

HW4

1 Exercise

1.1 Color Spaces

1. Give one example where the HSI color space is advantageous to the RGB color space. Also give one example where RGB is advantageous to HSI.

Ans:

HIS 优于 RGB 的例子：只改变一张图片的亮度，HIS 空间的处理很简单，只需要单独处理 I 分量即可，而 RGB 空间则要同时改变 R, G, B 三个分量。

RGB 优于 HIS 的例子：计算彩色图像的补色，RGB 空间只需要采用公式 $\begin{bmatrix} Rc \\ Gc \\ Bc \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R0 \\ G0 \\ B0 \end{bmatrix}$ 即可，而 HIS 空间则不能直接处理。

2. What is the effect of adding 60° to the Hue components on the R, G, and B components of a given image?

Ans:

这样处理后的 R 分量变为 $\frac{R+G}{2}$ ，G 分量变为 $\frac{G+B}{2}$ ，B 分量变为 $\frac{R+B}{2}$ 。

1.2 Color Composition

Consider any three valid colors c_1 , c_2 and c_3 with coordinates (x_1, y_1) , (x_2, y_2) and (x_3, y_3) , in the chromaticity diagram in Fig. 1 (or Fig. 6.5 of the textbook). Derive the general expression(s) for computing the relative percentages of c_1 , c_2 and c_3 composing a given color that is known to lie within the triangle whose vertices are at the coordinates of c_1 , c_2 and c_3 .

Ans: 假设该给定颜色的坐标为 (x, y) 。

该点到 c_1 、 c_2 、 c_3 的距离分别为：

$$d_1 = ((x-x_1)^2 + (y-y_1)^2)^{1/2};$$

$$d_2 = ((x-x_2)^2 + (y-y_2)^2)^{1/2};$$

$$d_3 = ((x-x_3)^2 + (y-y_3)^2)^{1/2}。$$

距离之和 $D = d_1 + d_2 + d_3$ 。

则三种颜色的比例分别为 $\frac{d_1}{D}$, $\frac{d_2}{D}$, $\frac{d_3}{D}$ 。

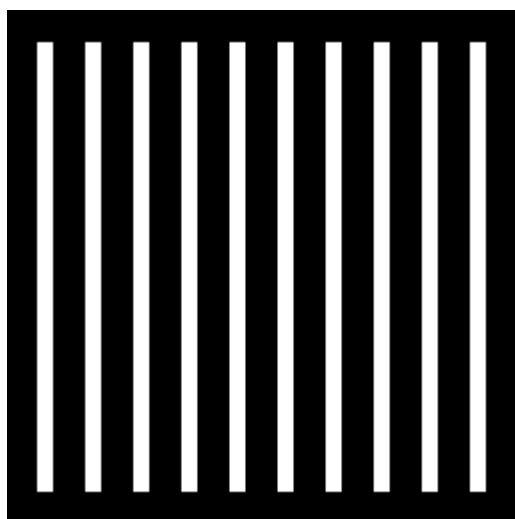
2 Programming Tasks

2.1 Pre-requirement

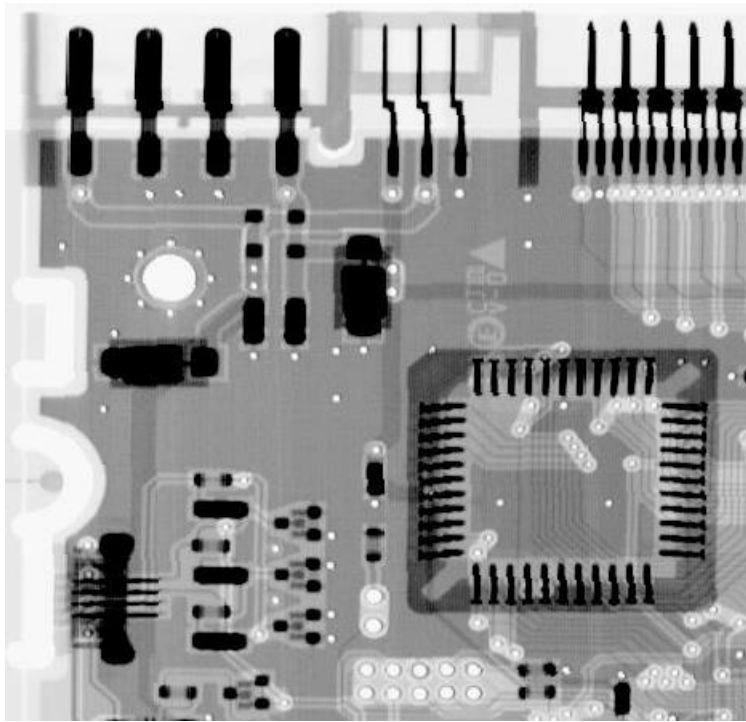
Third-party packages: openVC

Language: C++

Source Image:



task_1.png



task_2.png



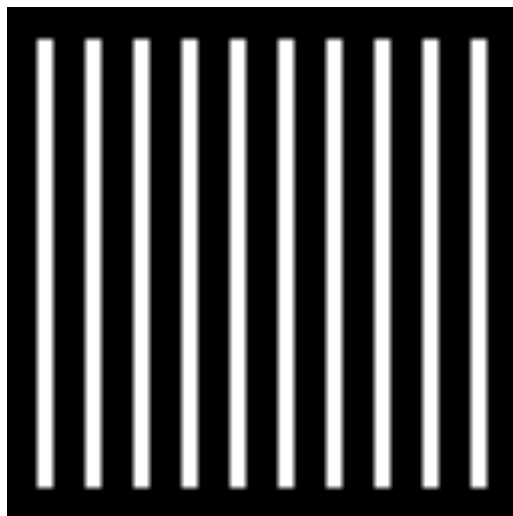
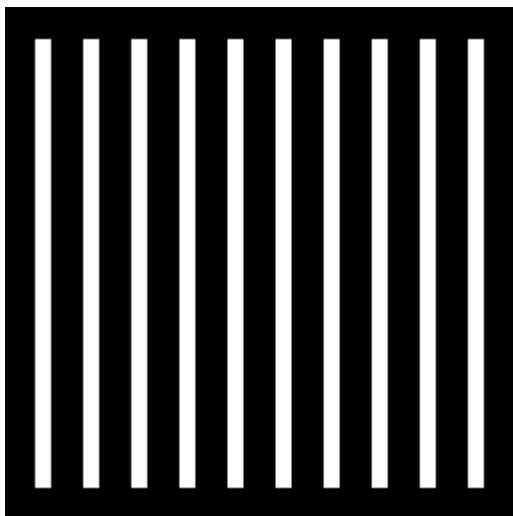
81.png

2.2 Image Filtering

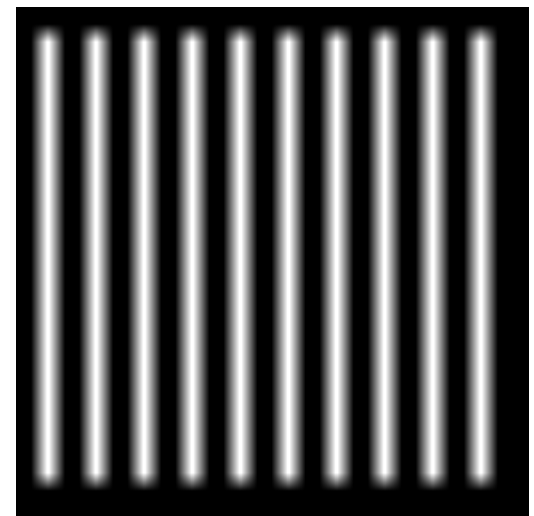
1. Using 3×3 and 9×9 arithmetic mean filters:

Source Image:

3×3 :



9×9 :

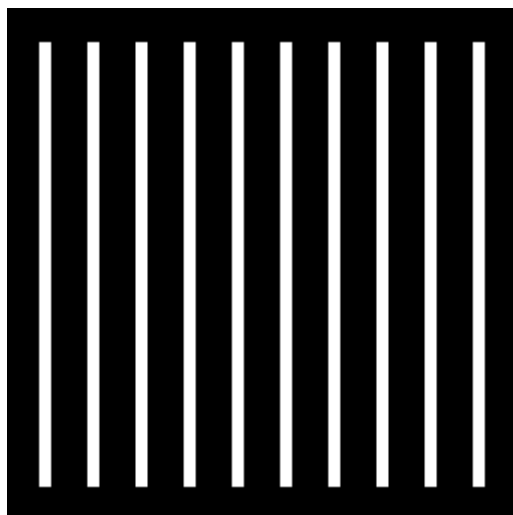
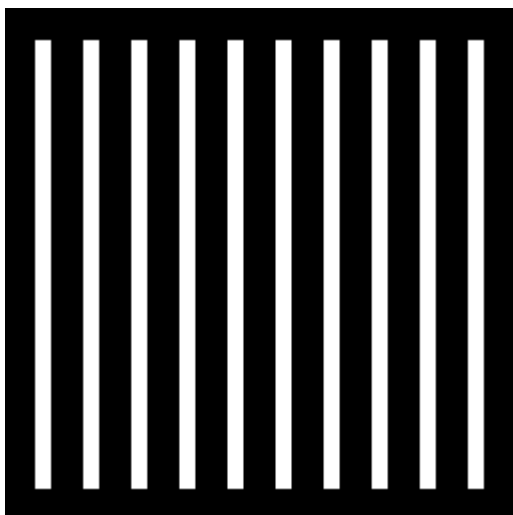


经过算术平均后的图像变得模糊，条纹变细变短，颜色变淡。

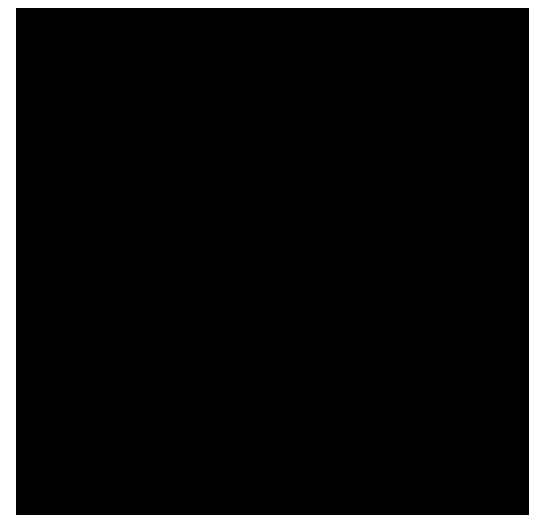
2. Using 3×3 and 9×9 harmonic mean filters:

Source Image:

3×3 :



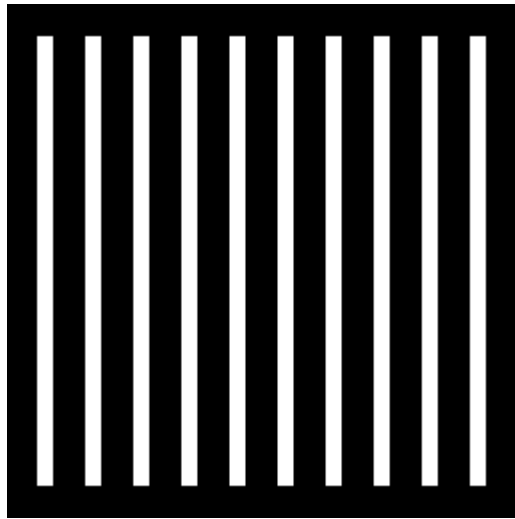
9×9 :



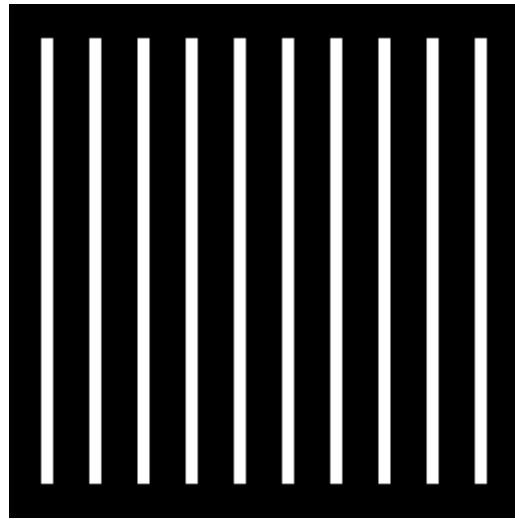
3×3 的调和均值处理后条纹变细变短，但颜色不变。而 9×9 的调和均值处理后的图像是全黑的。

3. Using 3×3 and 9×9 contraharmonic mean filters:

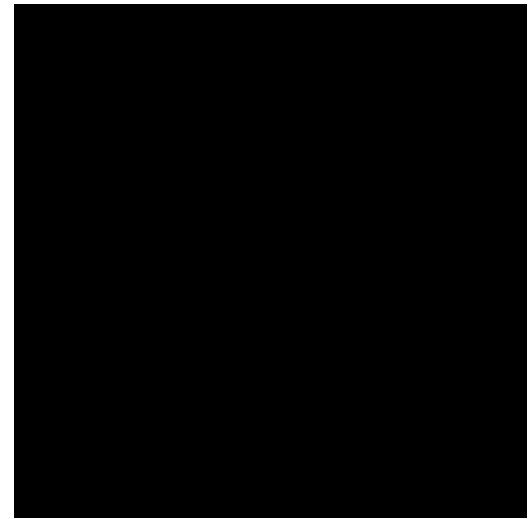
Source Image:



3×3 :



9×9 :



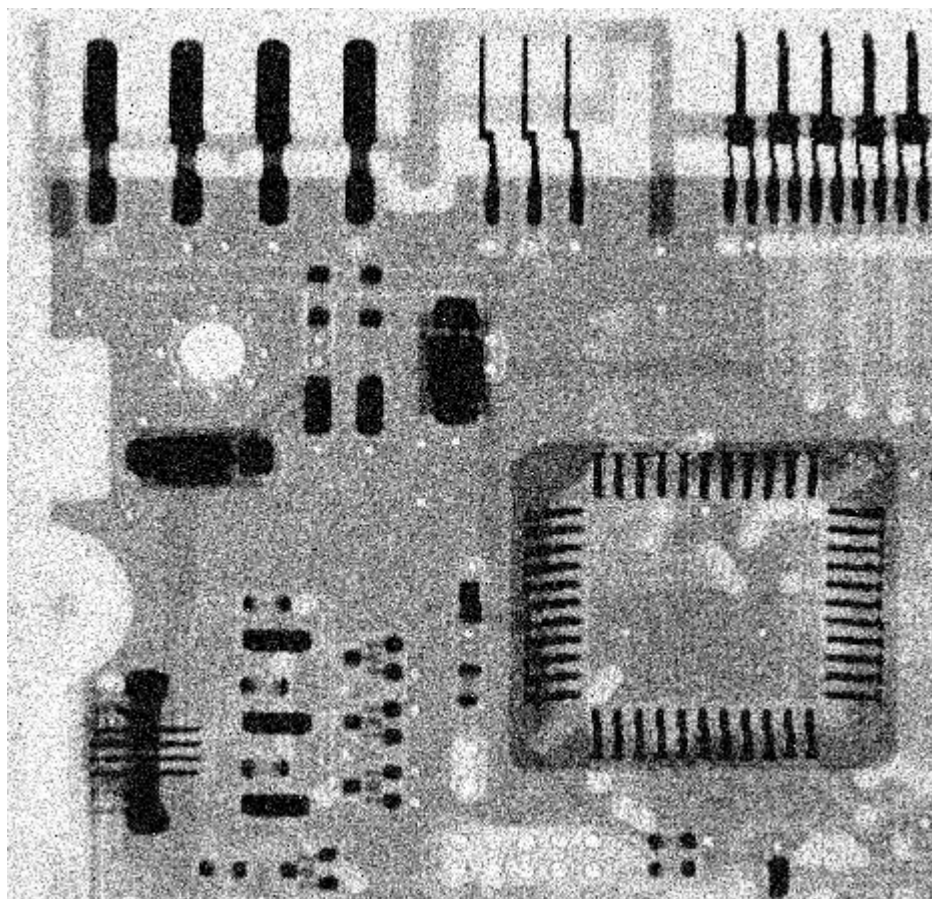
3×3 的反调和均值处理后条纹变细变短，但颜色不变。而 9×9 的反调和均值处理后的图像是全黑的。结果与调和均值很相似。

2.3 Image Denoising

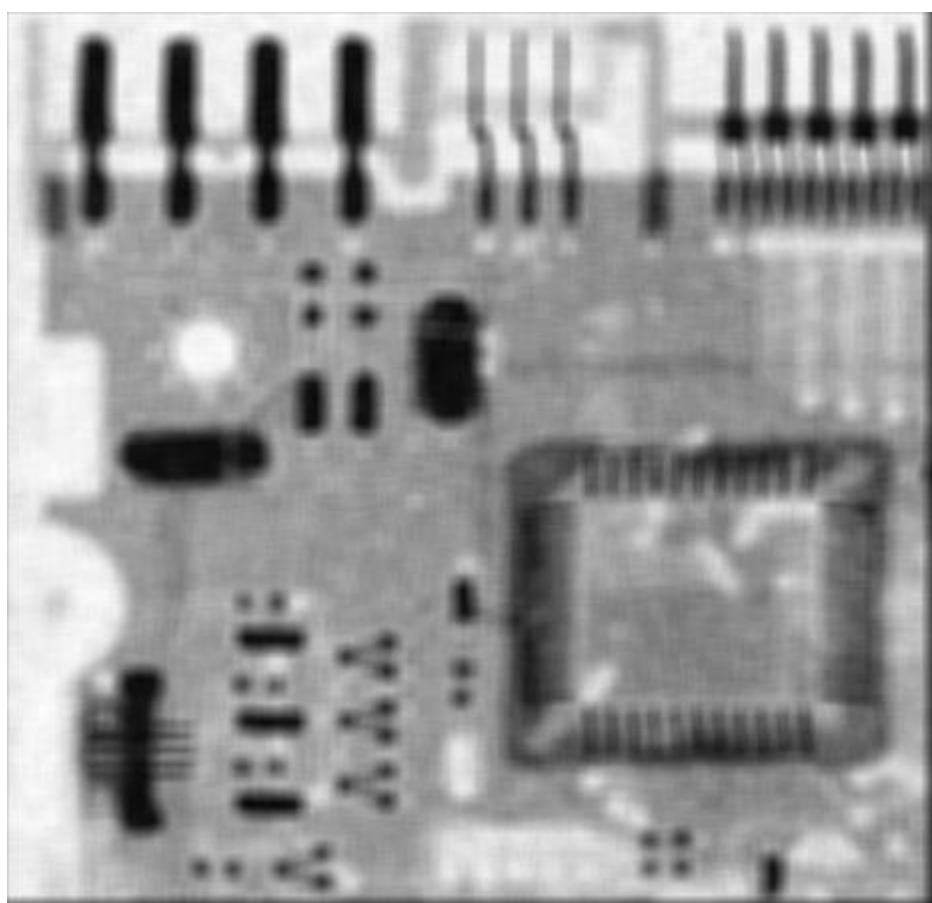
1. Noise Generator:

```
/**函数功能：在输入图片上产生高斯噪声
  输入：input_img 输入图片的灰度值信息
        h  图片的高度
        w  图片的宽度
        mean 高斯噪声的均值
        stdvar 高斯噪声的标准差
  输出：产生了高斯噪声的图片
*/
double* noiseGenerator1(double* input_img, int h, int w, double mean, double stdvar){ {... }}
/**函数功能：在输入图片上产生盐噪声
  输入：input_img 输入图片的灰度值信息
        h  图片的高度
        w  图片的宽度
        pro 产生盐噪声的概率
  输出：产生了盐噪声的图片
*/
double* noiseGenerator2(double* input_img, int h, int w, double pro){ {... }}
/**函数功能：在输入图片上产生椒盐噪声
  输入：input_img 输入图片的灰度值信息
        h  图片的高度
        w  图片的宽度
        pro 产生椒噪声和盐噪声的概率
  输出：产生了椒盐噪声的图片
*/
double* noiseGenerator3(double* input_img, int h, int w, double pro){ {... }}
```

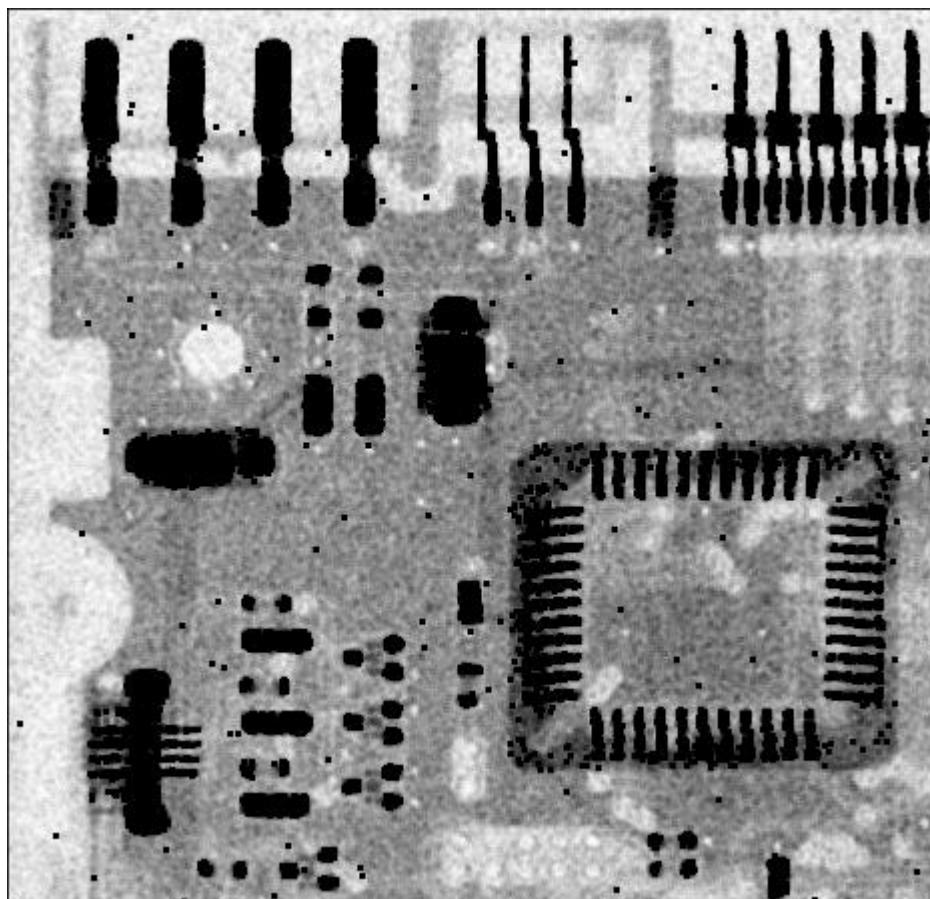
2. Gaussian noise(0, 40)



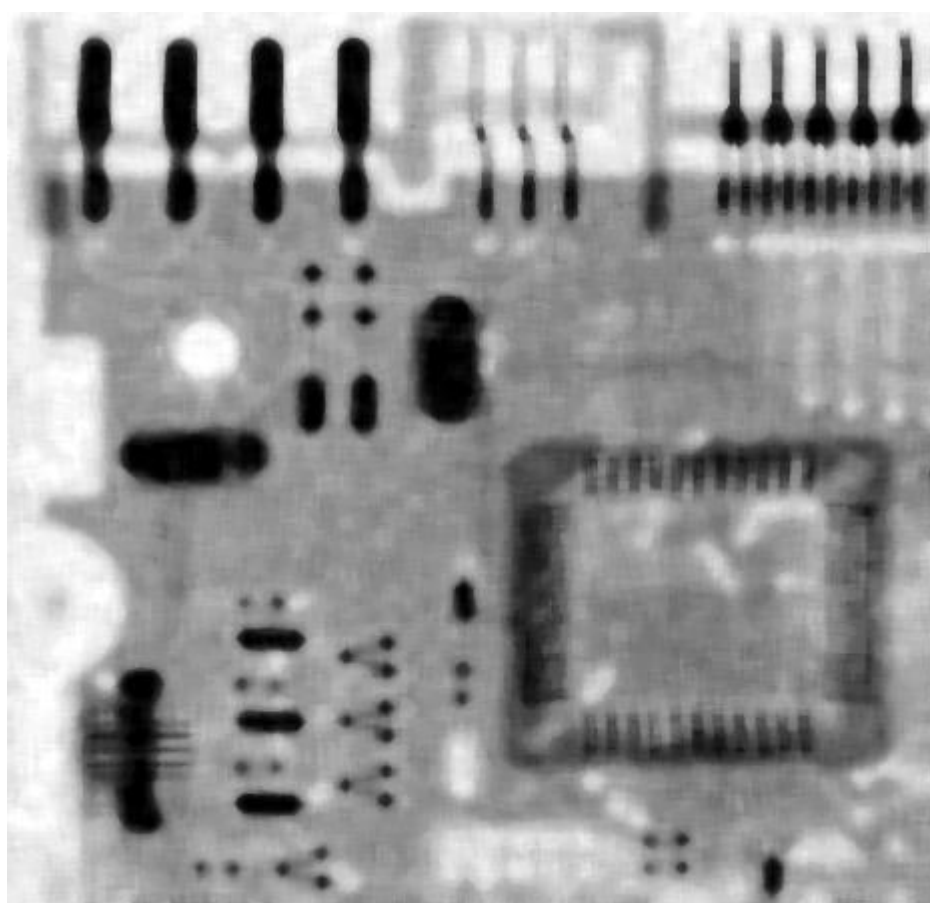
Arithmetic mean filtering



Geometric mean filtering

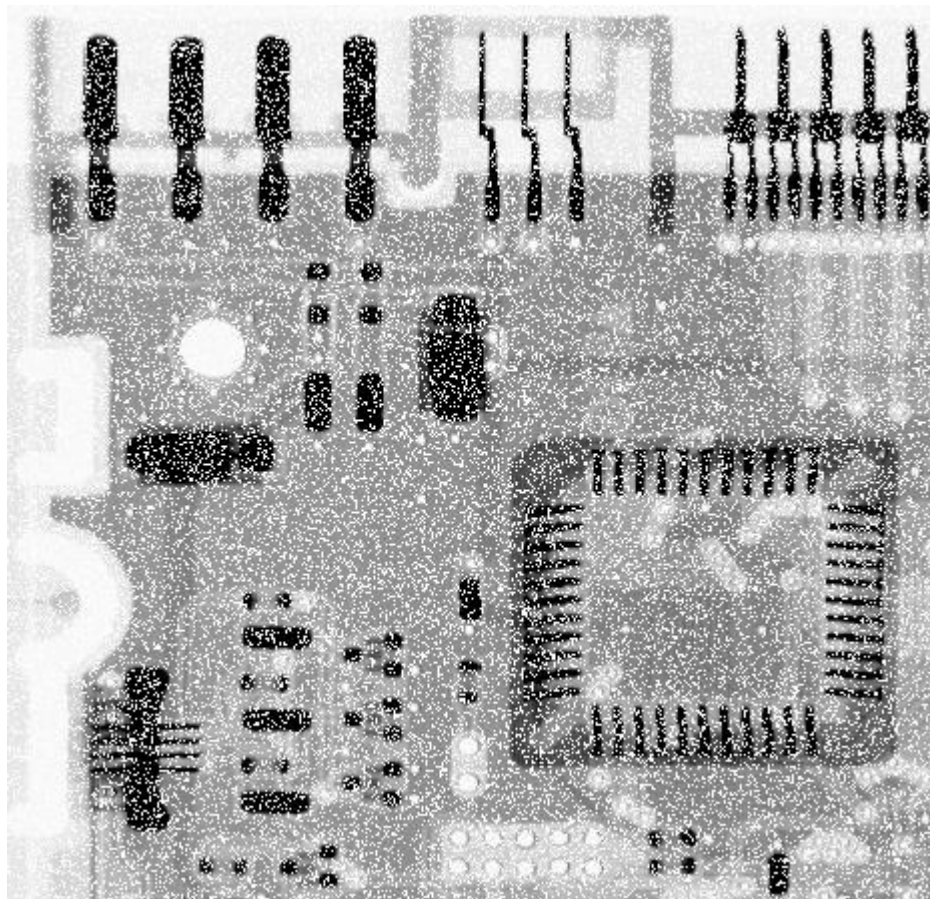


Median filtering

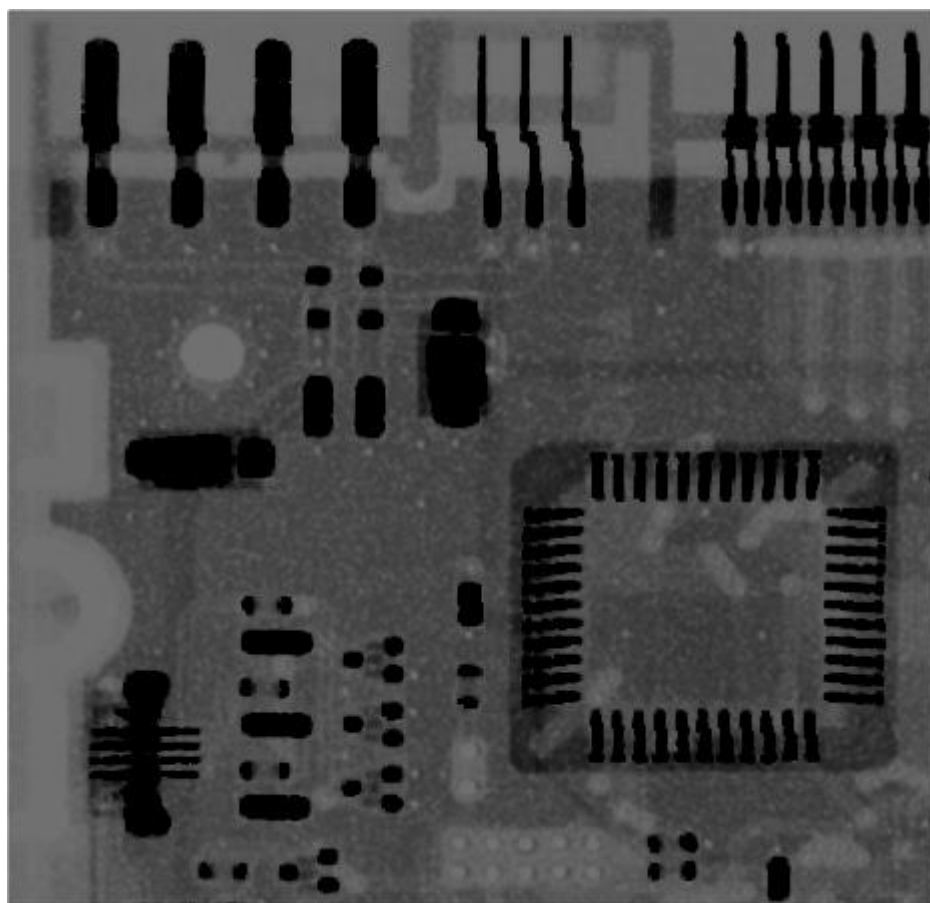


三个滤波的结果中最好的应该算是几何均值，在去除高斯噪声的同时带来的模糊感没有其他两个滤波器严重，由于需要将像素点的灰度值连乘，黑色点附近的点会受到影响，也会变成 0，因此黑色像素块的边界有所扩展，还有一些黑色的噪声点。其他两个滤波器能够平滑高斯噪声，但同时使图像变得模糊。

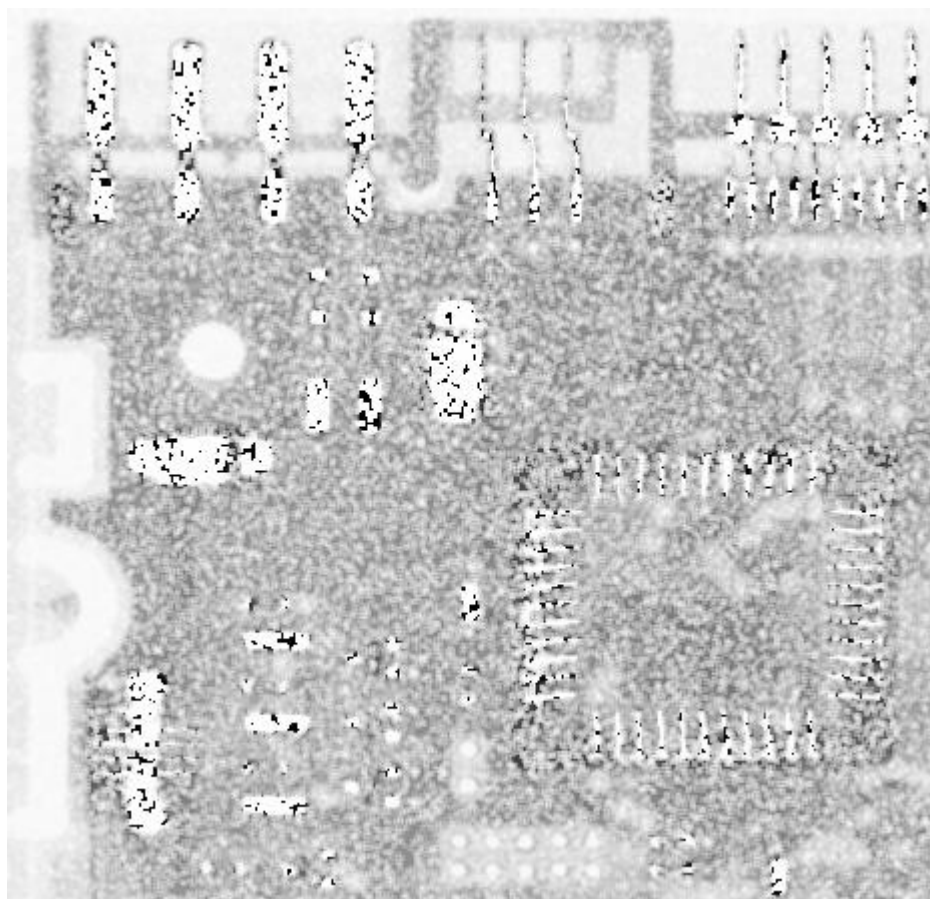
3. Salt noise(0,2)



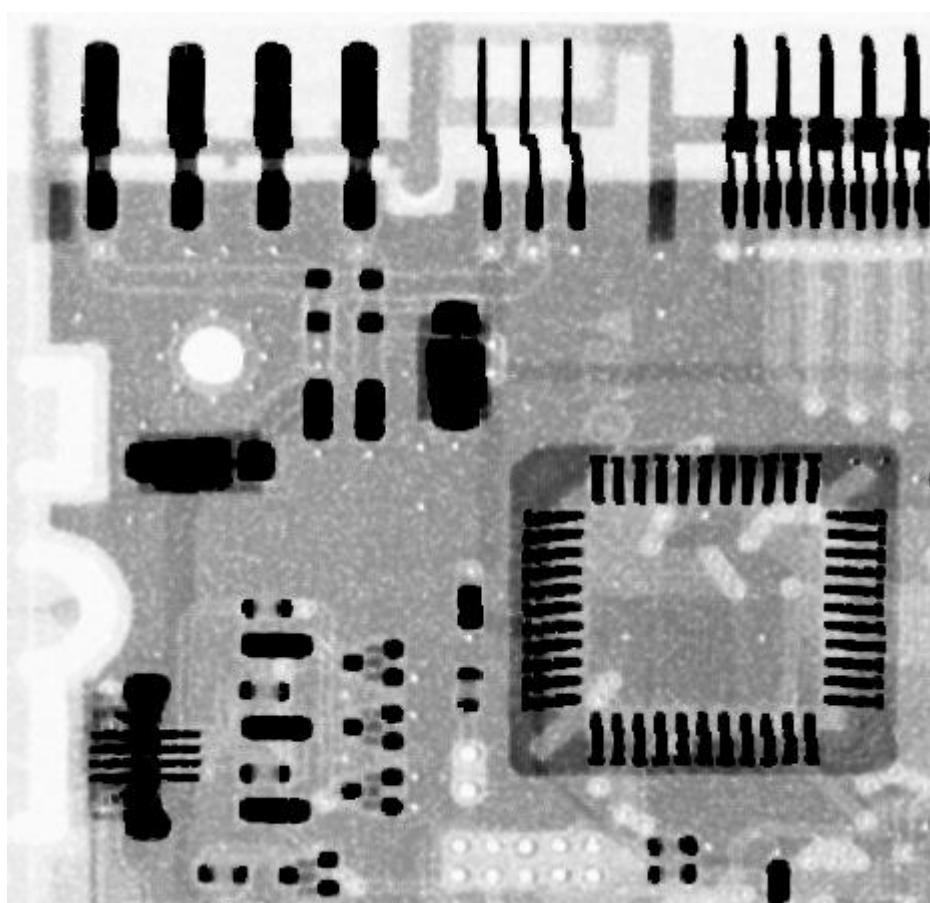
Harmonic mean filtering



Contraharmonic mean filtering($Q = 1.5$)



Contra-harmonic mean filtering(Q=-1.5)

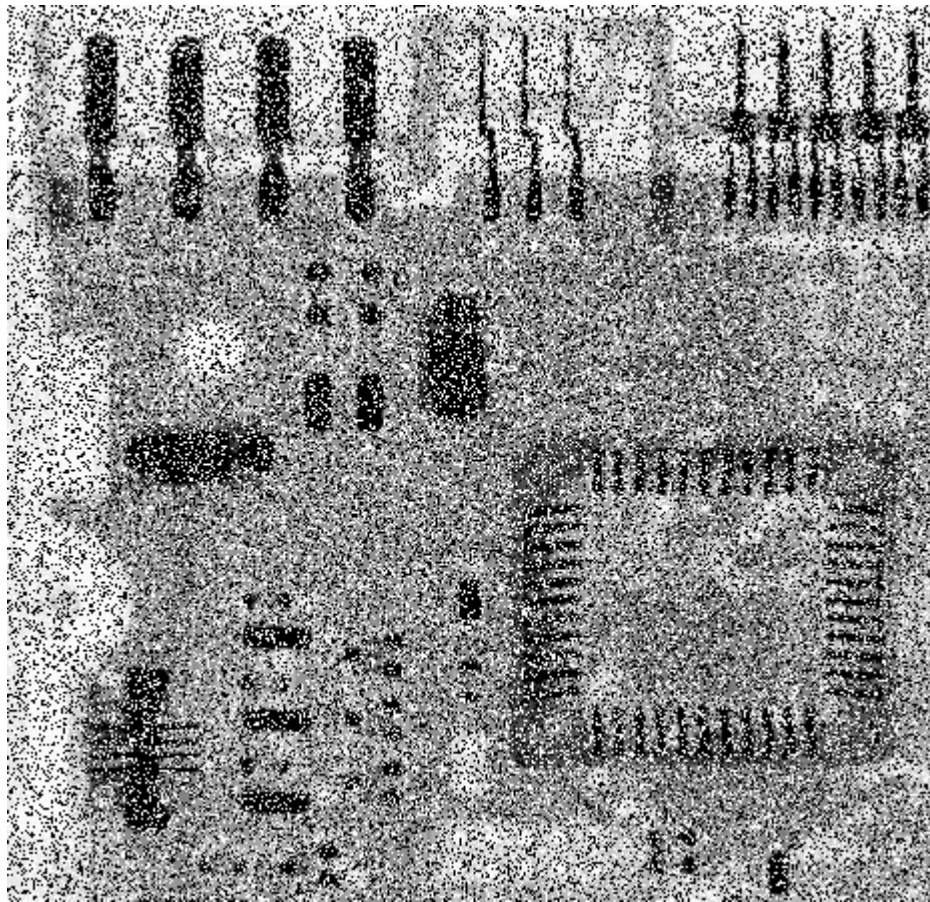


调和均值能够去掉绝大部分的盐噪声，但副作用是图片整体偏暗。Q 为 1.5 的反调和均值图像严重失真，大部分的像素点都显示白色，还有一些黑色点。Q 为-1.5 的反调和均值图像去掉了绝大部分的盐噪声，但黑色像素块的边界有所扩展。带负值的 Q 的反调和均值在处理盐噪声时有很好的效果，而正值的 Q 则会带来灾难性的结果。这是因为盐噪声是亮噪声，灰度值较大。由

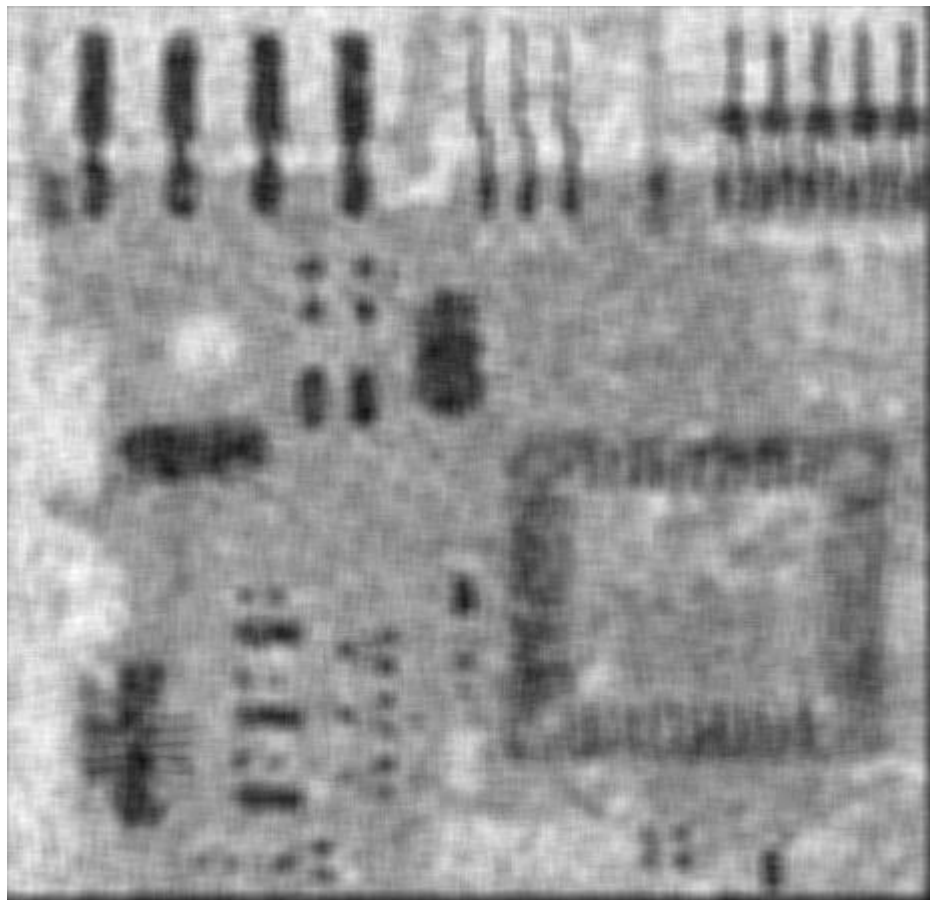
$$\hat{f}(x, y) = \frac{\sum_{(s, t) \in S_{xy}} g(s, t)^{Q+1}}{\sum_{(s, t) \in S_{xy}} g(s, t)^Q}$$

可知，负值的 Q 会降低滤波器窗口中灰度值大的像素点的影响，从而达到消除盐噪声的目的。而当 Q 为正值时，窗口中大的灰度值对即将被赋值的像素点的影响不降反升，因此会使得盐噪声更为严重，造成灾难性的效果。

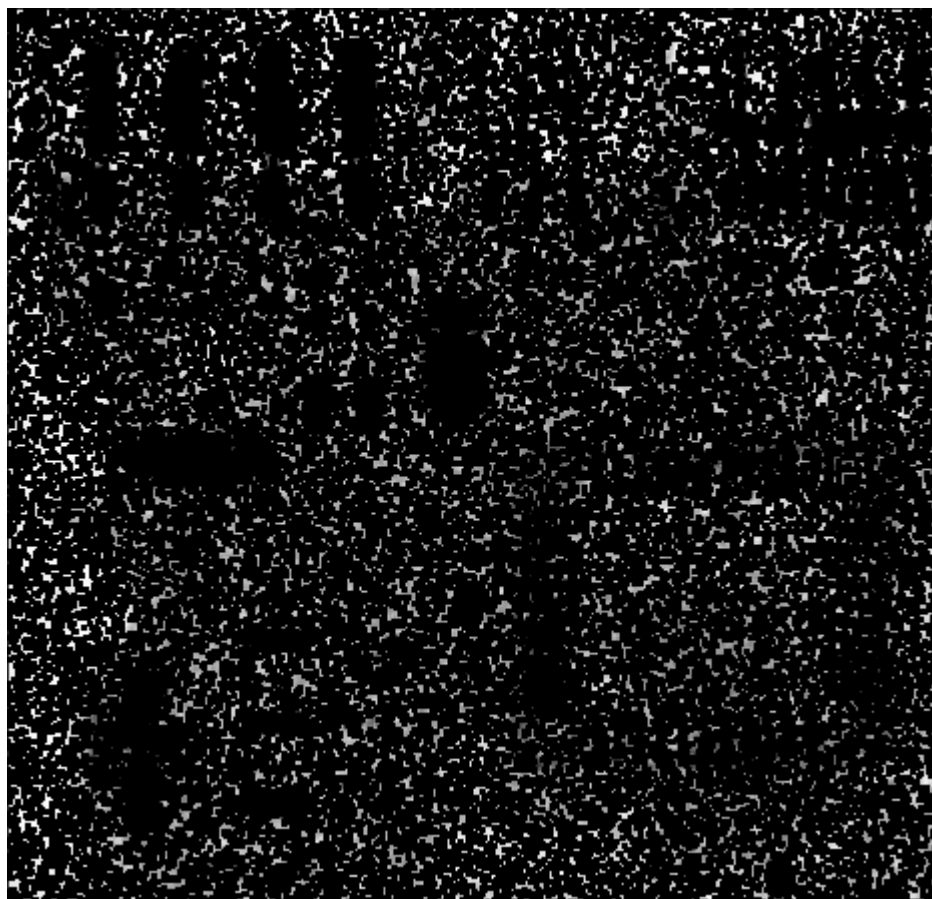
4. Salt-and-pepper noise(0.2,0.2)



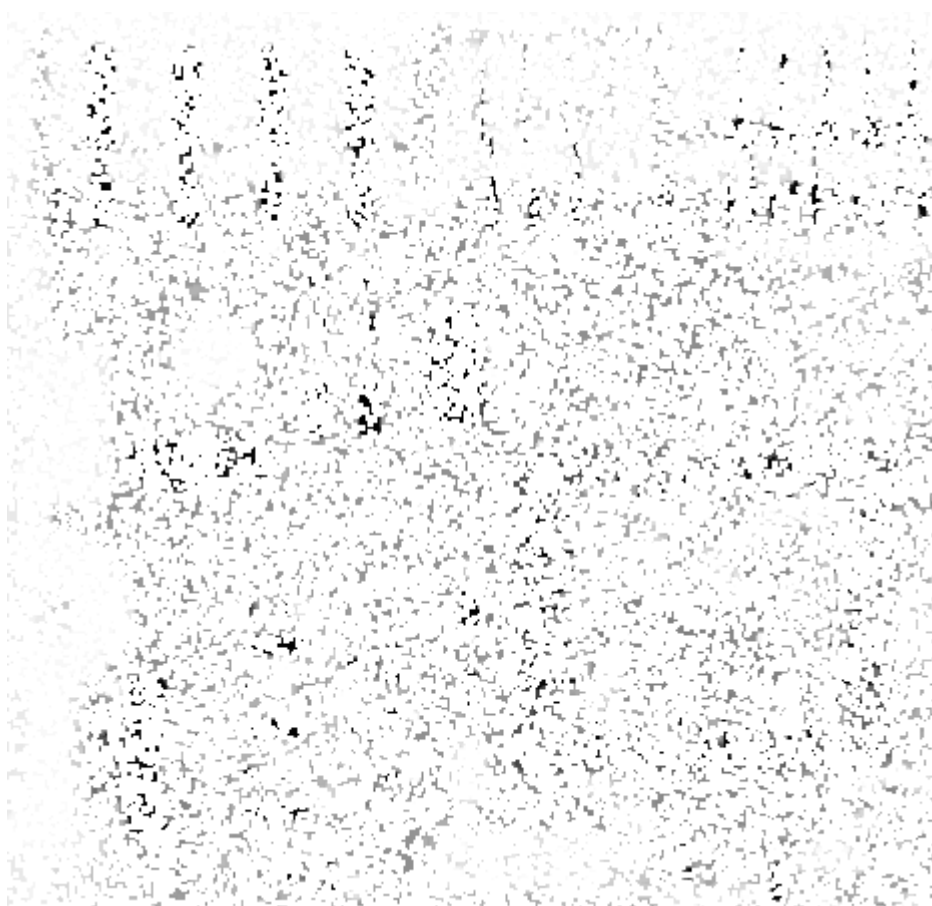
Arithmetic mean filtering



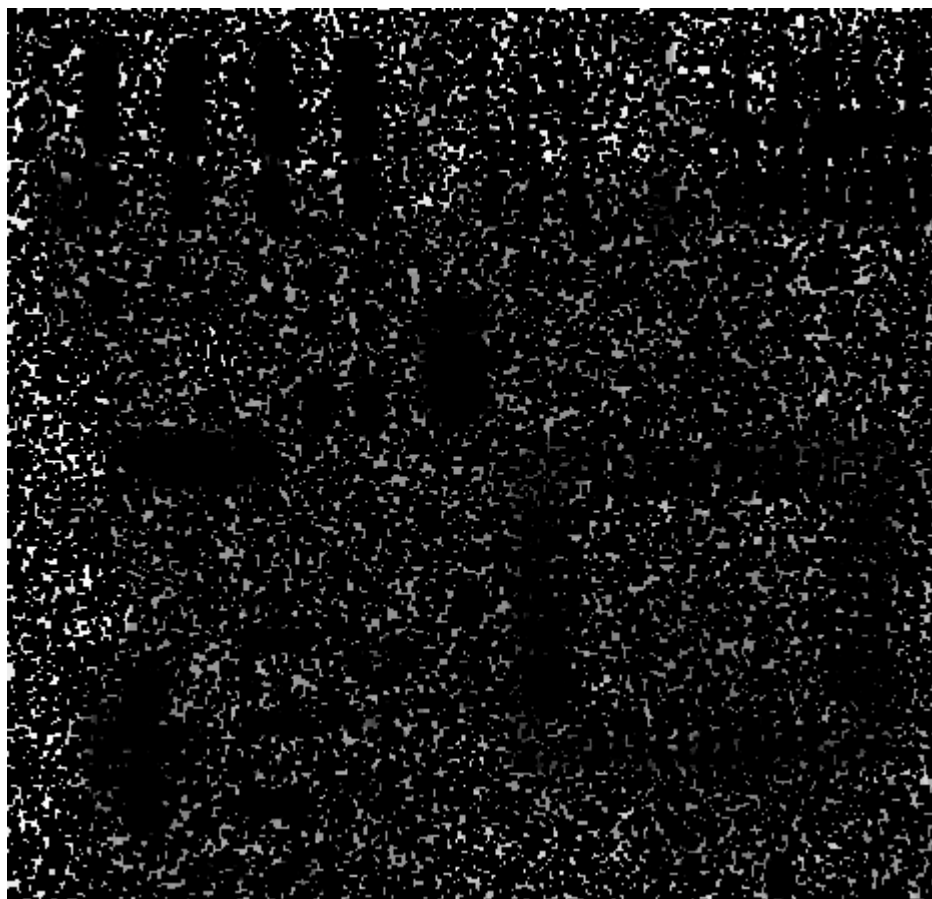
Geometric mean filtering



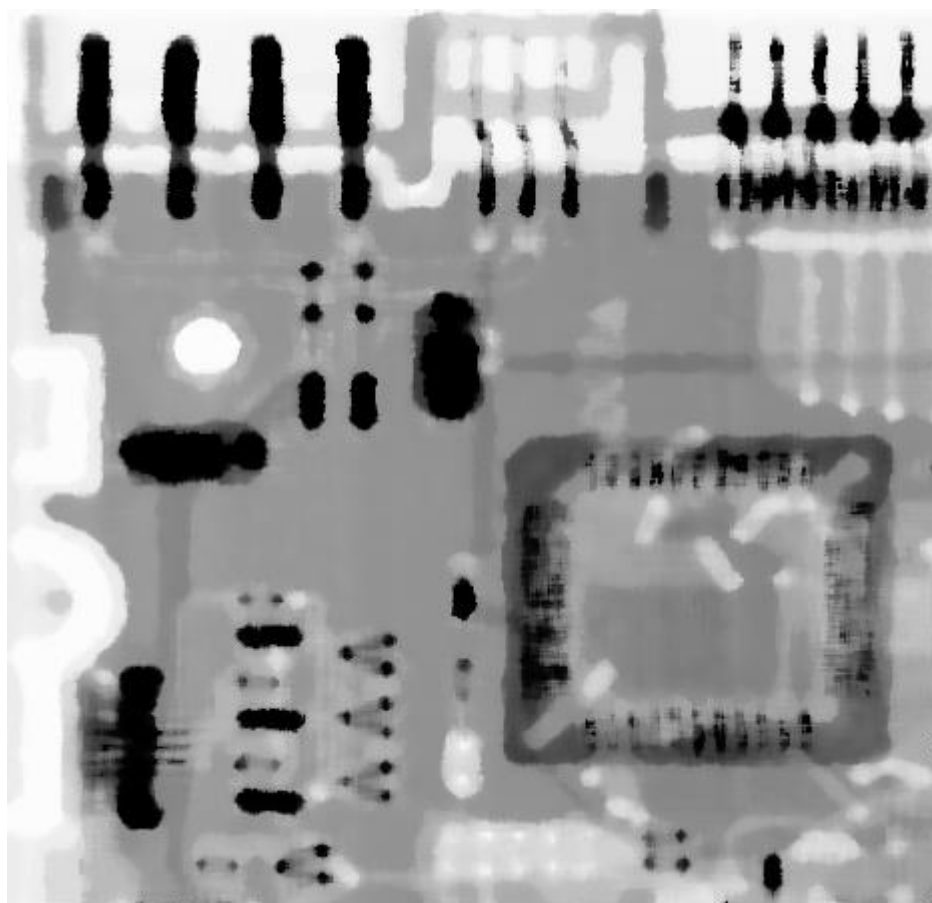
Max filtering



Min filtering



Median filtering



经过算数均值处理后的图像去除噪声的同时使得图像变得很模糊，而几何均值、最大值、最小值都会使得图像严重失真，要么大部分变为黑色，要么大部分变为白色。这是因为最大值会取得滤波器窗口最大的值作为像素点的值，如果窗口内有白色，就会变为白色。而最小值的情况与此相似，如果窗口内有黑色，会使得像素点变为黑色。几何均值则是因为窗口中有黑色，连乘后导致像素点变为黑色。相比之下，中值的结果最接近原图，但带来的涂抹感比较重。

5. Discussion of implementation

算数均值：将滤波器窗口范围内的像素点的灰度值加总，然后除以窗口大小，再将该值赋给对应的像素点。

几何均值：将滤波器窗口范围内的像素点的灰度值相乘，然后以窗口大小开方，再将该值赋给对应的像素点。

调和均值：先将图片的全部像素值取倒数，然后对每一个像素点做算术均值，再将结果取倒数，赋给对应的像素点。

反调和均值：先将图片的全部像素值取 $Q+1$ 次方，记为 $M1$ 。再将原图片的全部像素值取 Q 次方，记为 $M2$ 。分别对 $M1$ 和 $M2$ 进行算数均值，得到的结果记为 $M1'$ 和 $M2'$ 。将 $M1'$ 和 $M2'$ 的每个像素点的灰度值相除，赋给对应的像素点。

最大值：取出滤波器窗口范围内的像素点的灰度值的最大值，赋给对应的像素点。

最小值：取出滤波器窗口范围内的像素点的灰度值的最小值，赋给对应的像素点。

中值：取出滤波器窗口范围内的像素点的灰度值的中值，赋给对应的像素点。

最大值、最小值、中值等三个统计滤波器的实现是相似的，我是用 vector 存储窗口范围内的像素点，然后对 vector 进行排序，再得到最大值、最小值和中值。中值需要考虑偶数个像素点和奇数个像素点的情况。

2.4 Histogram Equalization on Color Images

1. Processing R, G and B channels separately

Source Image:



result:



2. Processing R, G and B channels with histogram equalization intensity transformation function

Source Image:



result:



3. Comparison and explanation

第一步的结果偏黄，与原图相差较大。而第二步的结果比原图要亮，细节部分更加清晰，例如云彩、地面的纹理和车厢中的暗处。

第一步的处理方式是分别对三个通道进行直方图均衡化，这种情况下，每个通道的转换函数 T 只与该通道有关。而实际上，彩色图像的颜色是由三个通道的色度值同时决定的，仅仅基于一个通道的信息而改变该通道的色度值，处理后的颜色会与原图有较大的出入。将均衡化后的结果合并，会出现失真的情况，即第一步的结果的发黄情况。

第二步的处理方式是根据三个通道的直方图，得出平均的色度值直方图，然后由此算出转换函数 T 。这种情况下，转换函数同时考虑到三个通道的色度值，将这个转换函数分别作用于三个通道，合并出来的结果在保持原有的色彩特点的同时，达到了均衡的效果。