SPATIAL

Filtering to Remove Noise

We can use spatial filters of different kinds to remove different kinds of noise

The *arithmetic mean* filter is a very simple one and is calculated as follows:

$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$
$^1/_9$	$^1/_9$	$^1/_9$

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

Here s_{xy} represents coordinates in a rectangular sub image window(mask).

This is implemented as the simple smoothing filter

- Blurs the image to remove noise
- Best works for Gaussian, Uniform and Erlang noise

Other Means

There are different kinds of mean filters all of which exhibit slightly different behaviour:

- Geometric Mean
- Harmonic Mean
- Contraharmonic Mean

Other Means (cont…)

There are other variants on the mean which can give different performance

Geometric Mean:

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

Here product of the pixels in a window raised by power 1/mn.

Achieves similar smoothing to the arithmetic mean, but tends to lose less image detail.

- **Geometric Mean:**

Important points:

a. Blurring effect is introduced in the processed image

b. Best works for Gaussian, Uniform and Erlang noise

c. Loose less image detail during the processing

Other Means (cont…)

Harmonic Mean:

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

Works well for salt noise, but fails for pepper noise

Also does well for other kinds of noise such as Gaussian noise

Other Means (cont…)

Contraharmonic Mean:

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

Q is the *order* of the filter and adjusting its value changes the filter's behaviour

Positive values of Q eliminate pepper noiseNegative values of Q eliminate salt noise. (But it cant do simultaneously).

If Q=0 then it works as AM, if Q=-1 then it reduces to HM

- **Contraharmonic Mean:**

Q: Order of the filter

Q>0 => Positive order filter: Eliminate pepper noise ==> Blurring of the dark areas

Q<0 => Negative order filter: Eliminate salt noise ==> Blurring of the bright areas

Q=0 => Arithmetic mean filter

Q=-1 => Harmonic mean filter ==> Suitable for impulse noise

Noise Removal Examples

Original Image

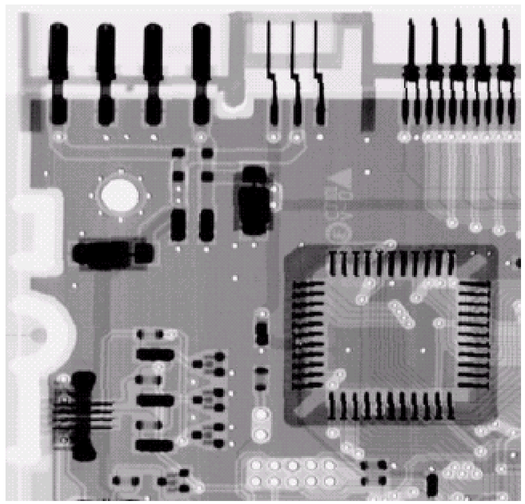
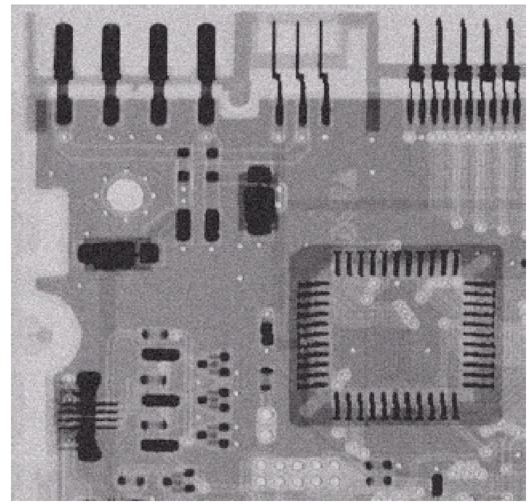
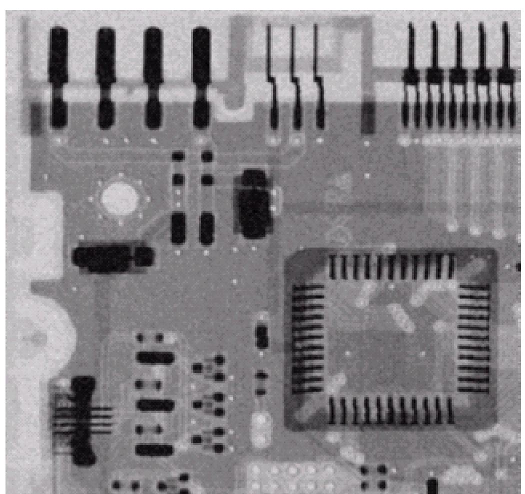


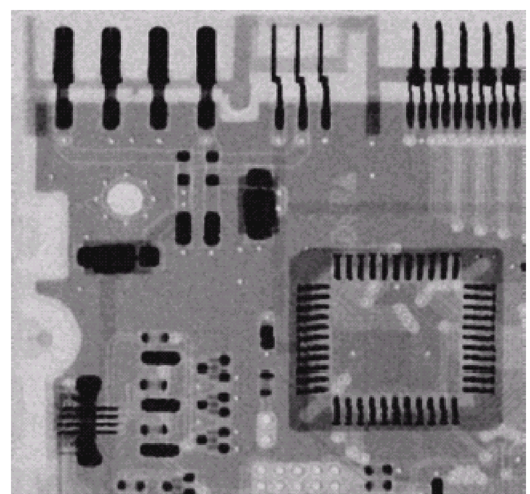
Image Corrupted By Gaussian Noise



After A 3*3 Arithmetic Mean Filter

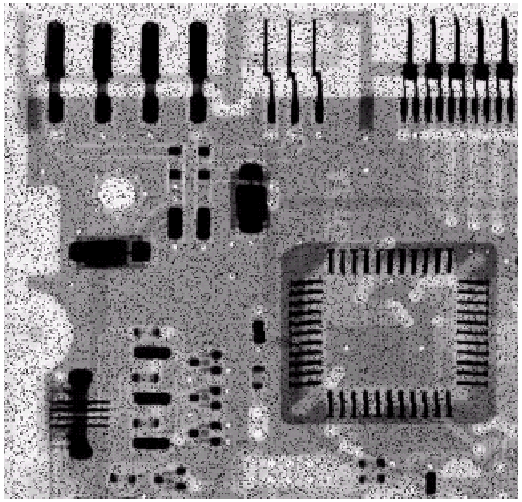


After A 3*3 Geometric Mean Filter

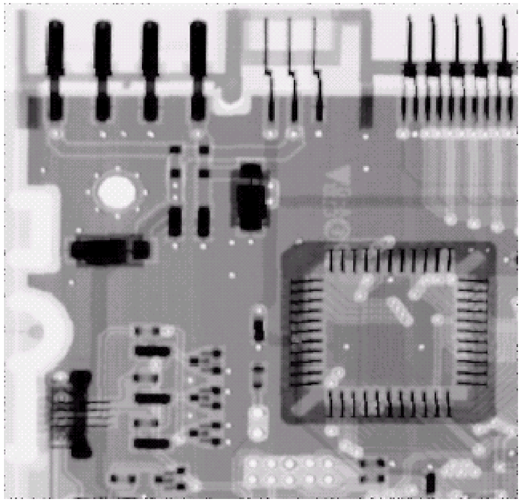


Noise Removal Examples (cont…)

Image Corrupted By Pepper Noise



Result of Filtering Above With 3*3 Contraharmonic Q=1.5



Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Noise Removal Examples (cont…)

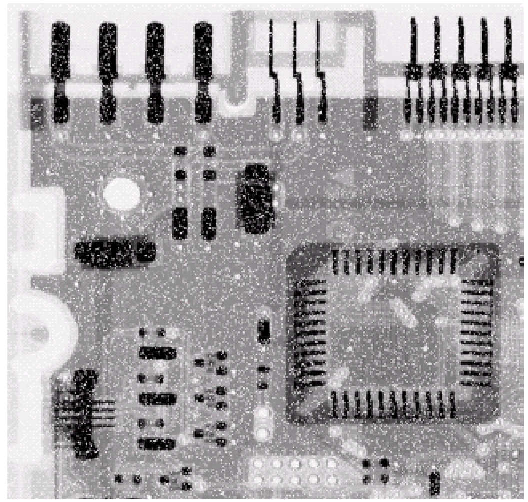
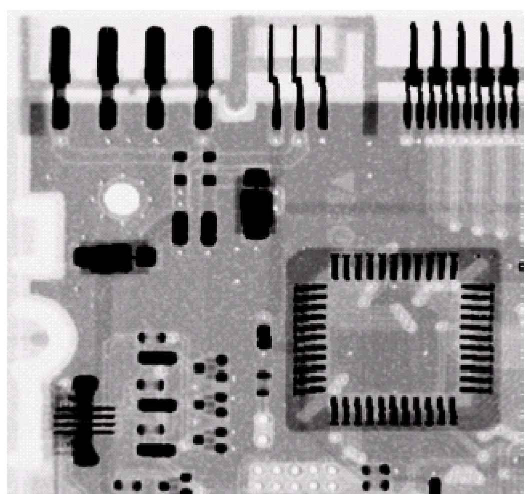


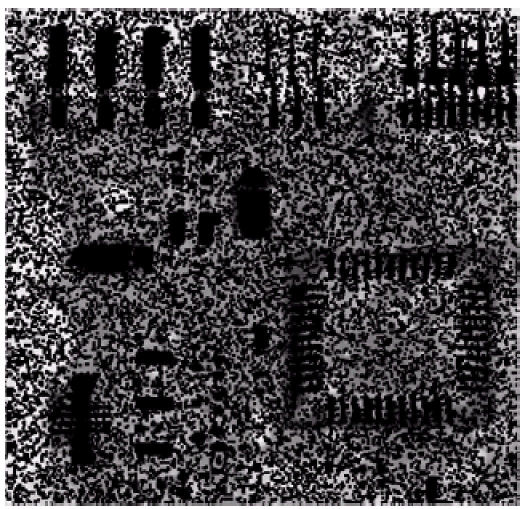
Image Corrupted By Salt Noise

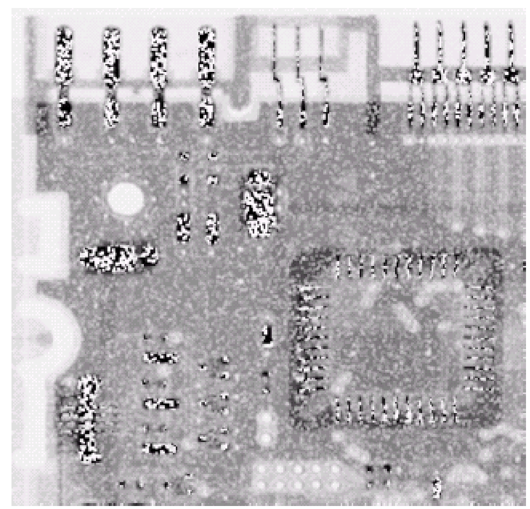


Result of Filtering Above With 3*3 Contraharmonic Q=-1.5

Contra harmonic Filter

Choosing the wrong value for Q when using the contraharmonic filter can have drastic results





Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Order Statistics Filters

Spatial filters that are based on ordering the pixel values that make up the neighbourhood operated on by the filter

Useful spatial filters include

- Median filter
- Max and min filter
- Midpoint filter
- Alpha trimmed mean filter

Median Filter

Median Filter:

$$\hat{f}(x, y) = \underset{(s,t)\in S_{xy}}{median}\{g(s,t)\}$$

Excellent at noise removal, without the smoothing effects that can occur with other smoothing filters

Particularly good when salt and pepper noise is present

Max and Min Filter

Max Filter:

$$\hat{f}(x,y) = \max_{(s,t)\in S_{xy}} \{g(s,t)\}$$

Min Filter:

$$\hat{f}(x,y) = \min_{(s,t)\in S_{xy}} \{g(s,t)\}$$

Max filter is good for pepper noise and min is good for salt noise

Midpoint Filter

Midpoint Filter:

$$\hat{f}(x,y) = \frac{1}{2}\left[\max_{(s,t)\in S_{xy}} \{g(s,t)\} + \min_{(s,t)\in S_{xy}} \{g(s,t)\} \right]$$

Good for random Gaussian and uniform noise

Alpha-Trimmed Mean Filter

Alpha-Trimmed Mean Filter:

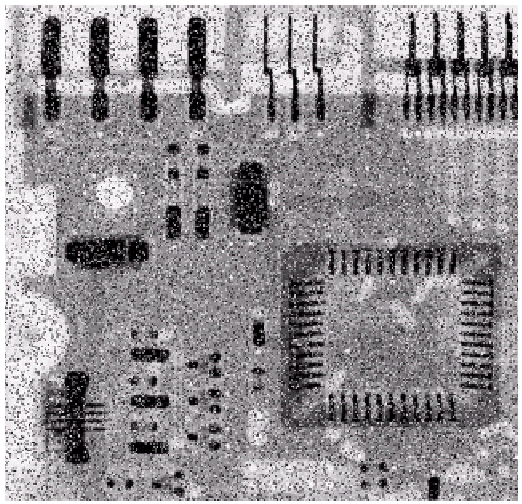
$$\hat{f}(x, y) = \frac{1}{mn - d} \sum_{(s,t)\in S_{xy}} g_r(s, t)$$

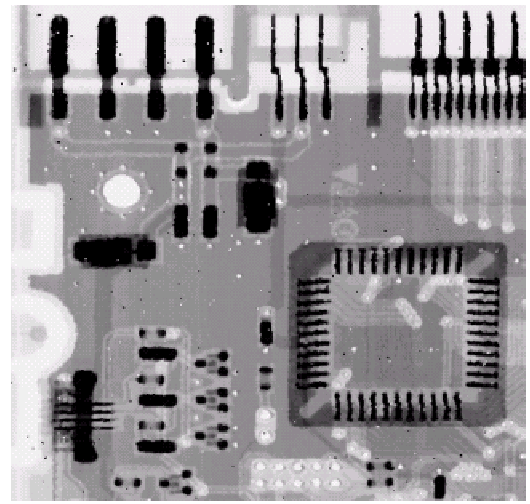
We can delete the $d/2$ lowest and $d/2$ highest grey levels

So $g_r(s, t)$ represents the remaining $mn - d$ pixels

Noise Removal Examples

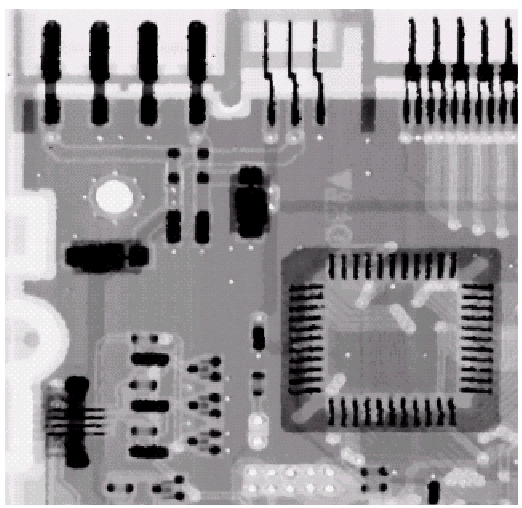
Image Corrupted By Salt And Pepper Noise

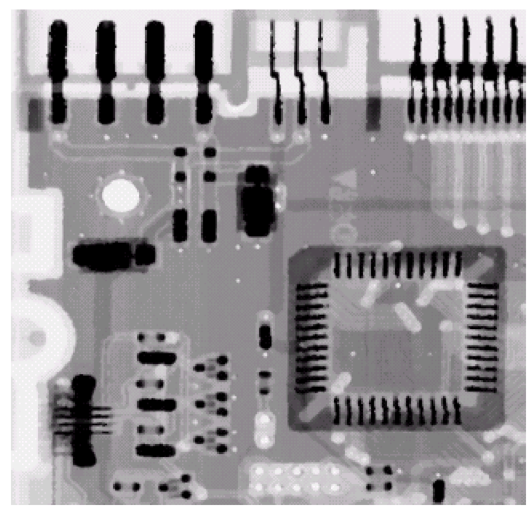




Result of 1 Pass With A 3*3 Median Filter

Result of 2 Passes With A 3*3 Median Filter



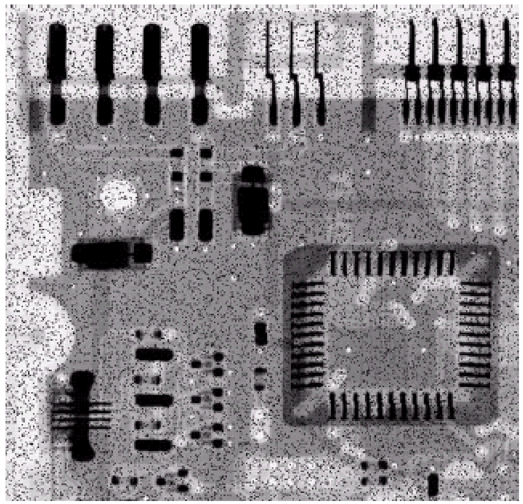


Result of 3 Passes With A 3*3 Median Filter

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Noise Removal Examples (cont…)

Image Corrupted By Pepper Noise



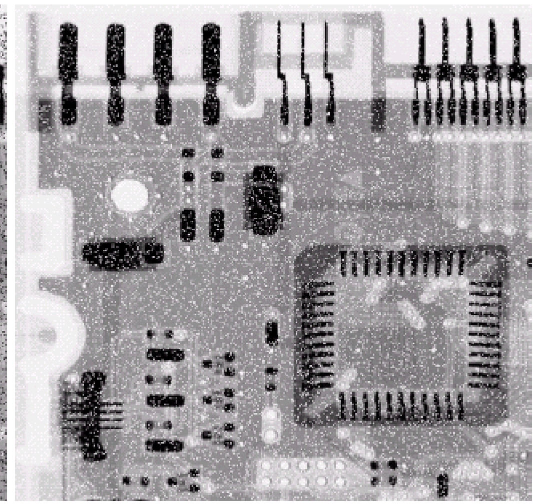
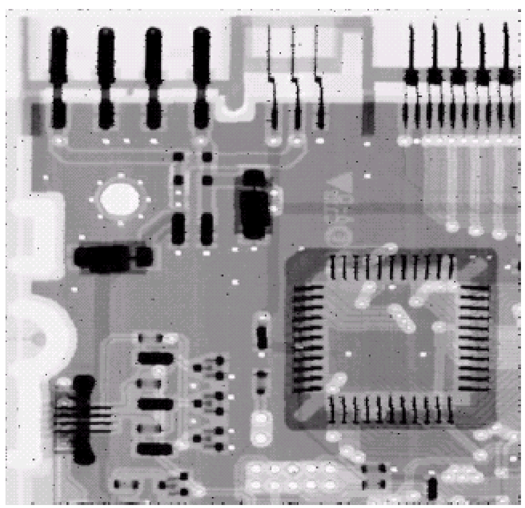
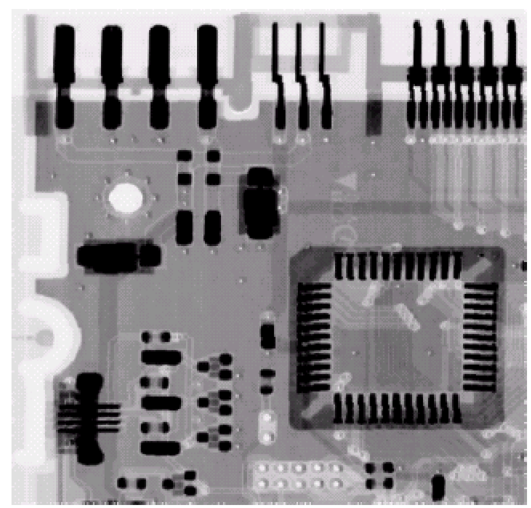


Image Corrupted By Salt Noise

Result Of Filtering Above With A 3*3 Max Filter





Result Of Filtering Above With A 3*3 Min Filter

Noise Removal Examples (cont…)

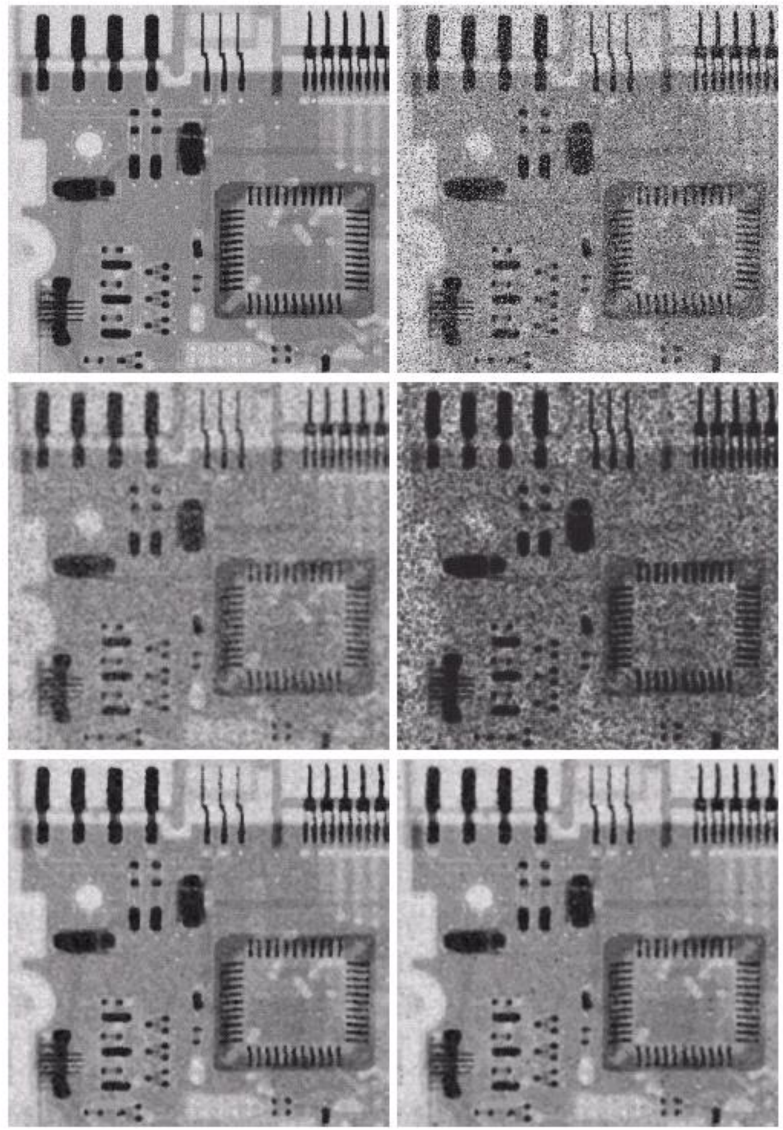


Image Corrupted By Uniform Noise

Image Further Corrupted By Salt and Pepper Noise

Filtered By 5*5 Arithmetic Mean Filter

Filtered By 5*5 Geometric Mean Filter

Filtered By 5*5 Median Filter

Filtered By 5*5 Alpha-Trimmed Mean Filter

Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Calculate Alpha Trimmed Filter

36	38	42	46	14
12	67	87	96	54
53	90	34	23	12

Alpha Trimmed Filter

- The alpha –trimmed filter is the average of the pixel values within the window, but with some of the endpoint –ranked values excluded.

Alpha – trimmed filter = $\dfrac{1}{N^2 - 2\alpha} \sum_{i=\alpha+1}^{N^2-\alpha} I_i$

Where α is the number of pixel values removed from each end of the list , and can range from 0 to $(N^2-1)/2$.