中所有像素乘积再开$1/mn$次方得到。

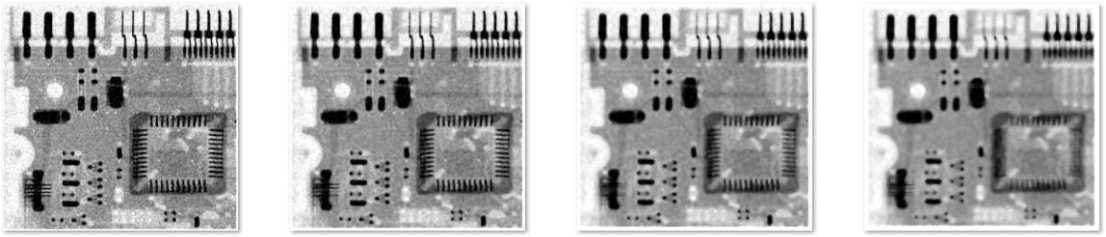
优点：与算术均值滤波同样，对高斯噪声的处理是合适的，平滑程度相当，但几何均值处理中丢失的图像细节更少。

缺点：若子图象窗口中有一个像素值为黑点，会造成整个窗口乘积为 0，即恢复图像像素显示为黑色，随着滤波器尺寸的增大，黑边黑线变粗的现象会逐渐严重。（右图为 task2.png 原图，与 geometric_9x9.png 对比可以明显看出黑边加粗的现象）



task_2.png

（3）中值滤波



median_3x3.png　median_5x5.png　median_7x7.png　median_9x9.png

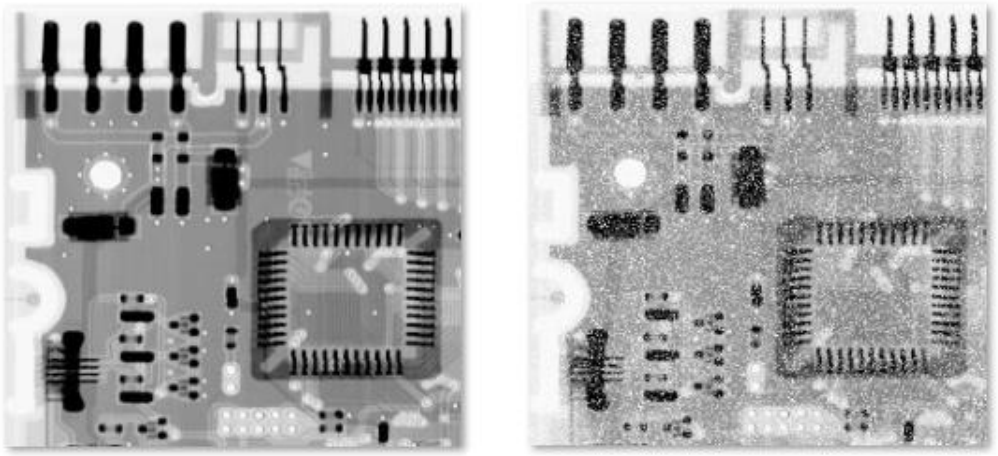
使用一个像素邻域中的灰度级的终止来替代该像素的值。（统计滤波器，非线性）

优点：对于某些类型的随机噪声，可提供更好的去噪能力，且比相同尺寸的线性平滑滤波器引起的模糊更少。在存在单极或双极脉冲时，中值滤波器尤其有效。

缺点：中值滤波对图像重复处理会使图像变模糊，希望保持尽可能低的处理次数，但要彻底去除噪声，往往要重复处理数次。

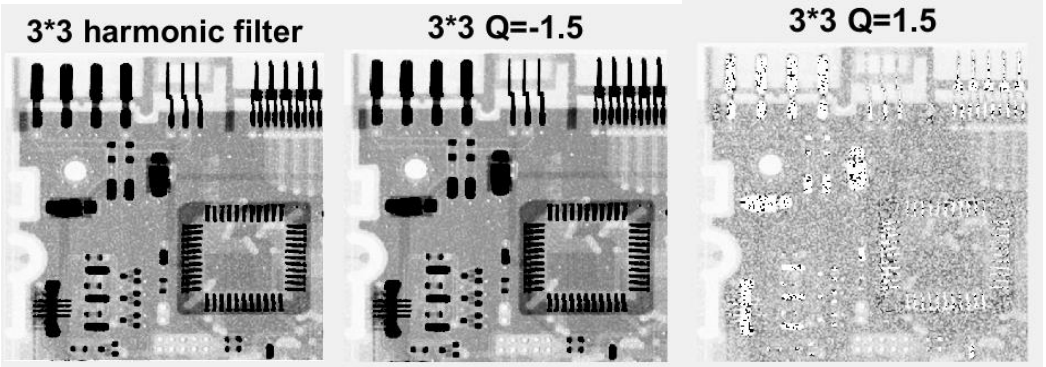
3. （10 分）盐噪声

1）给输入图片添加概率为 0.2 的盐噪声。



task_2.png　salt.png

2）分别用调和均值和谐波均值滤波来做处理（谐波均值要展示 Q 为正负的两种情况）



（选择合适的滤波结果展示，具体尺寸已标于 figure 中）

3）讨论：为什么错误的 Q 值会导致糟糕的结果？

$$\hat{f}(x,y)=\frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}}$$

Q 的选择是有依据的，

（1）Q>0：去除椒噪声

复原像素点以椒噪声（值为 0）为中心选择邻域时，由公式可知当噪声点周围有一些较亮的点时，中心像素对S_{xy}中像素求和影响不大，恢复像素的值将接近邻域中像素的值。

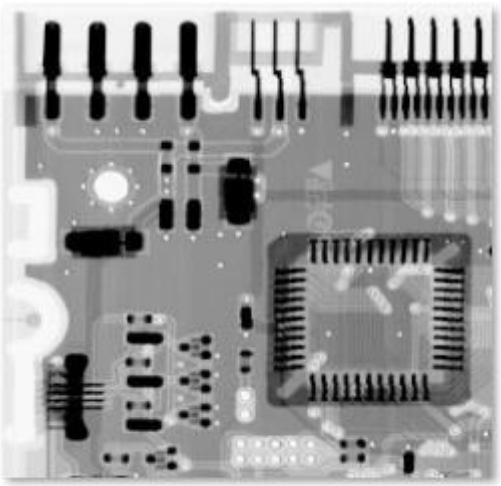
（2）Q<0：去除盐噪声

复原像素点以盐噪声（值为 1）为中心选择邻域时，由公式可知指数为负数时，较小的数在求和结果中起主导作用，恢复像素的值将接近邻域中像素的值。

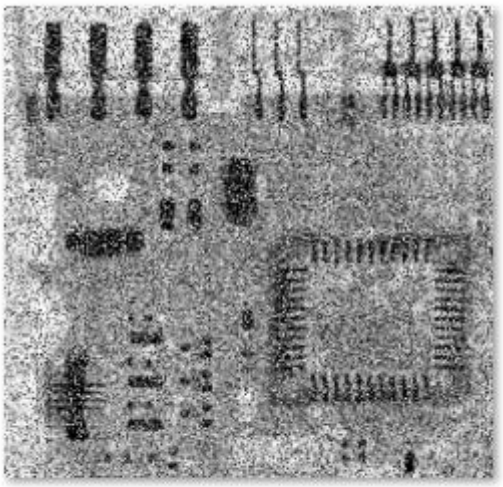
综合以上讨论，如果 Q 值选择错误，刚好在每种情况下都是噪声值在复原像素点的过程中起主导作用，导致有椒噪声的图像复原后变黑，有盐噪声的图像复原后变亮（见上图 Q=-1.5 的情况），不能达到去噪的效果。

4. （10 分）

1）给输入图片添加椒盐噪声（椒噪声和盐噪声的概率都为 0.2）。

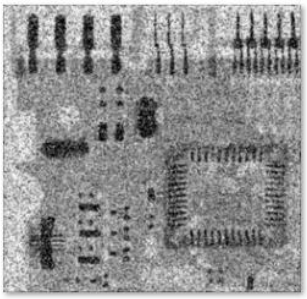


task_2.png

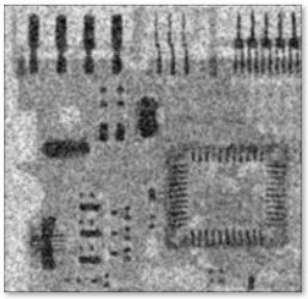


salt-and-pepper.png

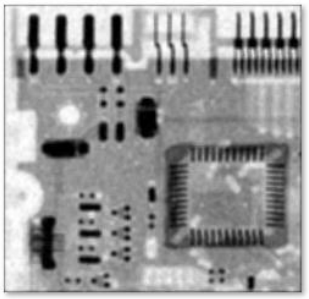
2）用算术均值滤波，几何均值滤波，最大值滤波，最小值滤波和中值滤波分别去噪。



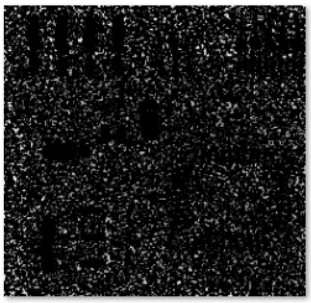
arithmetic_3x3.png



arithmetic_5x5.png

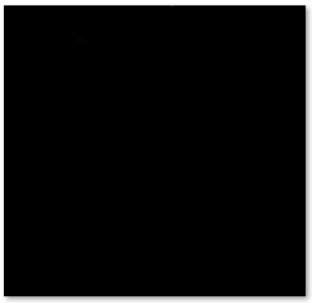


arithmetic_7x7.png

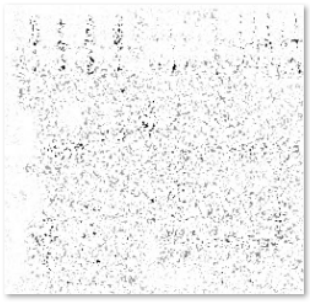


geometric_3x3.png

geometric_5x5.png

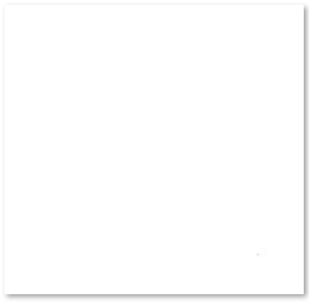


geometric_7x7.png

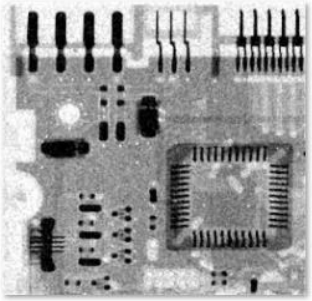


max_3x3.png

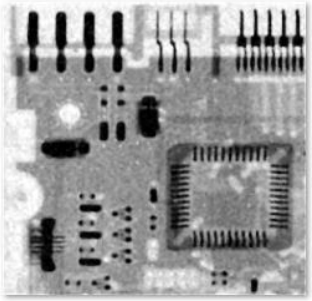
max_5x5.png



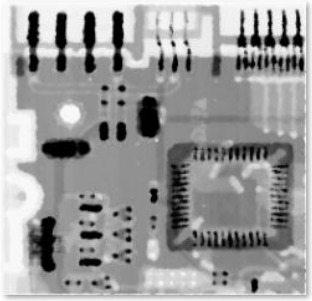
max_7x7.png



median_3x3.png



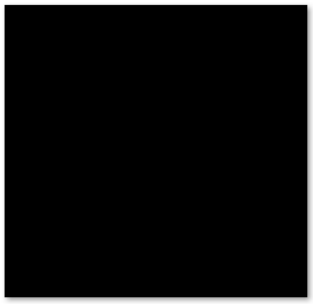
median_5x5.png



median_7x7.png

min_3x3.png

min_5x5.png



min_7x7.png

3）分析结果，哪一个滤波的结果更好或更差，为什么。

由上述滤波结果易知，中值滤波的结果最好，算术均值效果较差，其余滤波器效果极差。原因分析：

（1）算术均值滤波

主要起到平滑模糊的效果，算数均值滤波平滑一幅图像中的局部变化，对于高斯噪声或者均匀随机噪声的处理比较合适，对这种概率较大的双极脉冲噪声要想去除一定程度的噪声，代价就是图像模糊程度较大。

增大滤波器尺寸后，去噪效果提升，但图像细节基本已经模糊了。

（2）几何均值滤波

基本特性与算术均值滤波相近。

特别地，有黑点扩大黑边加粗的特点，对出现概率较高的椒盐噪声，有 0.2 的椒噪声（值为 0），会造成图像大范围变黑。并随着滤波器尺寸增加变黑程度加重。

（3）最大值滤波

遍历待恢复图像，对应每个子图象窗口选择最大值，在盐噪声（值为 1，最大）出现概率为 0.2 的情况下，最大值滤波会使整个图像趋于白色。并随着滤波器尺寸增加变白程度加重。

（4）中值滤波

效果较好。

中值滤波在存在单极或双极脉冲时，尤其有效，它可以过滤掉子图象窗口中的最大值和最小值（即盐噪声和椒噪声），使恢复像素点的值反映邻域内未被噪声污染的信息。

（5）最小值滤波

和最大值滤波同样。对应每个子图象窗口选择最小值，在椒噪声（值为 0，最小）出现概率为 0.2 的情况下，最小值滤波会使整个图像趋于黑色。并随着滤波器尺寸增加变黑程度加重。

5. （5 分）滤波操作实现详解

令S_{xy}表示中心点在(x,y)处，大小为$m \times n$的矩形子图象窗口（邻域）的一组坐标。

1）算术均值滤波

理论：算术均值滤波器在S_{xy}定义的区域中计算被污染图像$g(x,y)$的平均值，在点(x,y)处复原图像$\hat{f}$的值，就是简单使用S_{xy}定义的区域中的像素计算出的算术均值。

$$\hat{f}(x,y) = \frac{1}{mn}\sum_{(s,t)\in S_{xy}} g(s,t)$$

算法：

✧ 用大小为$m \times n$的滤波器（元素值均为 1）对输入图像滤波，相当于求$\sum_{(s,t)\in S_{xy}} g(s,t)$矩阵，矩阵每个元素再乘公式中$1/mn$因子即为滤波后图像。

```
input_img = im2double(input_img);
filter = ones(m,n);
output_img = filter2d( input_img, filter)./(m*n);
```

2）几何均值滤波

理论： 每个复原的像素由子图象窗口中像素的乘积的$1/mn$次幂给出。

$$\hat{f}(x,y) = \left[\prod_{(s,t)\in S_{xy}} g(s,t)\right]^{\frac{1}{mn}}$$

算法：

✧ 输入图像 1 扩展至$(2*(m-1)+M)\times(2*(n-1)+N)$大小。

✧ 遍历扩展后矩阵，对每个S_{xy}求所有元素乘积的$1/mn$次幂。

✧ 输出图像截取原来大小的合适区域。$(m-1:M+m-2, n-1:N+n-2)$

3）调和均值滤波

算法：

✧ 原矩阵每位取倒数与滤波器矩阵滤波（相当于倒数求和），由计算出的分母计算出像素的值

$$\hat{f}(x,y) = \frac{mn}{\sum_{(s,t)\in S_{xy}} \frac{1}{g(s,t)}}$$

```
output_img = m * n ./ filter2d(1./(input_img + eps),ones(m, n));
```

4）谐波均值滤波

算法：

$$\hat{f}(x,y)=\frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}}$$

- ✧ 输入图像矩阵每位取 Q+1 次幂，用 ones 滤波器矩阵滤波（相当于求和）
- ✧ 输入图像矩阵每位取 Q 次幂，用 ones 滤波器矩阵滤波（作为分母+eps 防止为 0）
- ✧ 分子分母按位相除

```
output_img = filter2d(input_img.^(Q+1), ones(m, n));
output_img = output_img ./ (filter2d(input_img.^Q, ones(m, n)) + eps);
```

5）最大值滤波

算法：

- ✧ 输入图像 1 扩展至$(2*(m-1)+M)\times(2*(n-1)+N)$大小。 $\hat{f}(x,y)=\max_{(s,t)\in S_{xy}}\{g(s,t)\}$
- ✧ 遍历扩展后矩阵，对每个S_{xy}求最大值。
- ✧ 输出图像截取原来大小的合适区域。$(\mathrm{m}-1{:}\,\mathrm{M}+\mathrm{m}-2,\mathrm{n}-1{:}\,\mathrm{N}+\mathrm{n}-2)$

6）最小值滤波

- ✧ 输入图像 0 扩展至$(2*(m-1)+M)\times(2*(n-1)+N)$大小。 $\hat{f}(x,y)=\min_{(s,t)\in S_{xy}}\{g(s,t)\}$
- ✧ 遍历扩展后矩阵，对每个S_{xy}求最小值。
- ✧ 输出图像截取原来大小的合适区域。$(\mathrm{m}-1{:}\,\mathrm{M}+\mathrm{m}-2,\mathrm{n}-1{:}\,\mathrm{N}+\mathrm{n}-2)$

7）中值滤波

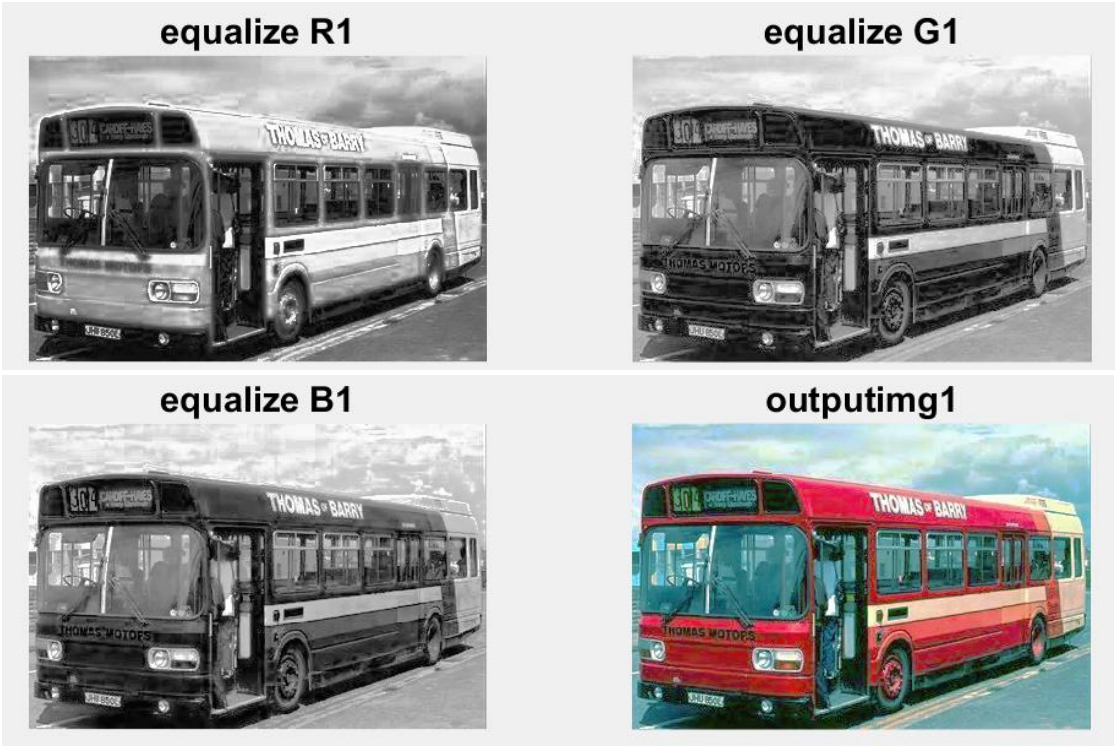
- ✧ 与最大值滤波步骤，区别在于第二步对S_{xy}取中值。 $\hat{f}(x,y)=\underset{(s,t)\in S_{xy}}{\mathrm{median}}\{g(s,t)\}$

2.4 彩色图像的直方图均衡化（30 分）

1. （6 分）

1）分别对 RGB 三个通道进行直方图均衡化

2）将处理后的三通道重构成一张 RGB 图

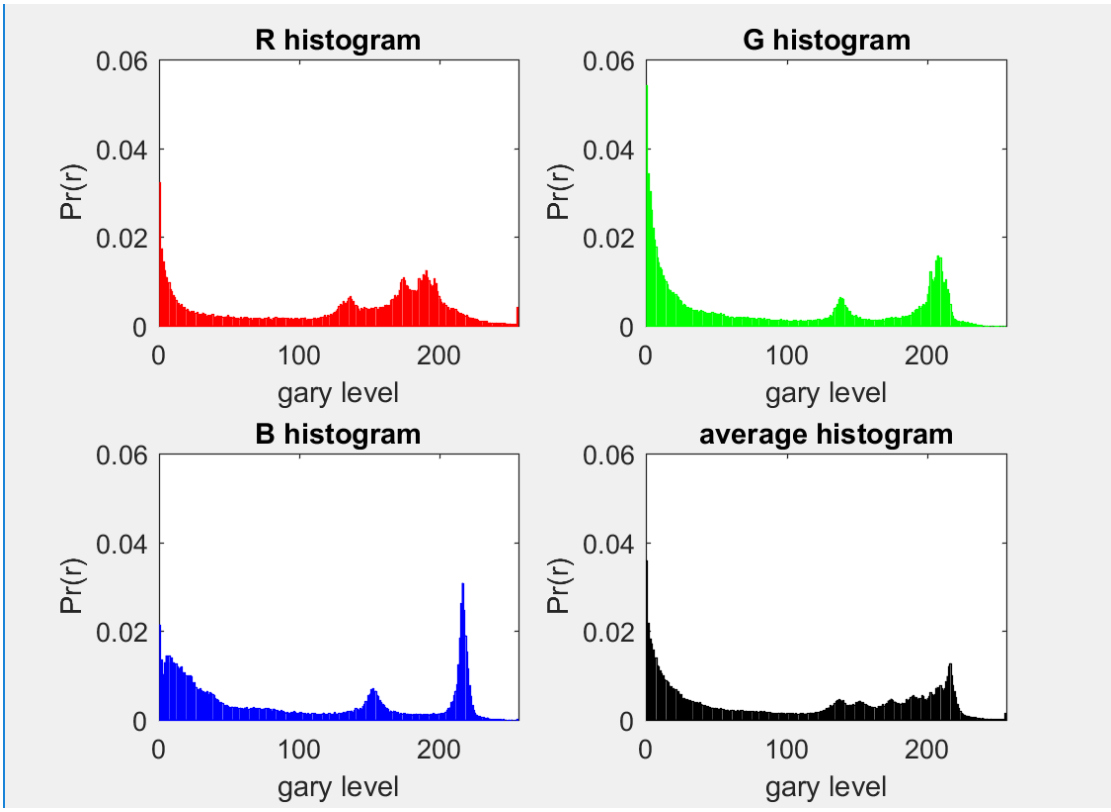


原图与重构 RGB 图 output_img1 对比：

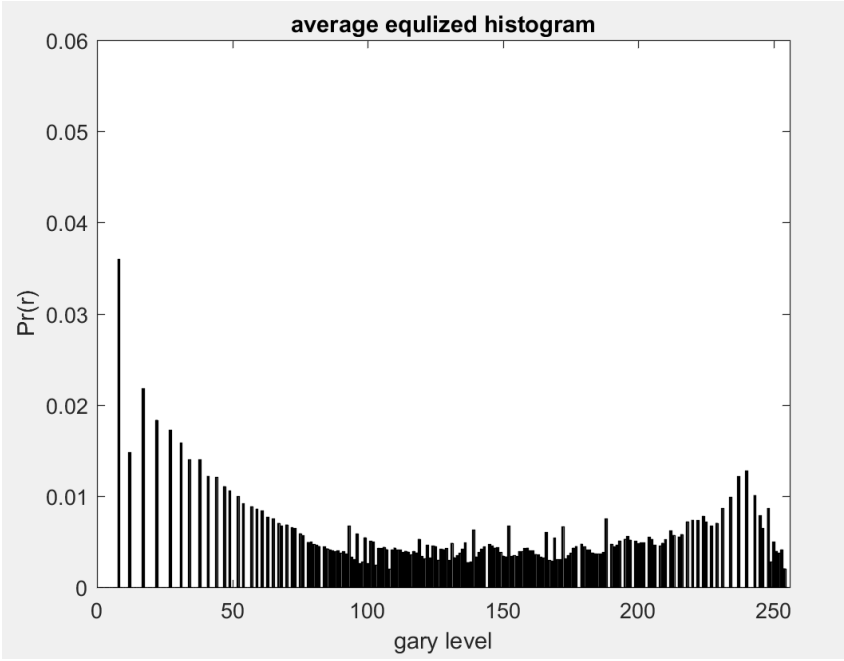
2. （6 分）

1）分别计算每一个通道的直方图

2）对这三个直方图取平均值得到一个平均直方图



3）对这个直方图做均衡化



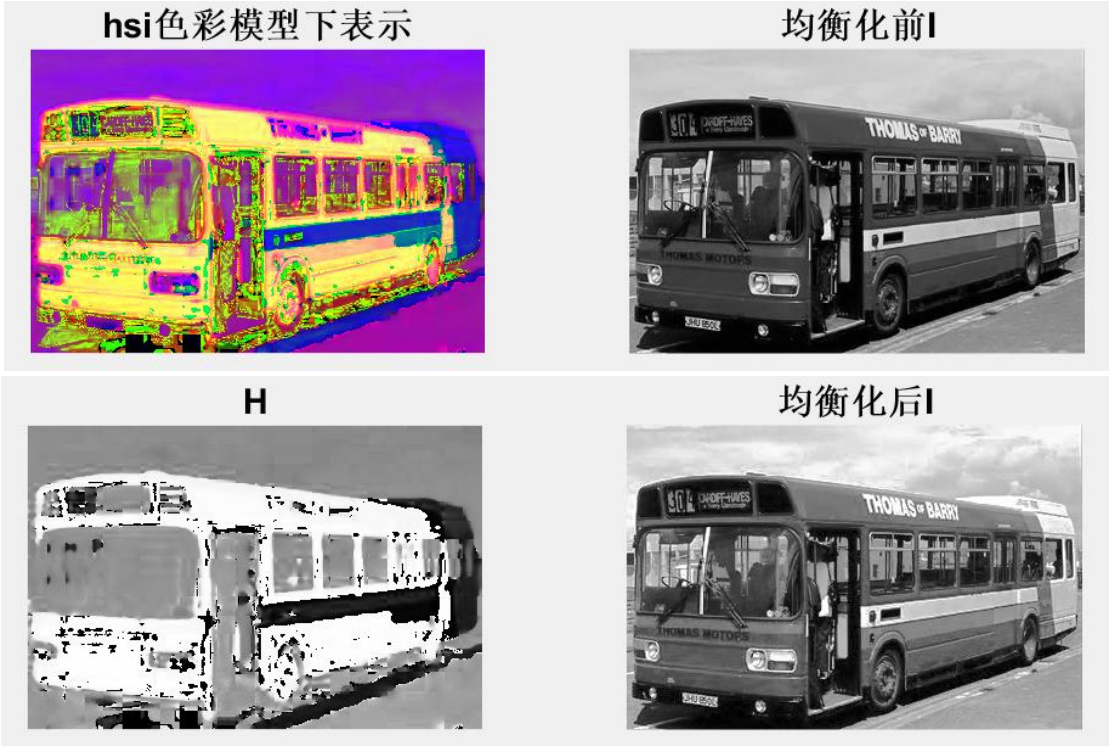
4）把平均直方图均衡化前后的像素值映射关系应用到 RGB 三个通道

5）再重构一张 RGB 图

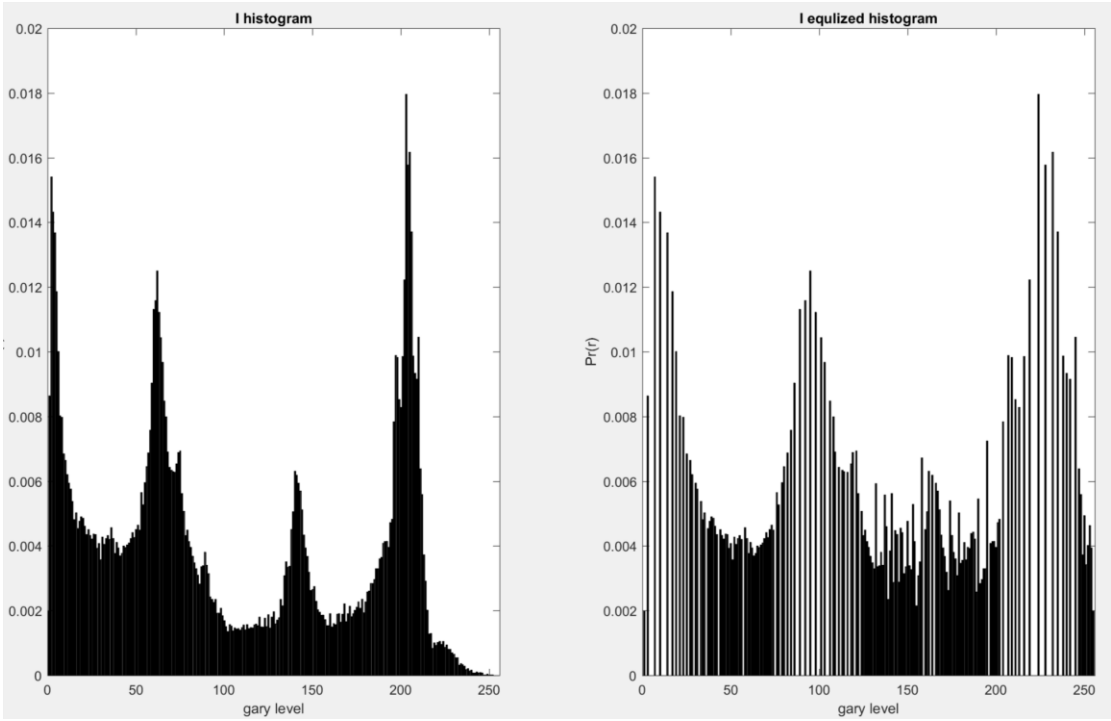
原图与重构 RGB 图 output_img2 对比：

3. （6 分）

1）将输入图片转到 HSV 色彩空间



2）对强度（intensity）进行直方图均衡化



3）将处理后的结果转换到 RGB 色彩空间

原图与重构的 RGB 图 output_img3 对比：

4. （12 分）比较上面得到的三个结果，说出他们的不同之处，解释其中的原因。

不同之处：

（1）分别对三个通道进行直方图均衡化再合并起来。

原图中的阴暗部分变得明亮起来，但造成了较大的颜色失真，色彩比较灰暗。（观察可见车窗和车底等暗色区域有明显失真）

（2）对三通道均值进行直方图均衡化，再用得到的变换关系分别处理各个通道重构 RGB 图。

在原图的基础上略有提亮，且相比于分别均衡化的方法色彩失真小。

（3）将 RGB 模型下图像转到 HIS 模型下，对强度分量做直方图均衡化。

亮度明显提升，色彩鲜艳且失真小，均衡化效果强于前两种方法。但感官上均衡化后后图像的色彩和原图有一定差异。

原因：

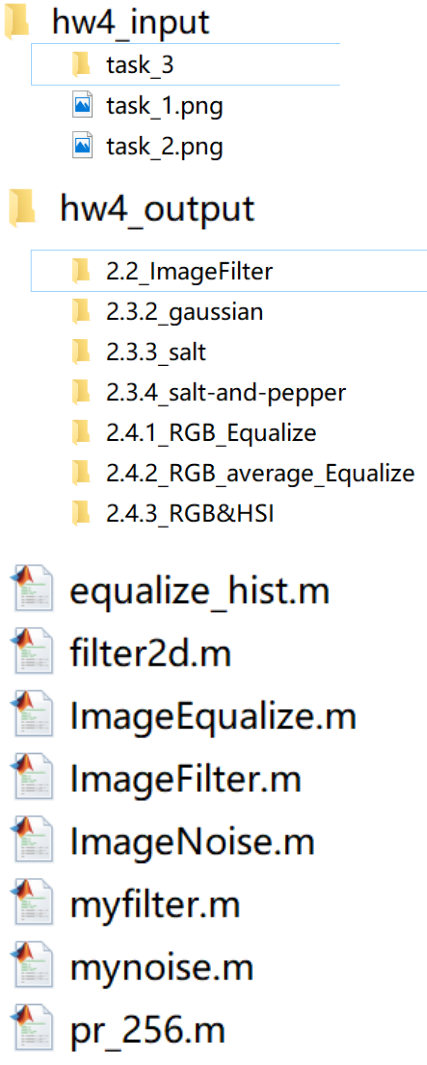
单独对彩色图像的 RGB 分量进行直方图均衡的效果通常不好，将会产生颜色差错。在对彩色图像均衡化时，应该均匀地展开这种彩色灰度，保持彩色本身（即色调）不变。HIS 彩色空间时适合这类方法的理想空间。

在 HIS 模型下对 I 分量进行直方图均衡化，没有改变图像的色调和饱和度的值，但它的确影响了图像的整体感观，即强度分量的改变同样会影响图像中颜色的表现。

应用于强度分量的直方图均衡化通过均衡图像的亮度分量使整个图像有效提亮，对色调和饱和度都不产生影响，对彩色图像是更好的均值化方法。

5. 附录

5.1 文件结构描述



matlab文件名	形式	功能
equalize_hist.m	function	直方图均衡化
filter2d.m	function	滤波
ImageEqualize.m	[script]	2.4彩色图像的直方图均衡化
ImageFilter.m	[script]	2.2图像滤波
ImageNoise.m	[script]	2.3图像去噪
myfilter.m	function	2.2,2,3所用各种滤波器
mynoise.m	function	2.3噪声生成器
pr_256.m	function	概率密度计算

5.2 运行指南

运行 ImageFilter.m，ImageNoise.m 和 ImageEqualize.m 三个代码文件，可以得到所有题目的输出结果。

注：ImageNoise 由于多次调用滤波函数，出结果速度较慢，需要等待十几秒。