HW4: 图像复原和彩色图像处理

1 习题

1.1 彩色空间

1. a) B通道图，因为黄 = 红 + 绿（r + g）, 所以原图中黄色部分（脸）在B通道图的颜色会更暗。
   b) G通道图，b的海绵宝宝右上角相比于其他三图少了一个斑点，该斑点在原图中是浅绿色，所以该斑点在G通道图中应最亮。
   d) R通道图，原图中的领带为红色，则领带在R通道图中领带应为白色（255最大值）
   c) 排除法知道c图是灰度图

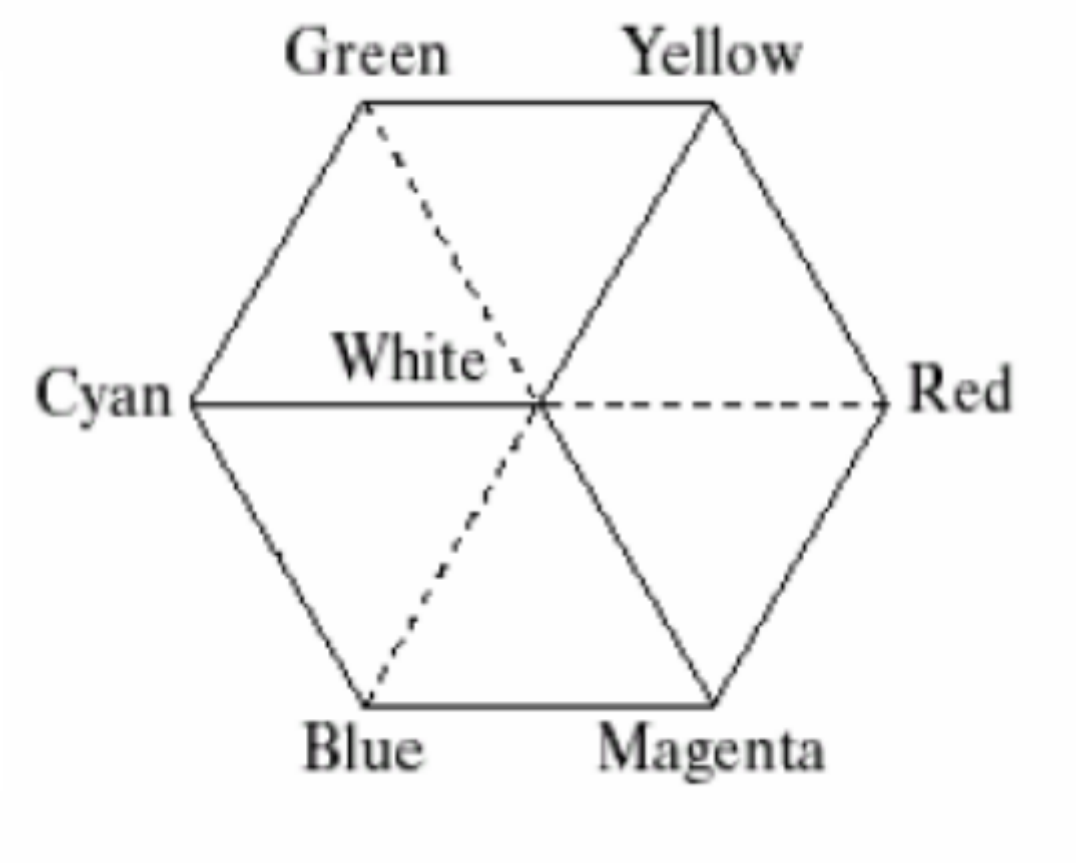
b与c相似的原因为:
由公式：

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

对于转换得到的灰度图而言,G的权值较大,因此b和c图较相似。

2. 选b, 首先，原图中的白色和黑色在转换后的图片不变，因为白色和黑色的色调不会改变，排除ac

另外，由ppt上的图:

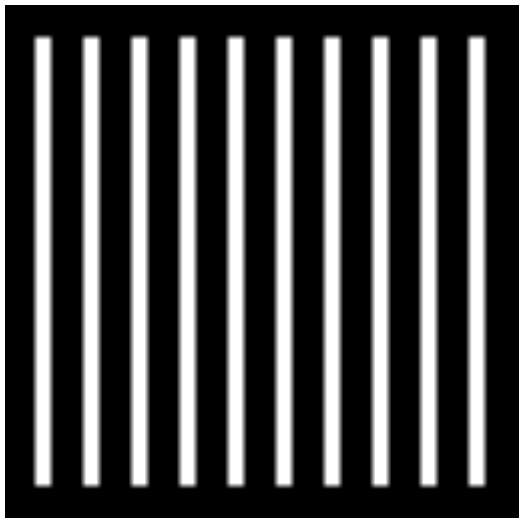


易知黄色将变成绿色，故选b

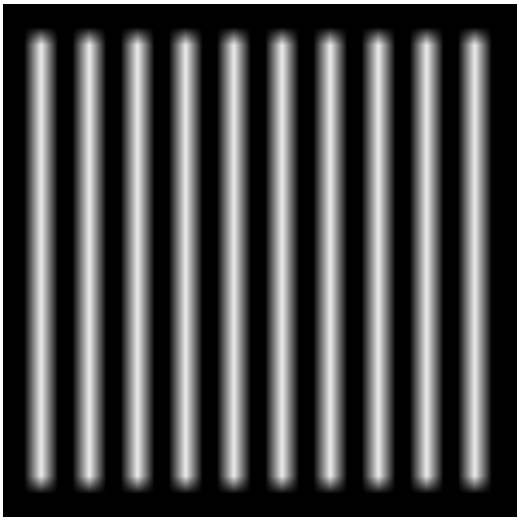
2 编程题

2.2 图像滤波

1. 算术均值滤波器



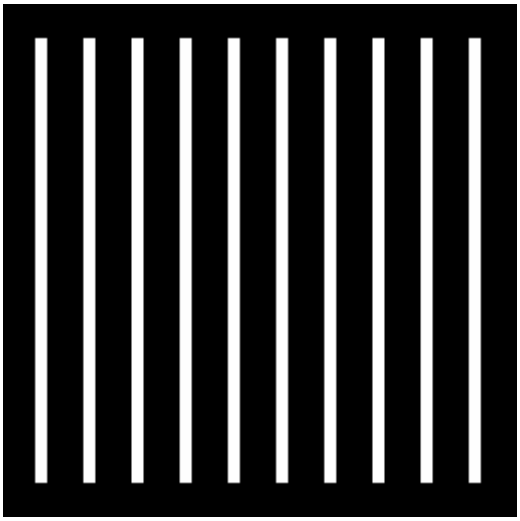
arithmetic_mean_filter3x3



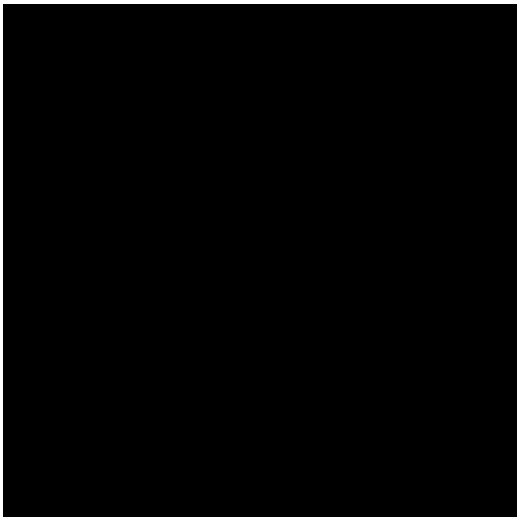
arithmetic_mean_filter9x9

两个算术均值滤波器处理后，白条都变长变宽，黑和白的边界之间的界限没有原图那么分明了，变得模糊。9x9的滤波器对图像上面所说的影响程度比3x3的滤波器更大。

2. 调和均值滤波器



harmonic_mean_filter3x3



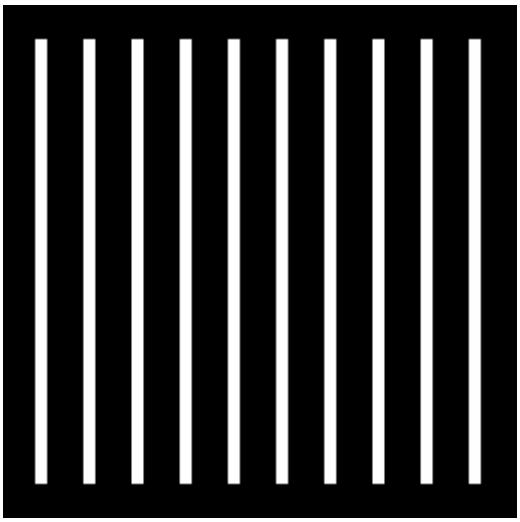
harmonic_mean_filter9x9

在3x3的调和均值滤波器的作用下，白条变窄，长度上应该也有细微的减少（肉眼有点难看出来）。图片更加清晰，黑白边界更加分明。

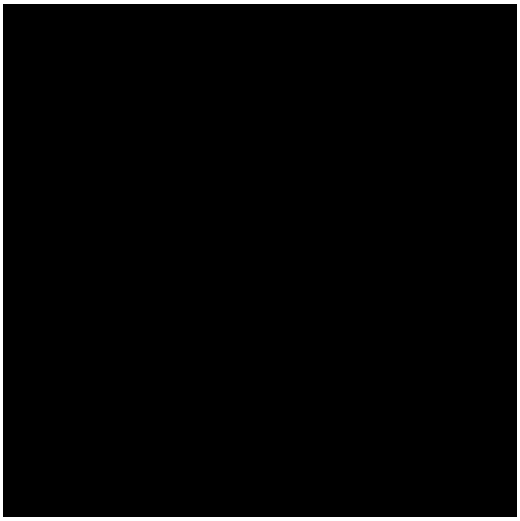
在9x9的调和均值滤波器的作用下，图片变得全黑，应为滤波器太大，白条宽度仅为8，每一次滤波mask里面都有灰度值为0的点，所以出来全黑。

(PS: 约定 x / 无穷大 等于 0， x / 0 等于无穷大)

3. 逆谐波均值滤波器



contra_harmonic_mean_filter3x3



contra_harmonic_mean_filter3x3

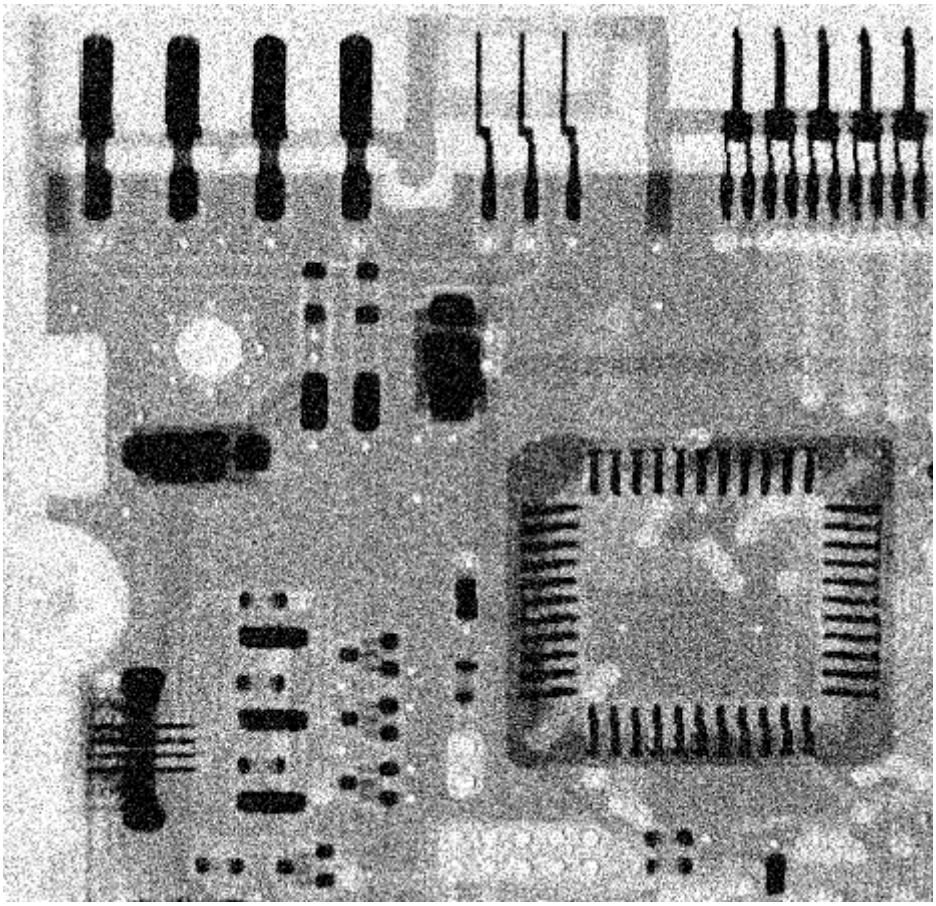
在3x3的逆谐波均值滤波器的作用下，白条变窄，长度上应该也有细微的减少（肉眼有点难看出来）。图片更加清晰，黑白边界更加分明。

在9x9的逆谐波均值滤波器的作用下，图片变得全黑，应为滤波器太大，白条宽度仅为8，每一次滤波mask里面都有灰度值为0的点，所以出来全黑。

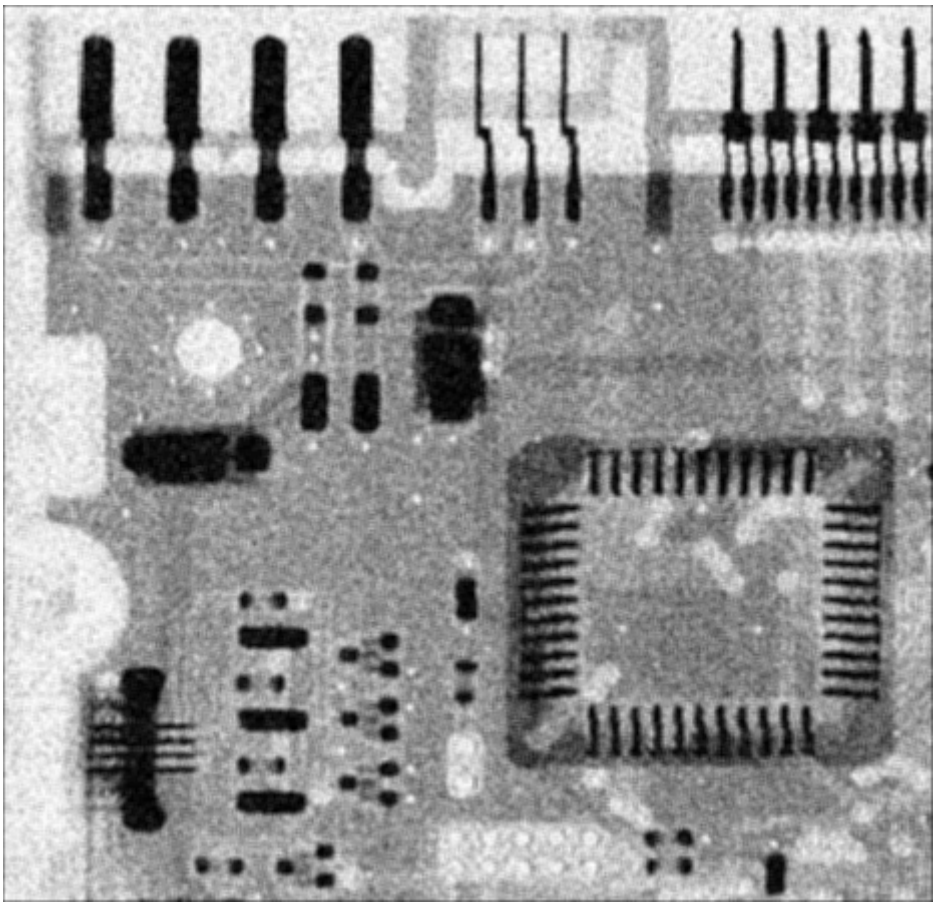
(PS: 约定 x / 无穷大 等于 0， x / 0 等于无穷大)

2.3 图像去噪

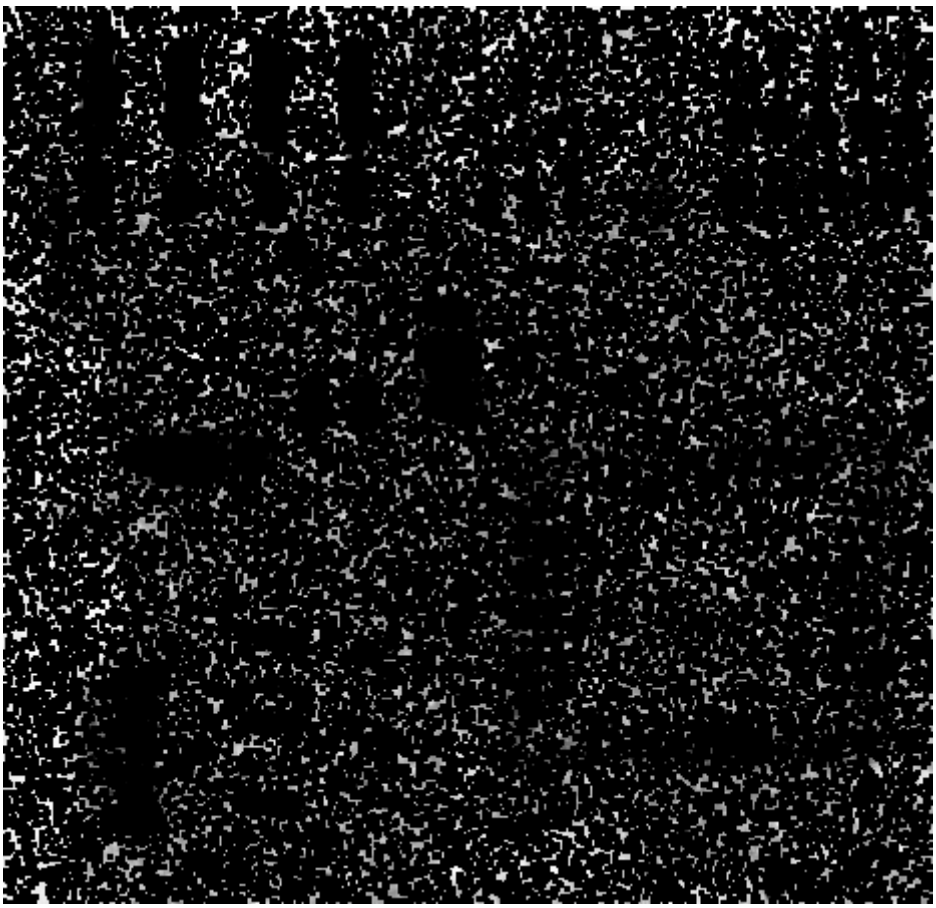
2. 添加高斯噪声



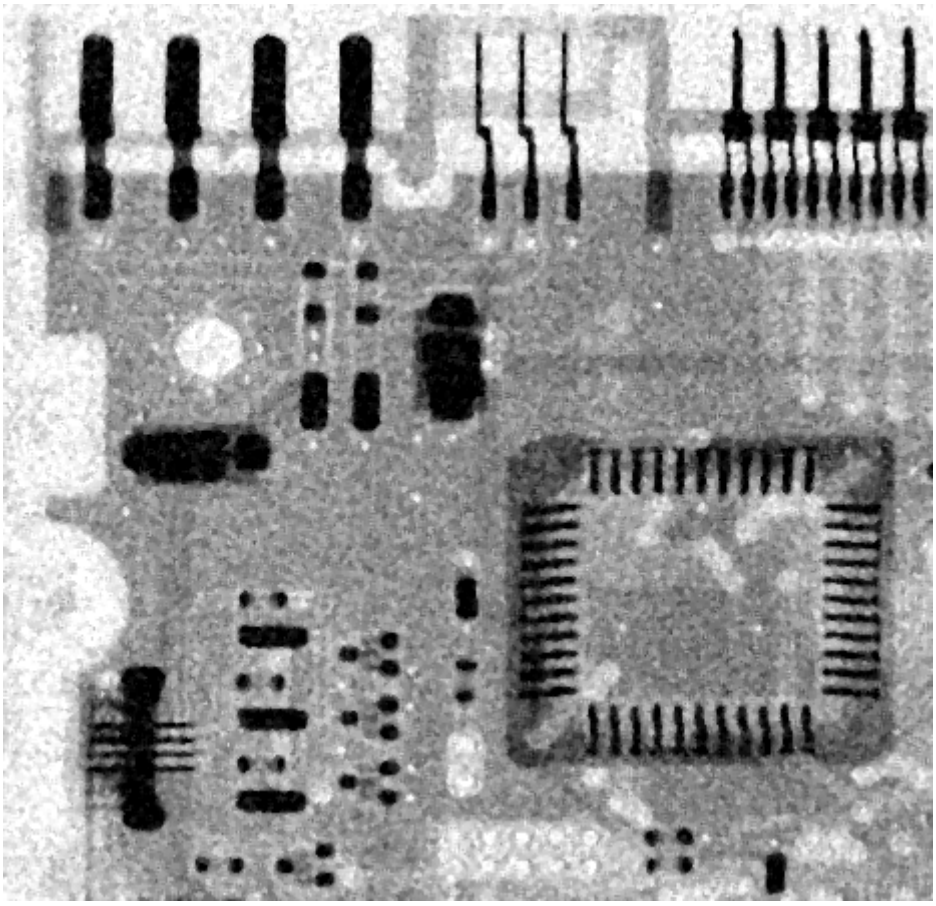
算术均值（3x3）



几何均值(3x3)



中值滤波（3x3）

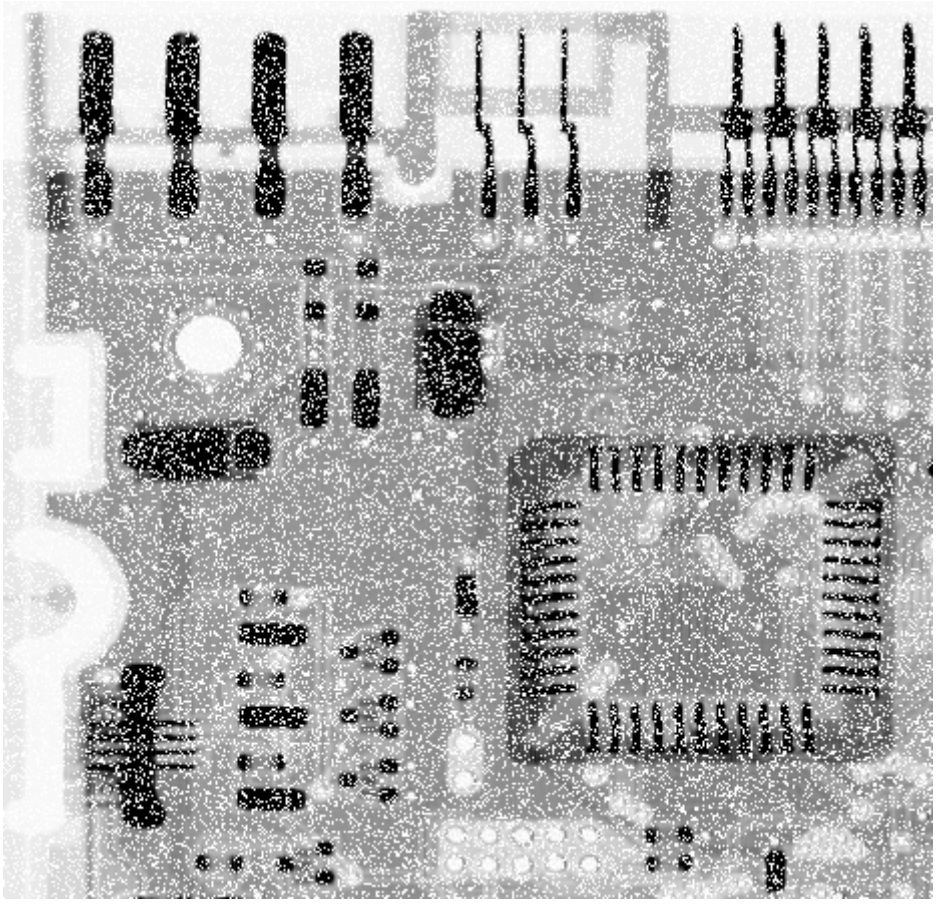


根据观察比较，这三种滤波器都是3x3的滤波器效果更好,主要思考原因是滤波器越大，所包含的噪声点越多，一些细节因为滤波器包含了太多的无关点反而被模糊了。

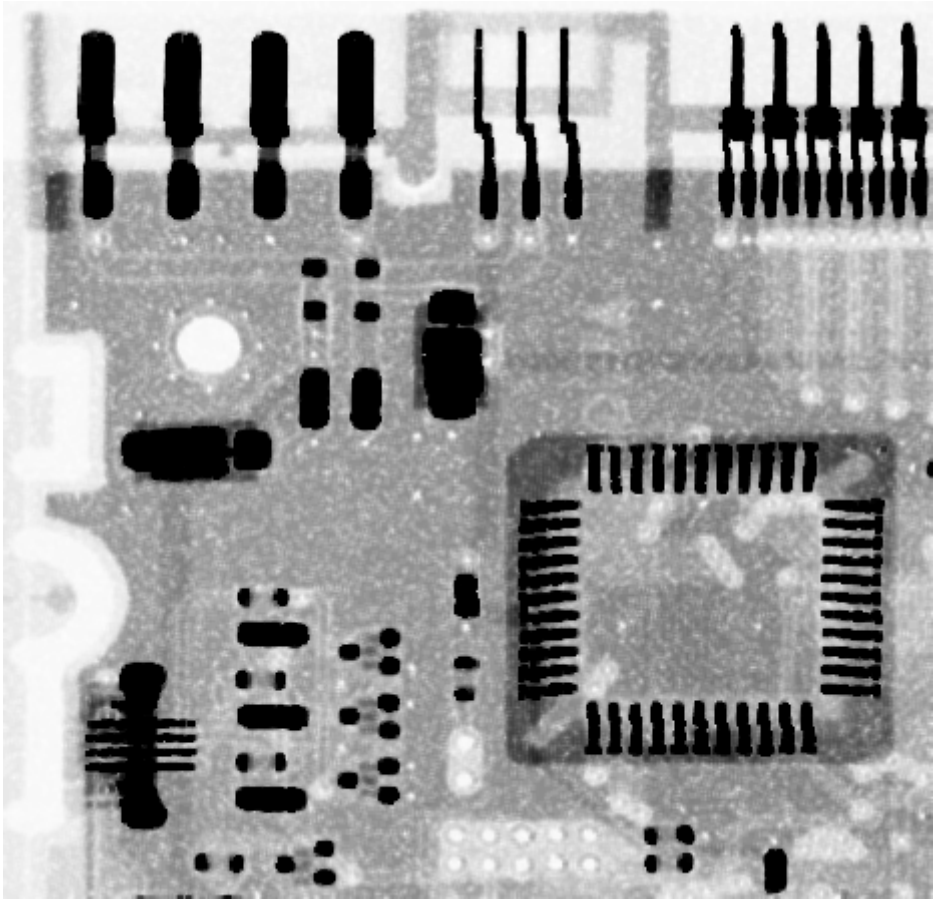
总体上来看中值滤波和算术平均差别不大，主要原因是滤波器较小，区别在一些黑白分明的地方体现出来，因为中值滤波会忽略极值影响，而算术滤波会被极值拉高或者拉低，模糊性更强。

几何平均的细节更加分明，但是从公式上解释，会加上一些新的黑色噪点，实际结果与猜想吻合。

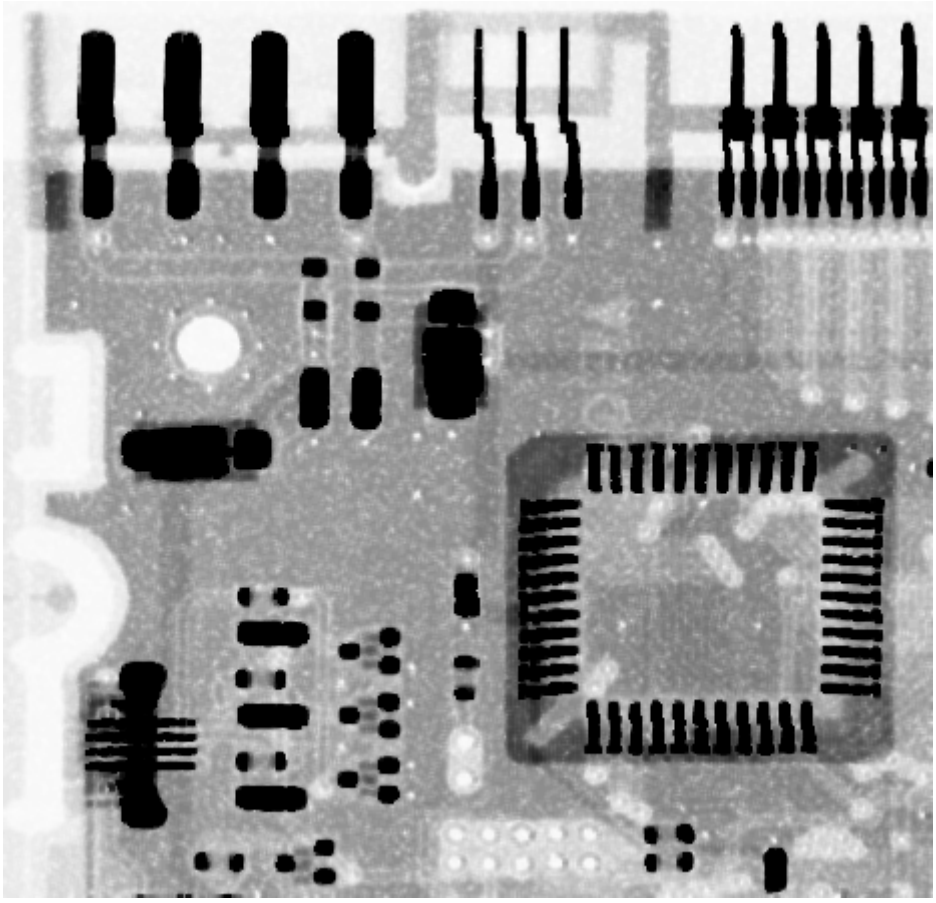
3. 添加盐噪声



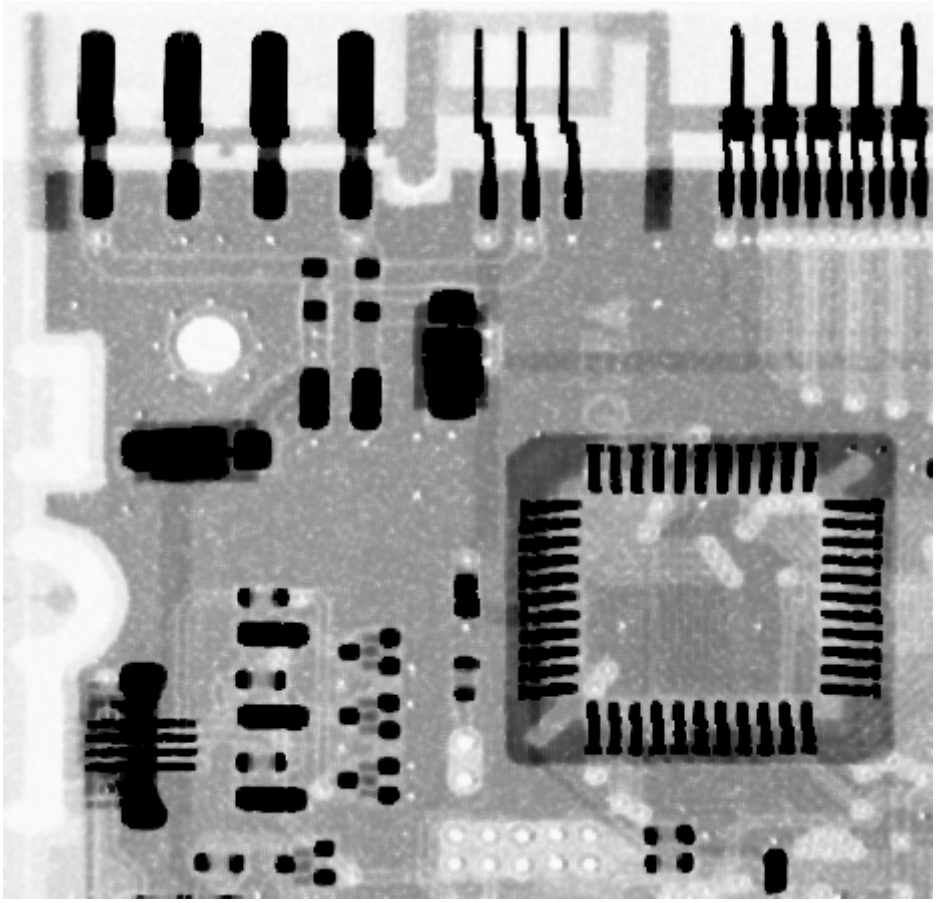
调和均值滤波器(3x3)



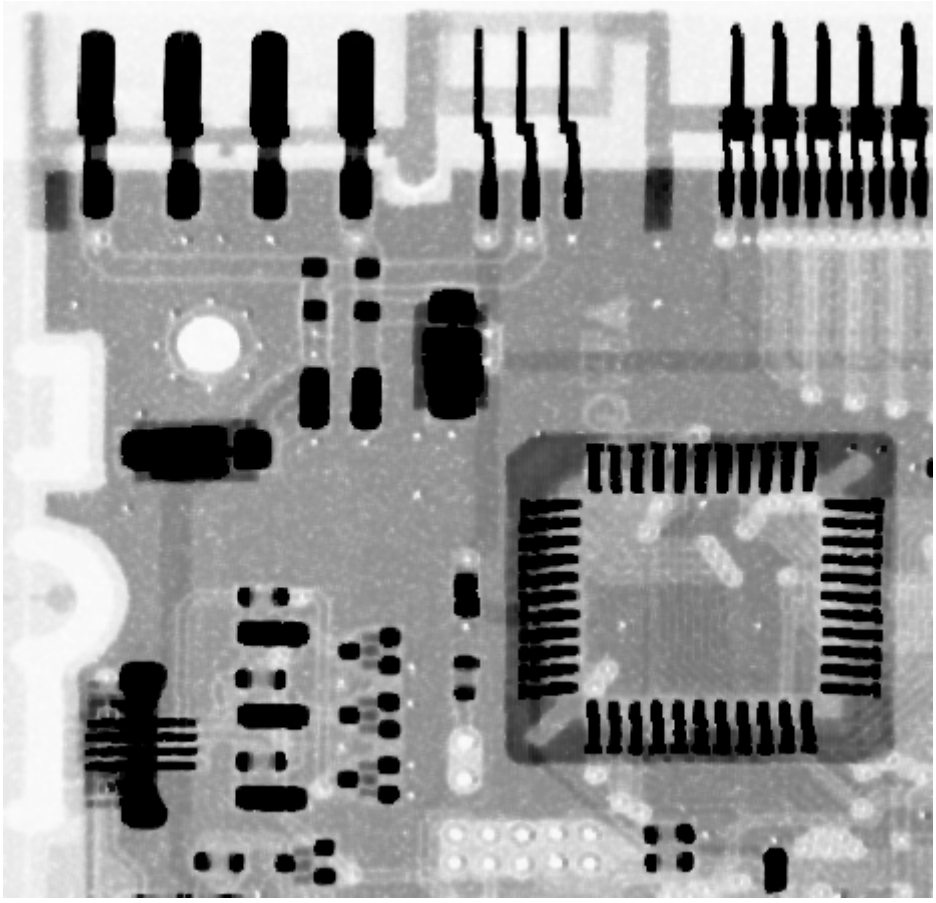
3x3逆谐波均值滤波器(Q = -1.5)



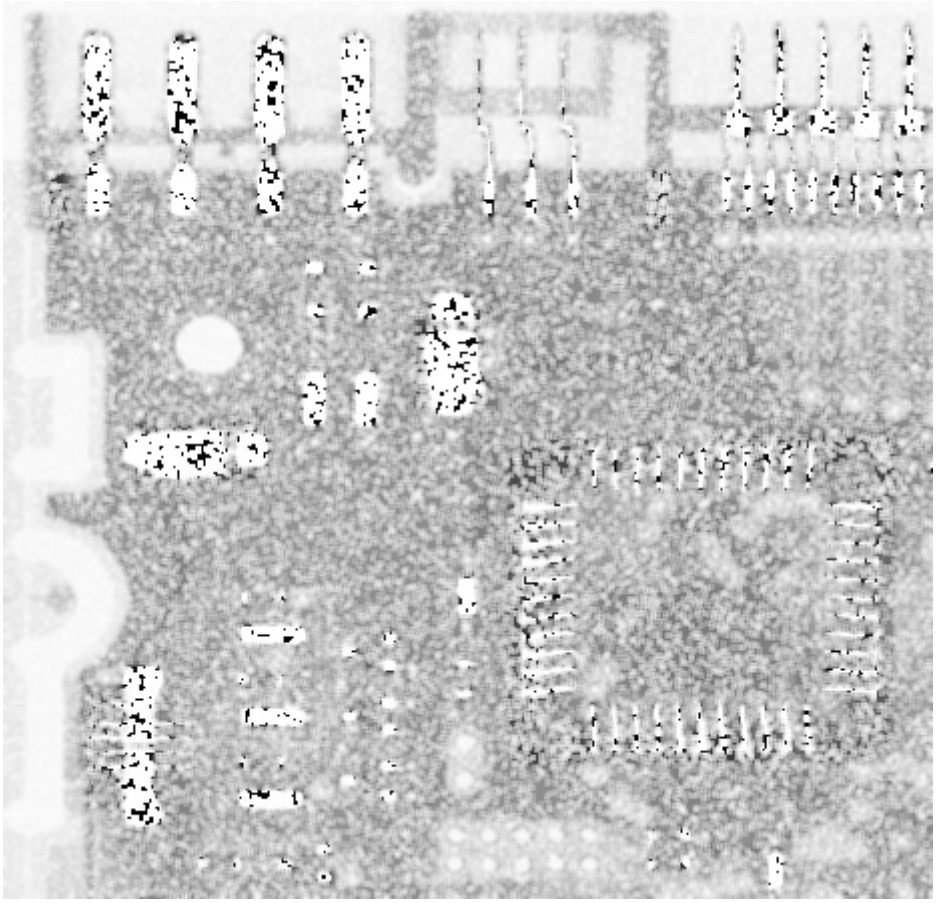
3x3逆谐波均值滤波器(Q = -2.0)



3x3逆谐波均值滤波器(Q = -2.5)



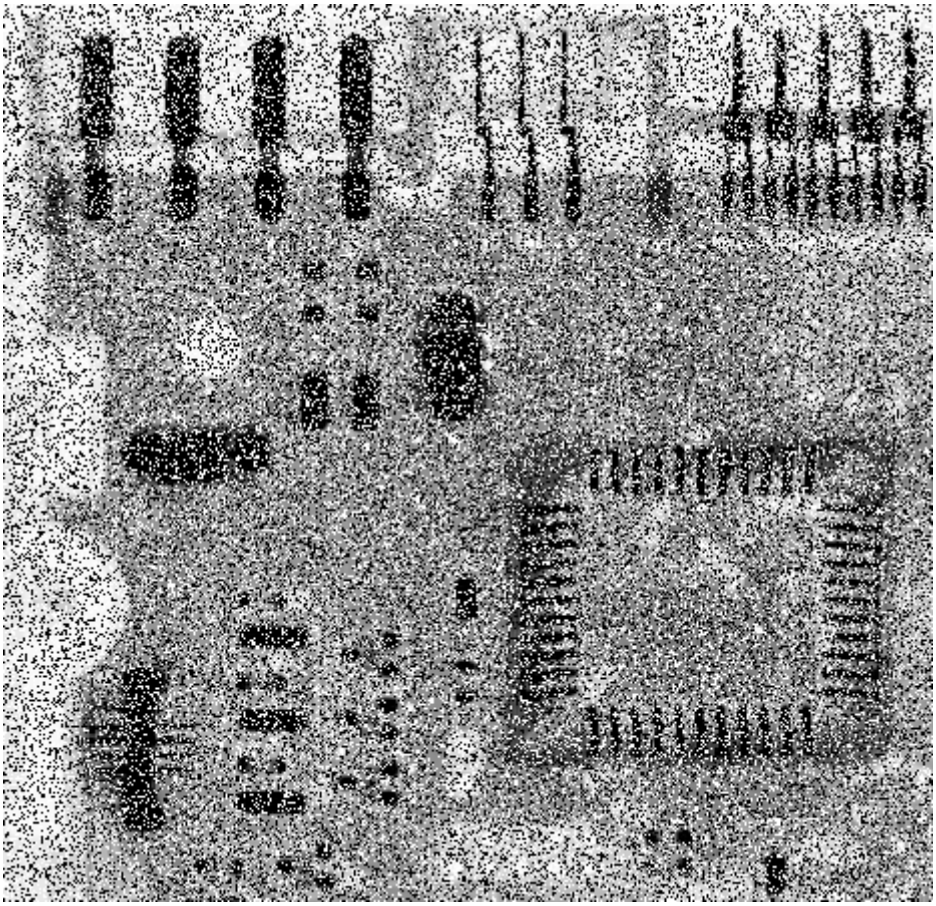
3x3逆谐波均值滤波器(Q = 1.5)



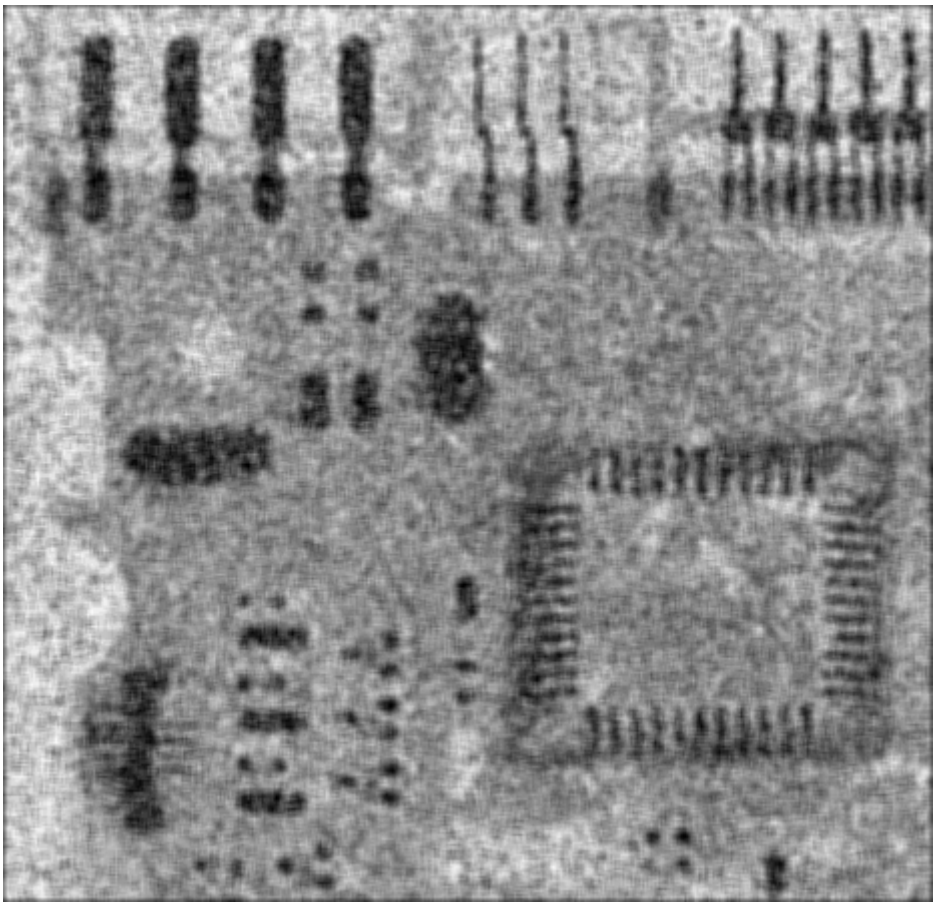
当Q取正值,图片严重失真。原因在于,当Q为正数时,各像素点上的灰度值直接相加,在经过幂运算之后,对于新的灰度值影响更大，由此会得到很多亮白色的点。故一般将Q>0用于椒噪声, 将Q<0用于盐噪声。

4.

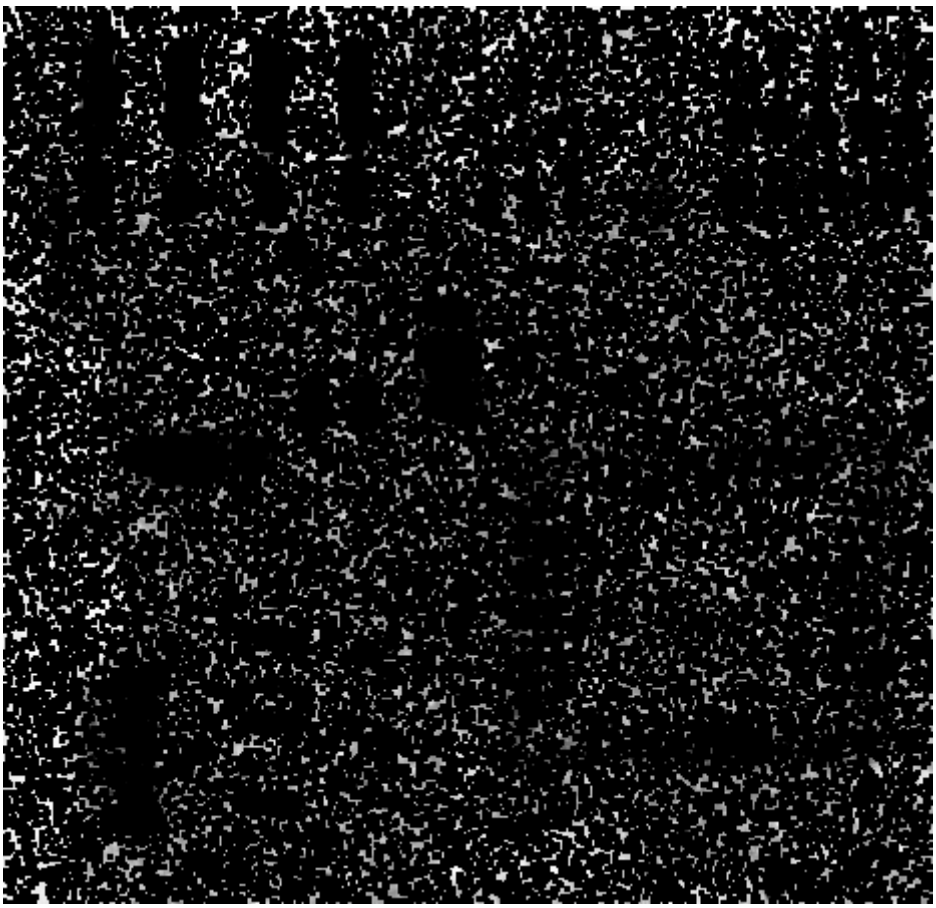
添加椒盐噪声



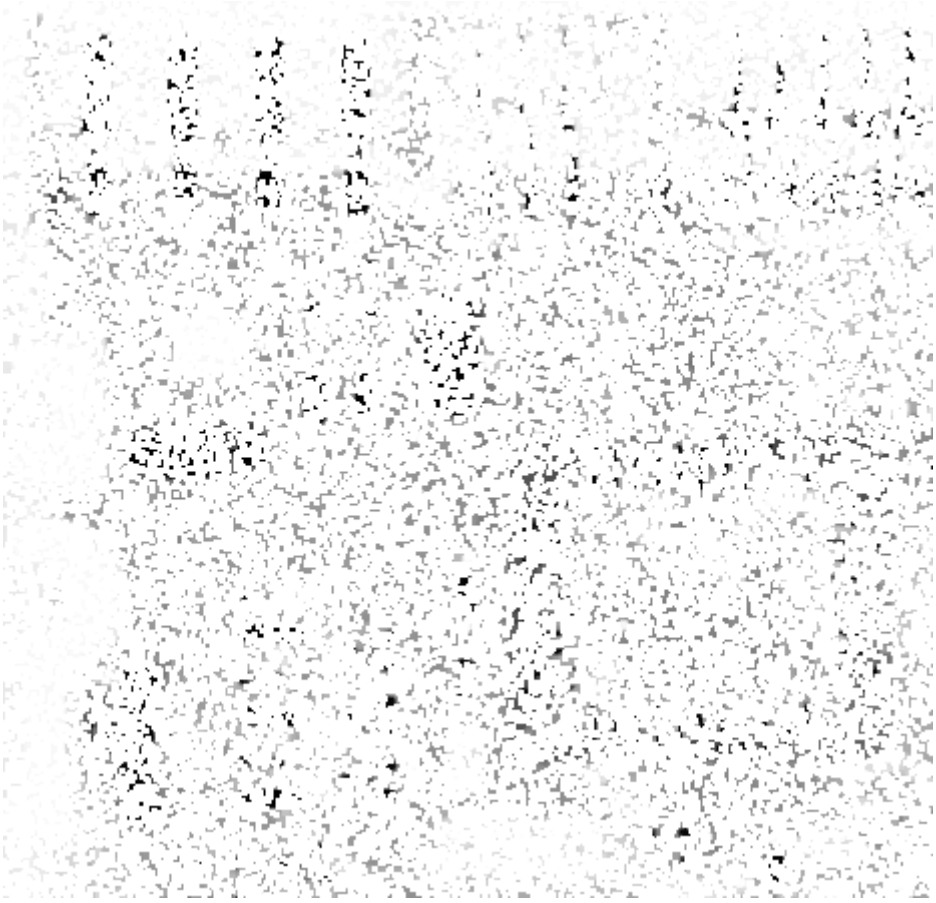
算术均值（5x5）



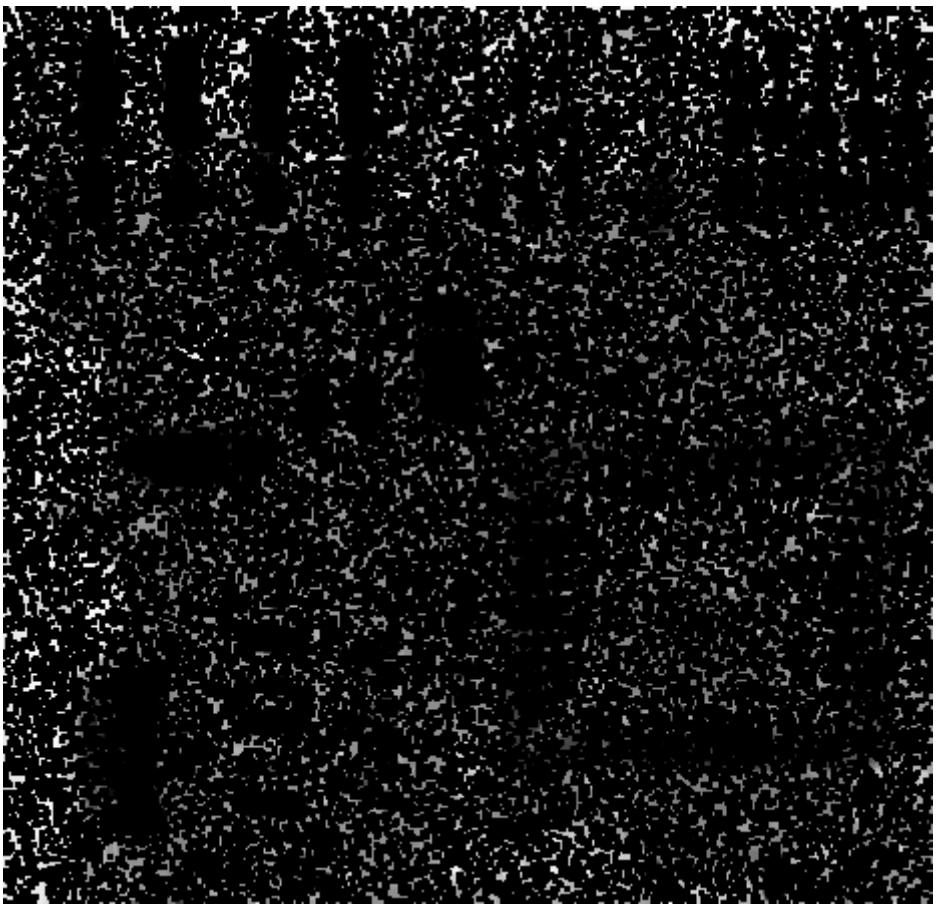
几何均值(3x3)



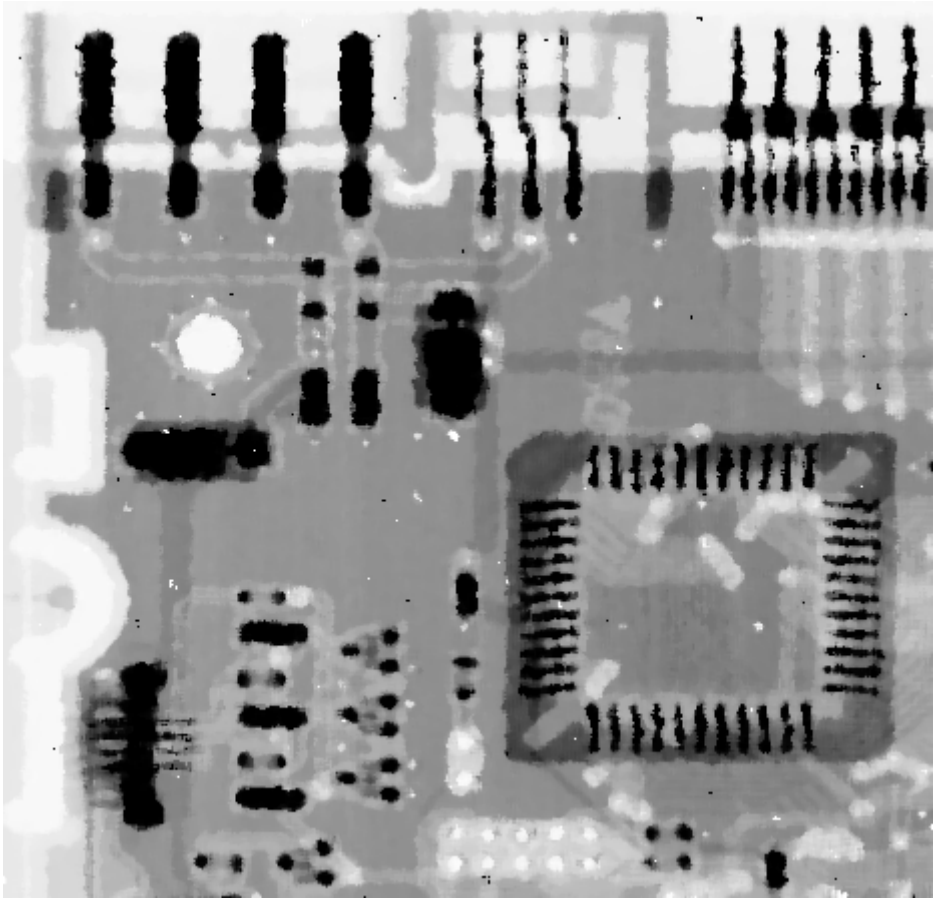
最大值濾波(3x3)



最小值濾波(3x3)



中值滤波（5x5）



几何，最大值，最小值的滤波结果惨不忍睹，原因应该是椒盐噪声的概率比较大，所用的滤波器中极值出现的概率很大，而这三个滤波方法受极值的影响很大，所以效果不好。

算术均值5x5比3x3要好，原因是是椒盐噪声的概率比较大，选用大一点的滤波器，更加能够中和椒盐两端的机制。

中值处理最好，因为它收到极致的影响最小。

5. 算术均值滤波器调用作业2的接口。
   另外四个滤波器的实现非常相似，主要是只要找好了filter的范围，根据课本上的公式就可以算出来。

```
for x in range(height):
    for y in range(width):
        rst = x - a
        red = x + a + 1
        cst = y - b
        ced = y + b + 1

        if rst < 0:
            rst = 0
        if red > height:
            red = height
        if cst < 0:
            cst = 0
        if ced > width:
            ced = width

        submatrix = img[rst:red, cst:ced].reshape(
            (1, (red - rst) * (ced - cst)))
```

所以只要构造好正确的submatrix，既可以完成。

几何均值滤波

```
            submatrix = img[rst:red, cst:ced]
                        temp = 1
                        m = red - rst
                        n = ced - cst
                        for i in range(m):
                            for j in range(n):
                                if temp == 0:
                                    break
                                temp *= submatrix[i][j]
                            if temp == 0:
                                break

                        newIm.putpixel((y, x), int(
                            pow(temp, 1 / (m * n))))
```

调和均值滤波，使用Q为-1的逆谐波滤波器来处理。

逆谐波均值滤波，按照以下公式实现即可：

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

最大值滤波，取滤波器中的最大值

最小值滤波，取滤波器中的最小值

中值滤波：

```
submatrix = img[rst:red, cst:ced].reshape((1, (red
 - rst) * (ced - cst)))
newIm.putpixel((y, x), int(median(submatrix[0])))
```

详细的实现可见 `filters.py`

2.4 彩色图像的均衡化

1.

RGB分别直方图均衡化再合并.png

2.

RGB分别直方图取平均直方图再映射

3.

RGB转HSI处理后再转回RGB

4.

总体而言,方法1和方法2整体的色调和原图有明显差别，尤其是车身上的黄色。方法3能较好的保持原图主体的色调。原图中除去车的背景较暗，直方图均衡化后,图片整体变亮，背景变亮明显。方法3的汽车和枫叶的颜色鲜艳突出，车身的颜色有点点过亮。

原因如下:

1. 方法1分别将3通道的直方图均衡化,再重组。同一幅图中,3个通道的直方图可能差异较大,均衡化后,rgb三个值可能会根据各自的直方图均衡化被映射到相差较大的值，因此新构成的点R,G,B值所占权值会有很大改变，同一个像素点上的rgb三值可能会变化很大，组合出来的颜色也就会变化很大，导致整体色调的变化。
2. 方法2与方法1不同之处在于, 先根据三个直方图计算出平均直方图, 对这个直方图来做均衡化，再用结果来对三个通道处理，这样的话三个通道的映射规则是统一的，在一定程度上降低了方法1中rgb新映射到差异很大的值发生的概率，但是实际实验的效果和方法1相比不明显，可能是因为此图的本来rgb三个值的分布就比较平均，三通道各自的直方图的差异不大。
3. 在方法3中,RGB空间先转换到了HSI空间,再对强度通道进行均衡化。这样处理仅仅改变了图像的强度,对于整体色调的保持最好。