Machine Vision

Lecture 6

Image Restoration

Based on lectures of

Brian Mac Namee

Contents

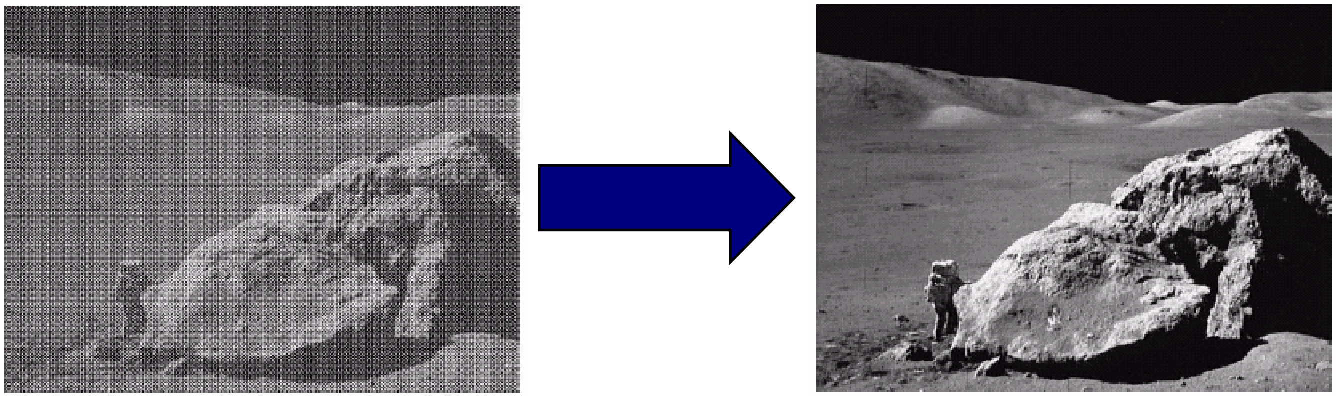
In this lecture we will look at image restoration techniques used for noise removal

- What is image restoration?
- Noise and images
- Noise models
- Noise removal using spatial domain filtering
- Periodic noise
- Noise removal using frequency domain filtering

What is Image Restoration?

Image restoration attempts to restore images that have been degraded

- Identify the degradation process and attempt to reverse it
- Similar to image enhancement, but more objective



Noise and Images

The sources of noise in digital images arise during image acquisition (digitization) and transmission

- Imaging sensors can be affected by ambient conditions
- Interference can be added to an image during transmission

Noise Model

We can consider a noisy image to be modelled as follows:

$$g(x, y) = f(x, y) + \eta(x, y)$$

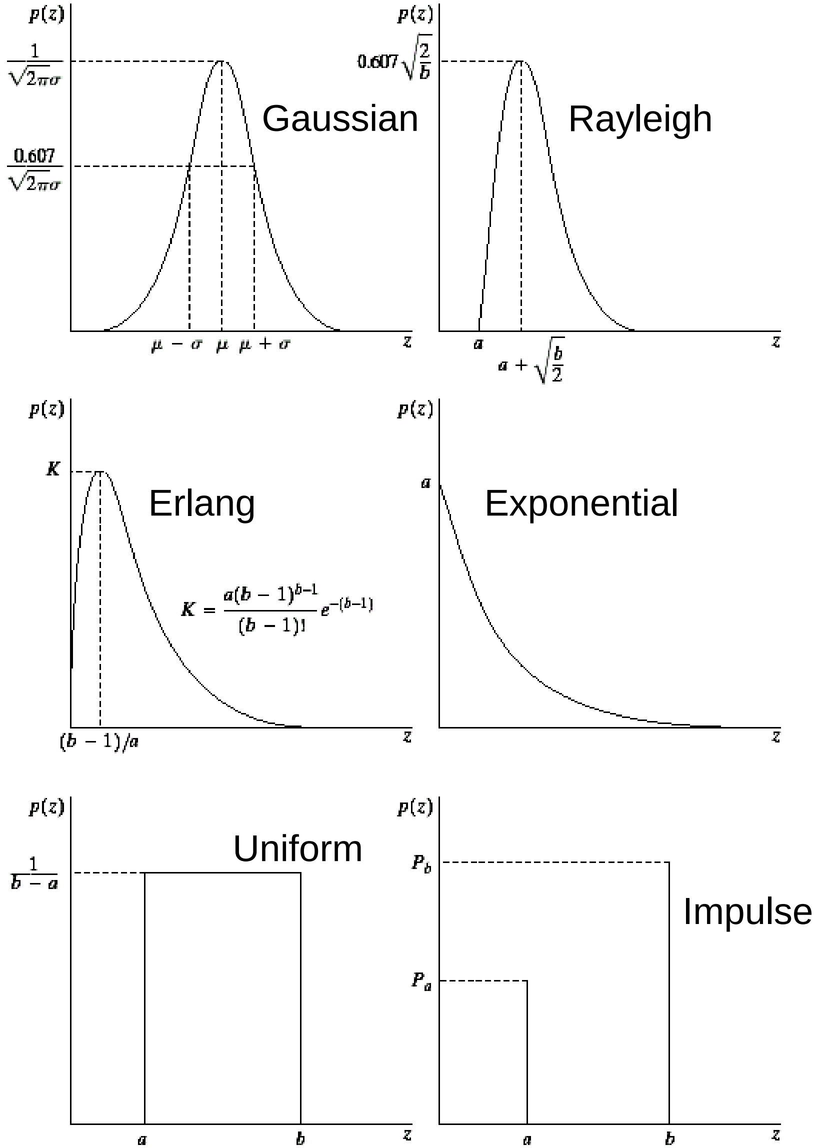
where $f(x, y)$ is the original image pixel, $\eta(x, y)$ is the noise term and $g(x, y)$ is the resulting noisy pixel

If we can estimate the model the noise in an image is based on this will help us to figure out how to restore the image

Noise Models

There are many different models for the image noise term $\eta(x, y)$:

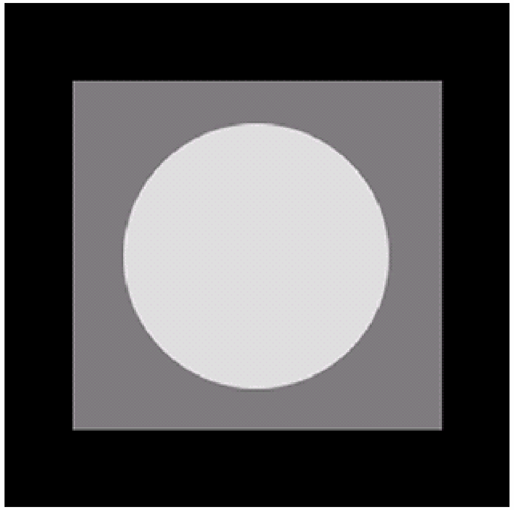
- Gaussian
  - Most common model
- Rayleigh
- Erlang
- Exponential
- Uniform
- Impulse
  - *Salt and pepper* noise



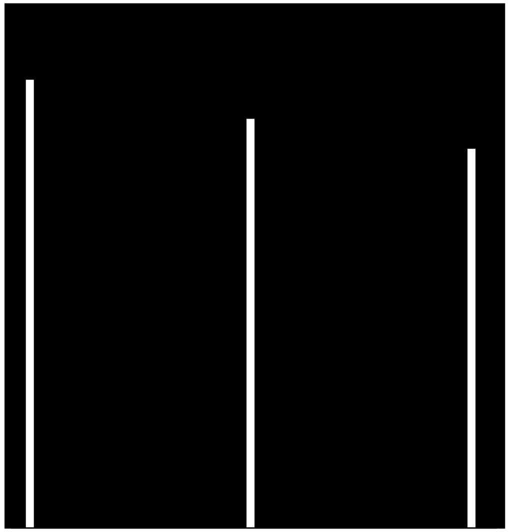
Noise Example

The test pattern to the right is ideal for demonstrating the addition of noise

The following slides will show the result of adding noise based on various models to this image

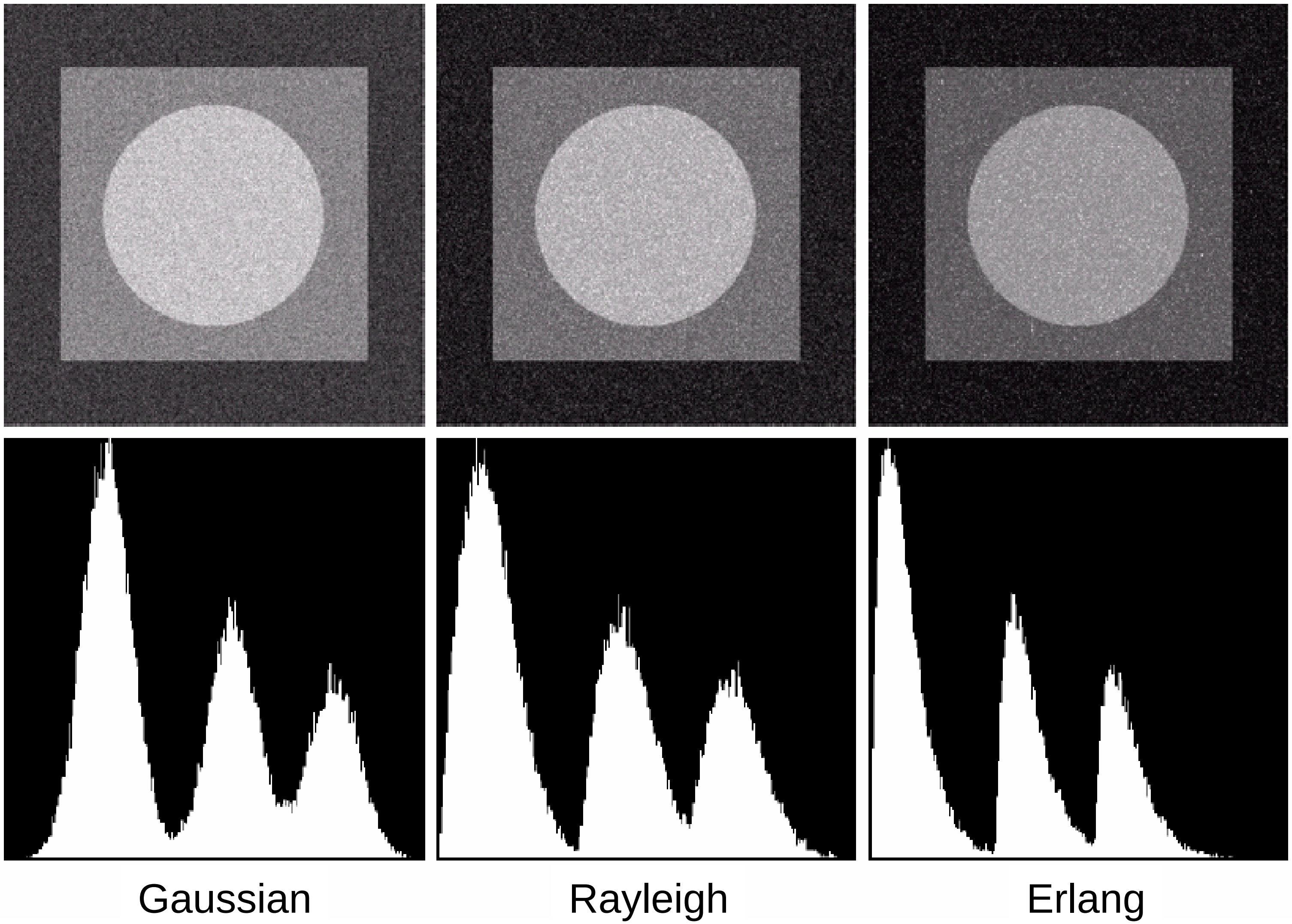


Image

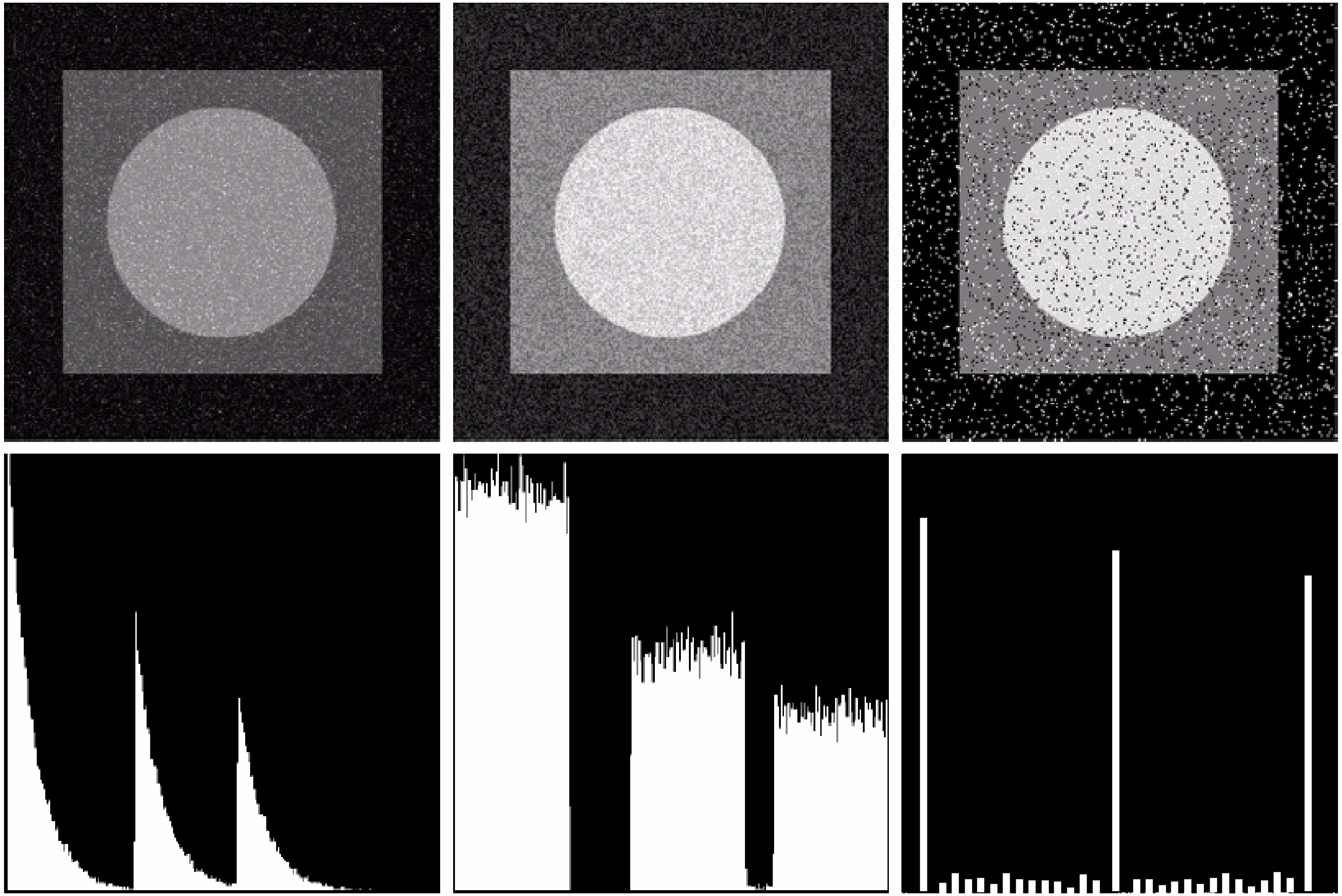


Histogram

Noise Example (cont…)



Noise Example (cont…)



Exponential Uniform Impulse

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:

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

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

This is implemented as the simple smoothing filter

Blurs the image to remove 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:

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

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

Other Means (cont…)

Harmonic Mean:

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

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:

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

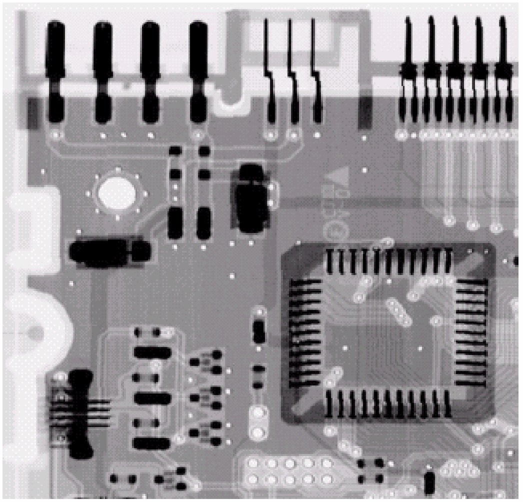
Q is the *order* of the filter and adjusting its value changes the filter's behaviour

Positive values of Q eliminate pepper noise

Negative values of Q eliminate salt noise

Noise Removal Examples

Original Image



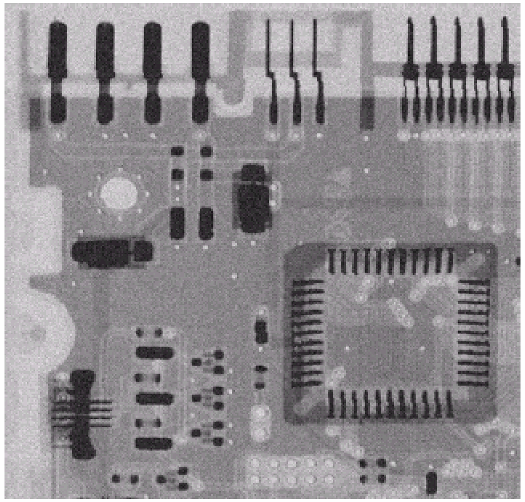
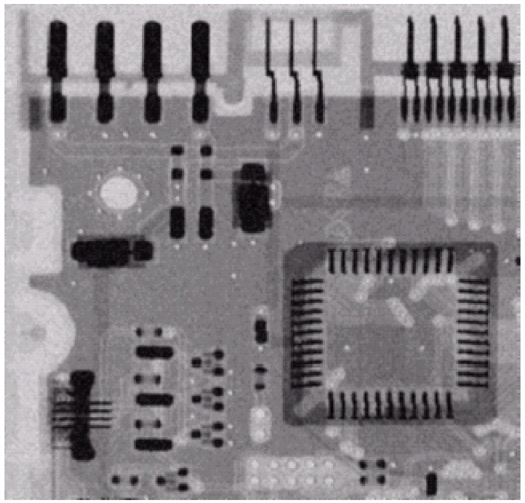
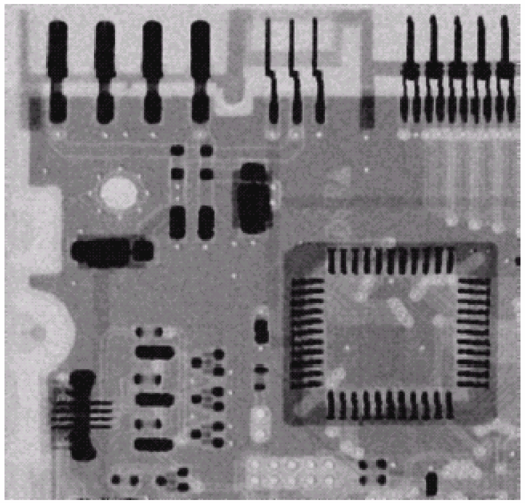


Image Corrupted By Gaussian Noise

After A 3*3 Arithmetic Mean Filter

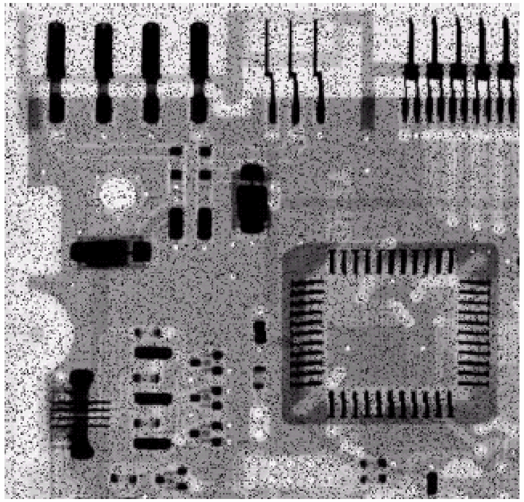




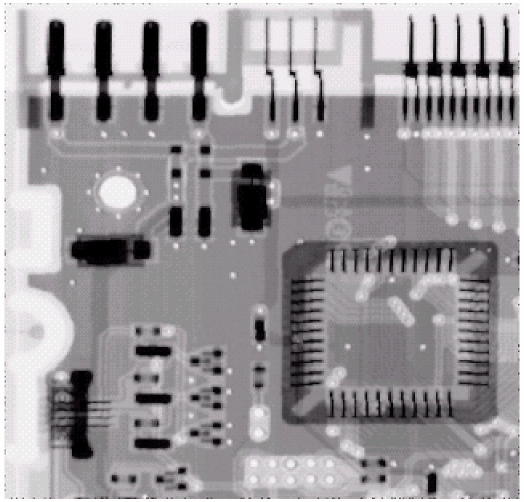
After A 3*3 Geometric Mean Filter

Noise Removal Examples (cont…)

Image Corrupted By Pepper Noise



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



Noise Removal Examples (cont…)

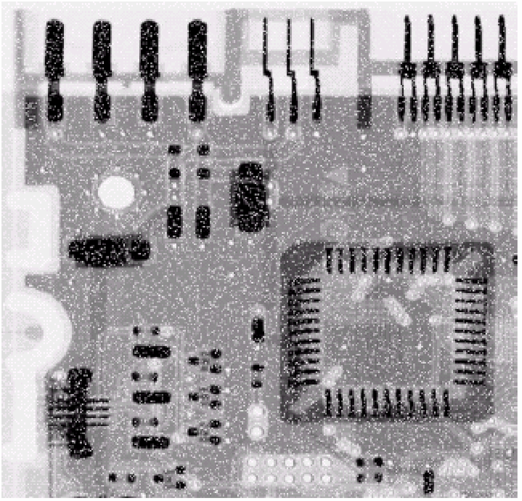
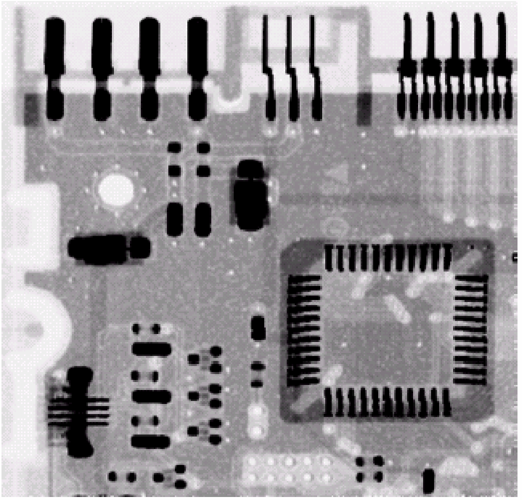


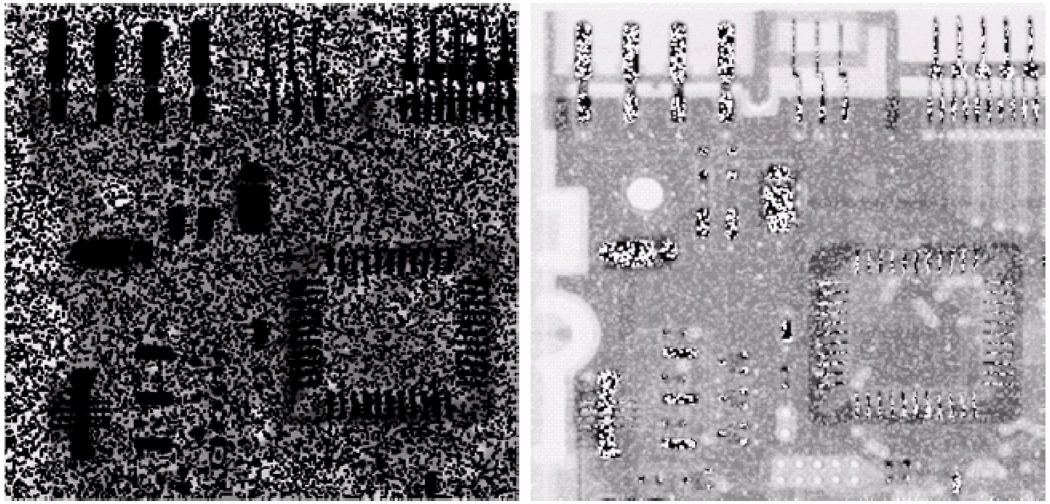
Image
Corrupted
By Salt
Noise



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

Contraharmonic Filter: Here Be Dragons

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



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

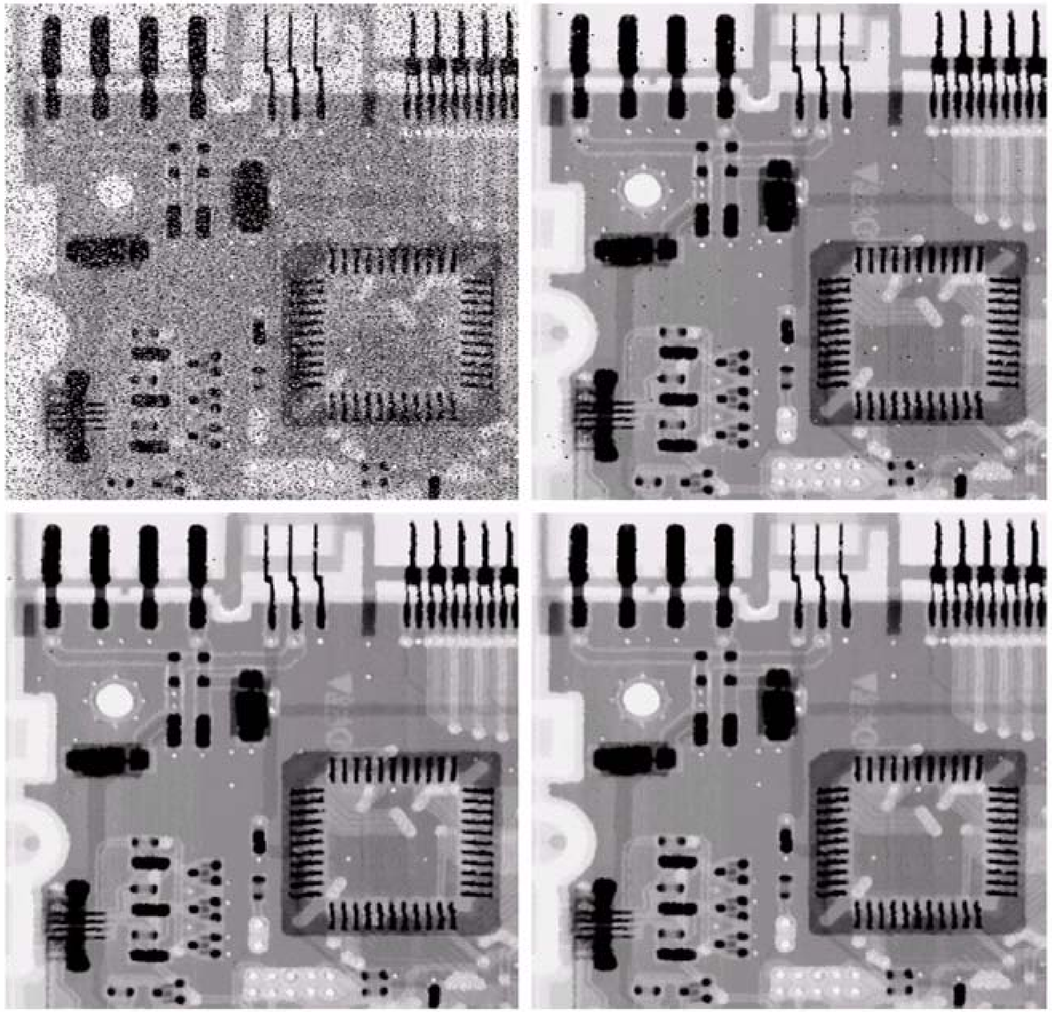
So $g_r(s, t)$ represents the remaining $mn - d$ pixels

Noise Removal Examples

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

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

Noise Removal Examples (cont…)

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

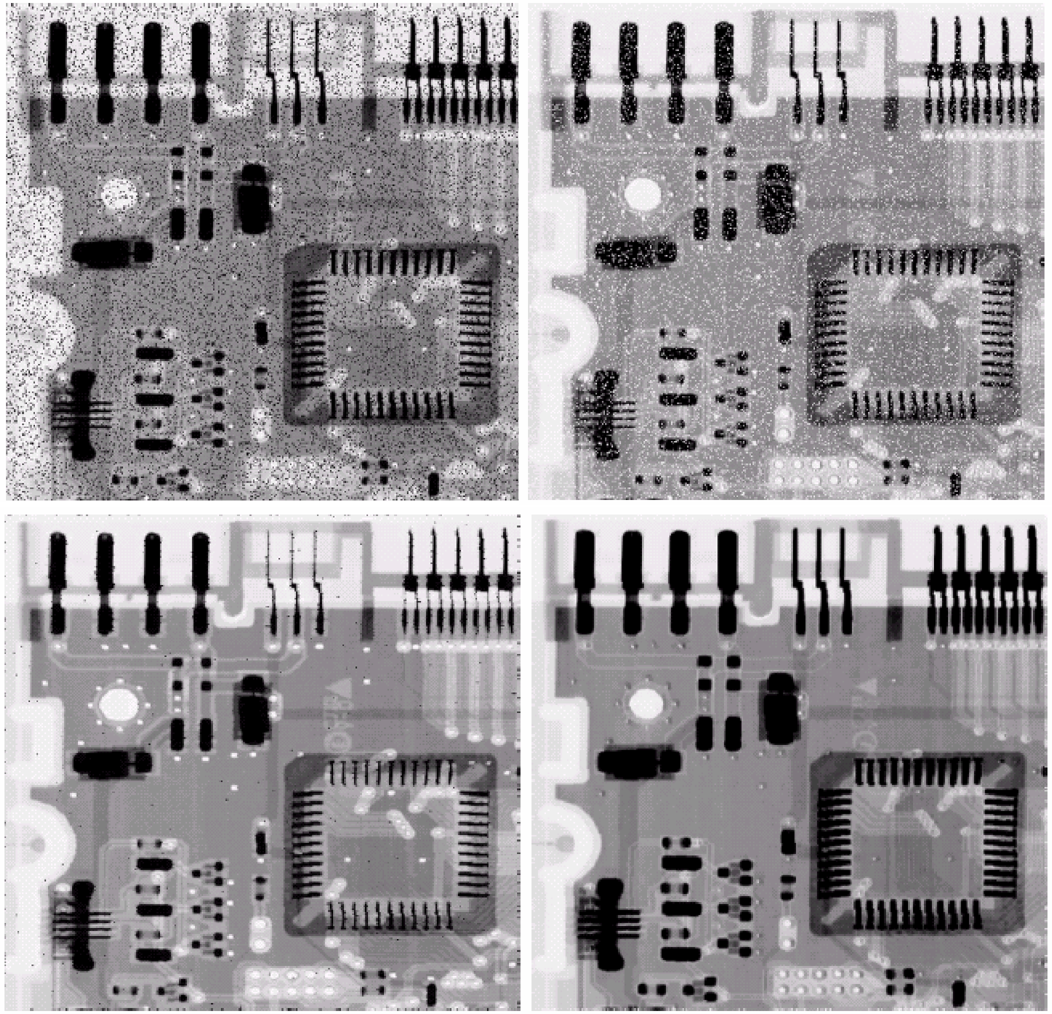


Image Corrupted By Pepper Noise

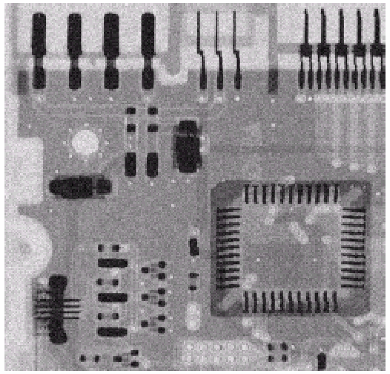
Image Corrupted By Salt Noise

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



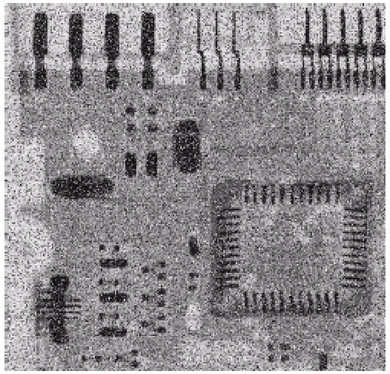
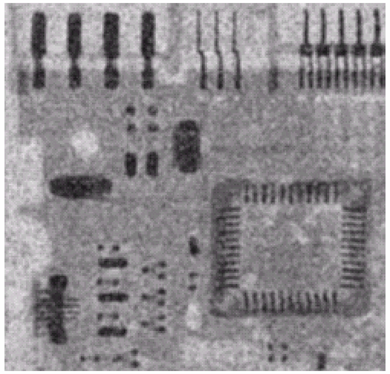
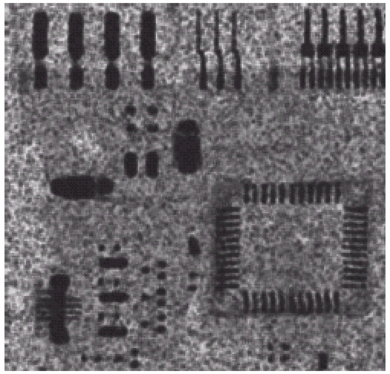


Image Further Corrupted By Salt and Pepper Noise

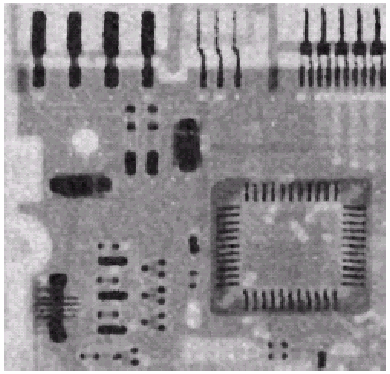
Filtered By 5*5 Arithmetic Mean Filter

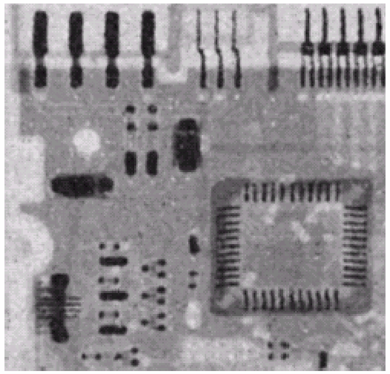




Filtered By 5*5 Geometric Mean Filter

Filtered By 5*5 Median Filter





Filtered By 5*5 Alpha-Trimmed Mean Filter

Adaptive Filters

The filters discussed so far are applied to an entire image without any regard for how image characteristics vary from one point to another

The behaviour of **adaptive filters** changes depending on the characteristics of the image inside the filter region

We will take a look at the **adaptive median filter**

Adaptive Median Filtering

The median filter performs relatively well on impulse noise as long as the spatial density of the impulse noise is not large

The adaptive median filter can handle much more spatially dense impulse noise, and also performs some smoothing for non-impulse noise

The key insight in the adaptive median filter is that the filter size changes depending on the characteristics of the image

Adaptive Median Filtering (cont…)

Remember that filtering looks at each original pixel image in turn and generates a new filtered pixel

First examine the following notation:

- z_{min} = minimum grey level in S_{xy}
- z_{max} = maximum grey level in S_{xy}
- z_{med} = median of grey levels in S_{xy}
- z_{xy} = grey level at coordinates (x, y)
- S_{max} =maximum allowed size of S_{xy}

Adaptive Median Filtering (cont…)

Level A: $A1 = z_{med} - z_{min}$

$A2 = z_{med} - z_{max}$

If $A1 > 0$ and $A2 < 0$, Go to level B

Else increase the window size

If window size $\leq$ repeat S_{max} level A

Else output z_{med}

Level B: $B1 = z_{xy} - z_{min}$

$B2 = z_{xy} - z_{max}$

If $B1 > 0$ and $B2 < 0$, output z_{xy}

Else output z_{med}

Adaptive Median Filtering (cont…)

The key to understanding the algorithm is to remember that the adaptive median filter has three purposes:

– Remove impulse noise

– Provide smoothing of other noise

– Reduce distortion

Adaptive Filtering Example

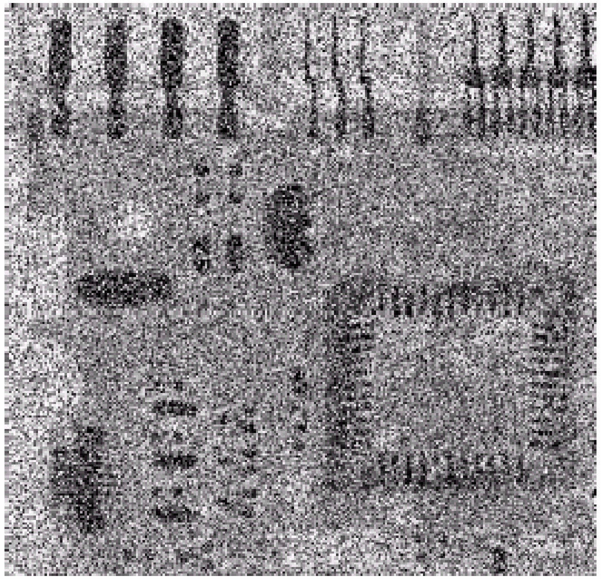
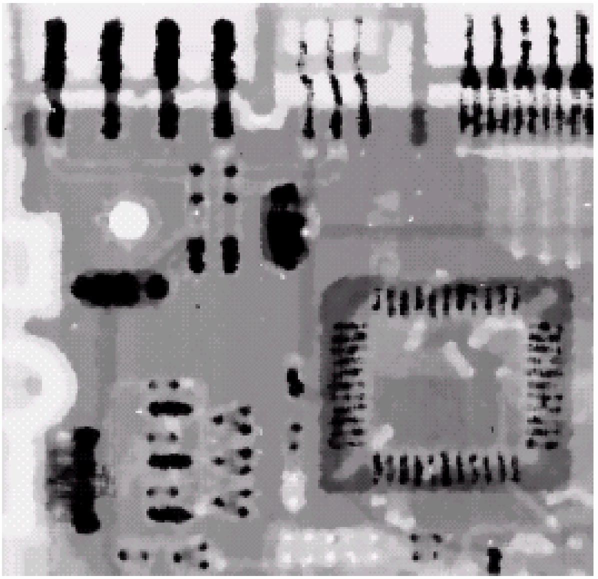
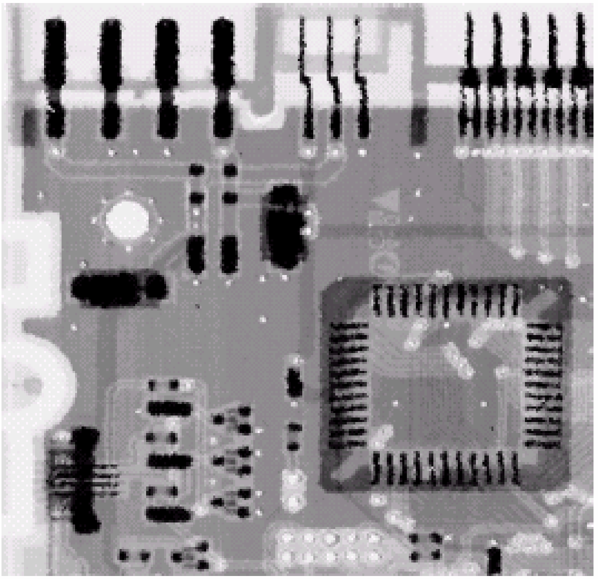


Image corrupted by salt and pepper noise with probabilities $P_a = P_b = 0.25$



Result of filtering with a 7 * 7 median filter



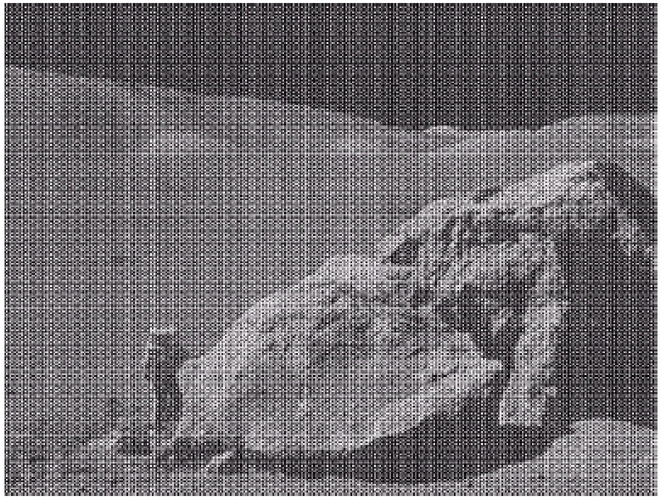
Result of adaptive median filtering with i = 7

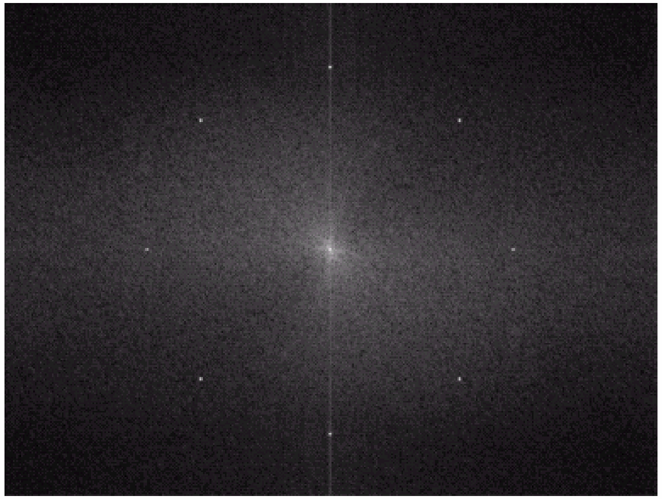
Periodic Noise

Typically arises due to electrical or electromagnetic interference

Gives rise to regular noise patterns in an image

Frequency domain techniques in the Fourier domain are most effective at removing periodic noise





Band Reject Filters

Removing periodic noise form an image involves removing a particular range of frequencies from that image

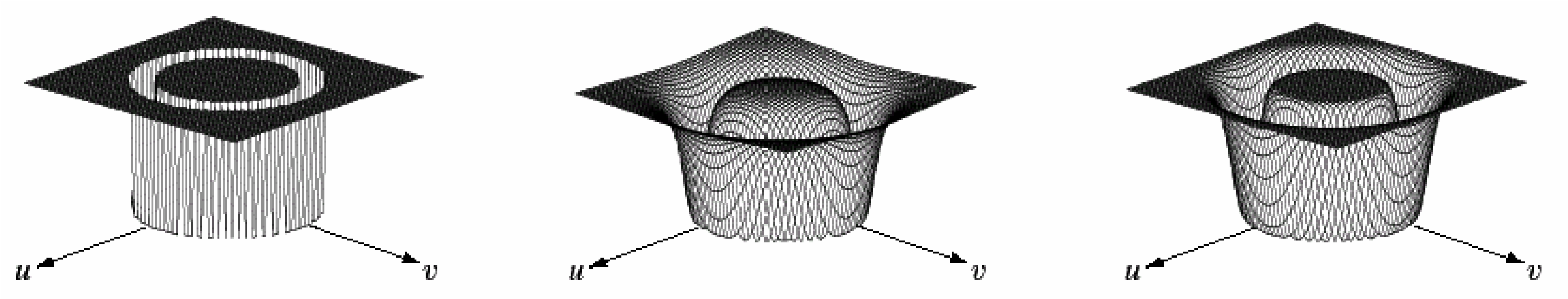
Band reject filters can be used for this purpose

An ideal band reject filter is given as follows:

$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) < D_0 - \dfrac{W}{2} \\ 0 & \text{if } D_0 - \dfrac{W}{2} \le D(u,v) \le D_0 + \dfrac{W}{2} \\ 1 & \text{if } D(u,v) > D_0 + \dfrac{W}{2} \end{cases}$$

Band Reject Filters (cont…)

The ideal band reject filter is shown below, along with Butterworth and Gaussian versions of the filter



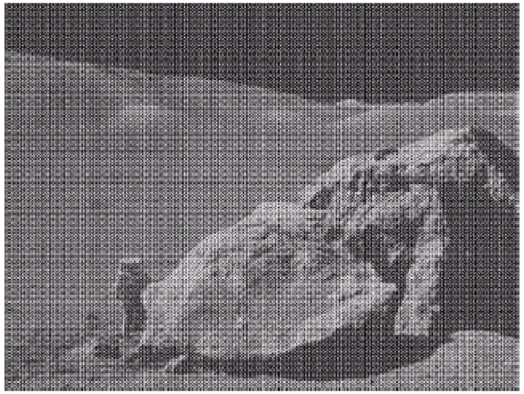
Ideal Band Reject Filter

Butterworth Band Reject Filter (of order 1)

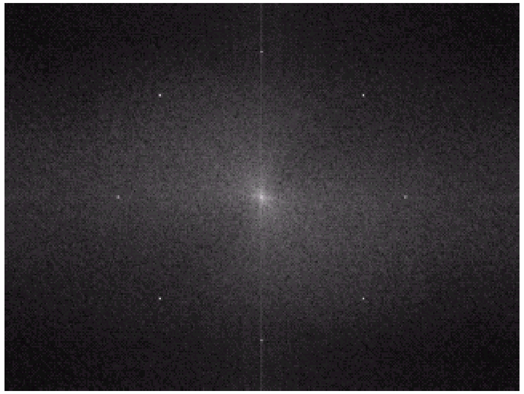
Gaussian Band Reject Filter

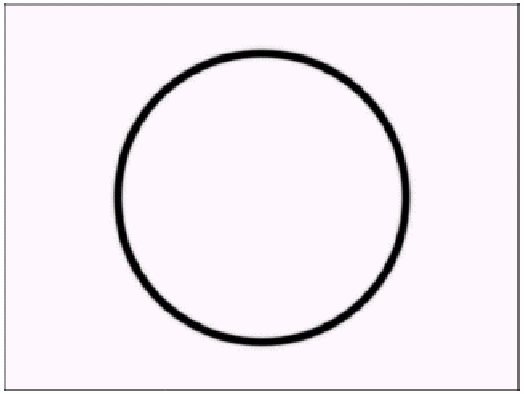
Band Reject Filter Example

Image corrupted by sinusoidal noise



Fourier spectrum of corrupted image





Butterworth band reject filter

Filtered image

Summary

In this lecture we will look at image restoration for noise removal

Restoration is slightly more objective than enhancement

Spatial domain techniques are particularly useful for removing random noise

Frequency domain techniques are particularly useful for removing periodic noise