Digital Image Processing

Image Restoration and Reconstruction (Noise Removal)

Christophoros Nikou
cnikou@cs.uoi.gr

Image Restoration and Reconstruction

Things which we see are not by themselves what we see…

It remains completely unknown to us what the objects may be by themselves and apart from the receptivity of our senses. We know nothing but our manner of perceiving them.

Immanuel Kant

Contents

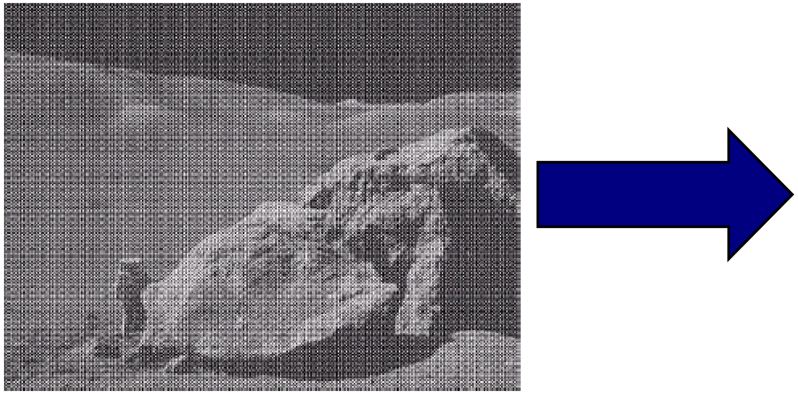
In this lecture we will look at image restoration techniques used for noise removal

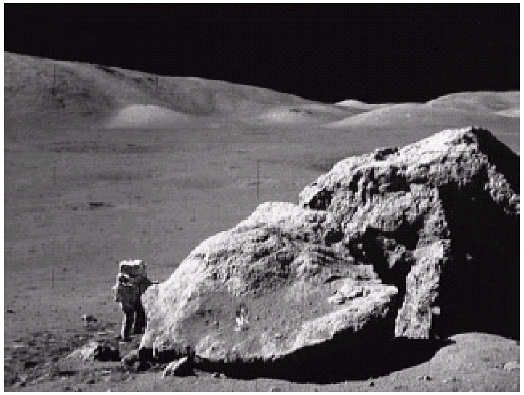
- What is image restoration?
- Noise and images
- Noise models
- Noise removal using spatial domain filtering
- Noise removal using frequency domain filtering

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:

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

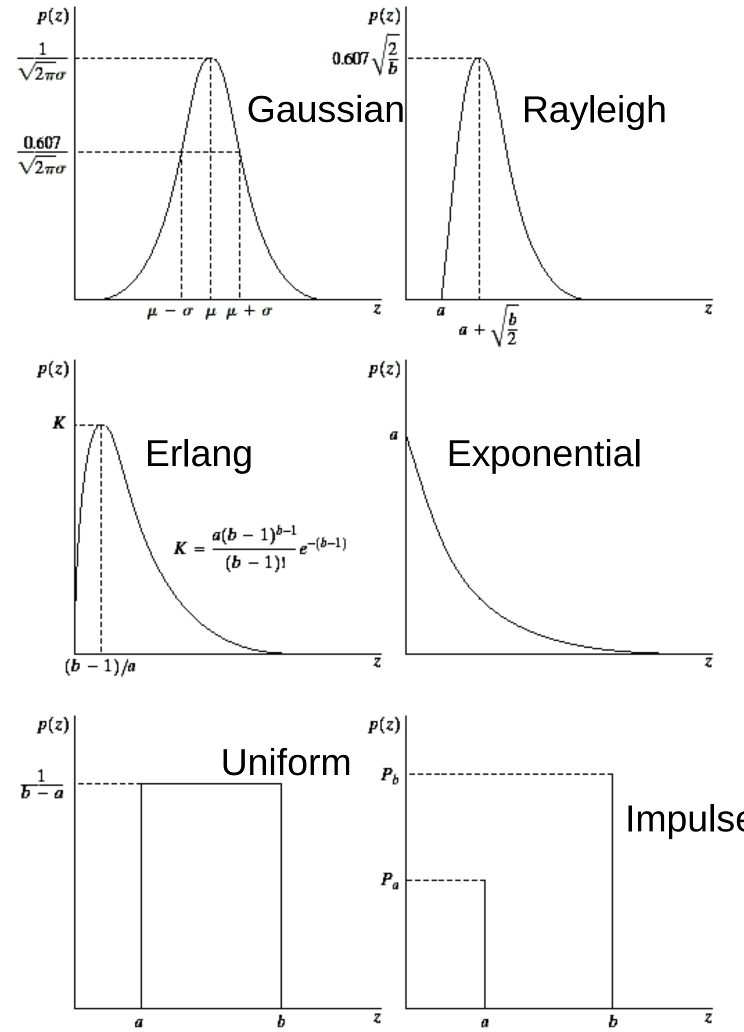
where $f(x, y)$ is the original image pixel, $\eta(x, y)$ is the noise term and $g(x, y)$ is the resulting noisy pixel

If we can estimate the noise model we can figure out how to restore the image

Noise Models (cont…)

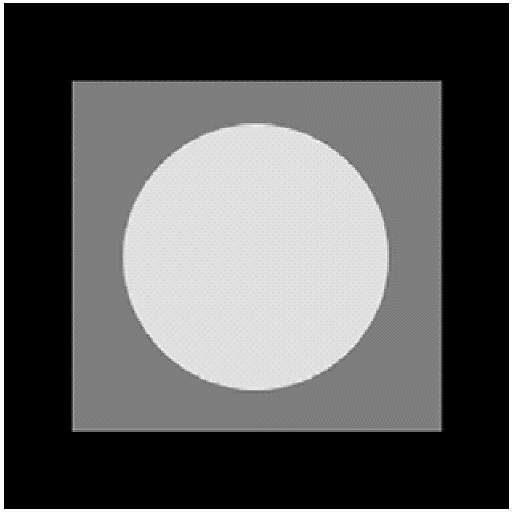
There are many different models for the image noise term $\eta(x, y)$:

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

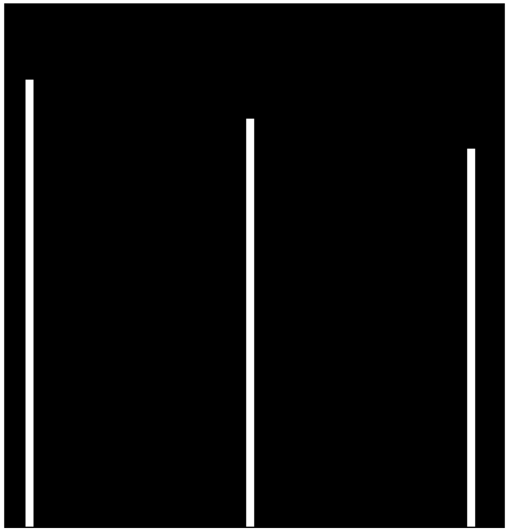


Noise Example

- 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

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

Noise Example (cont…)

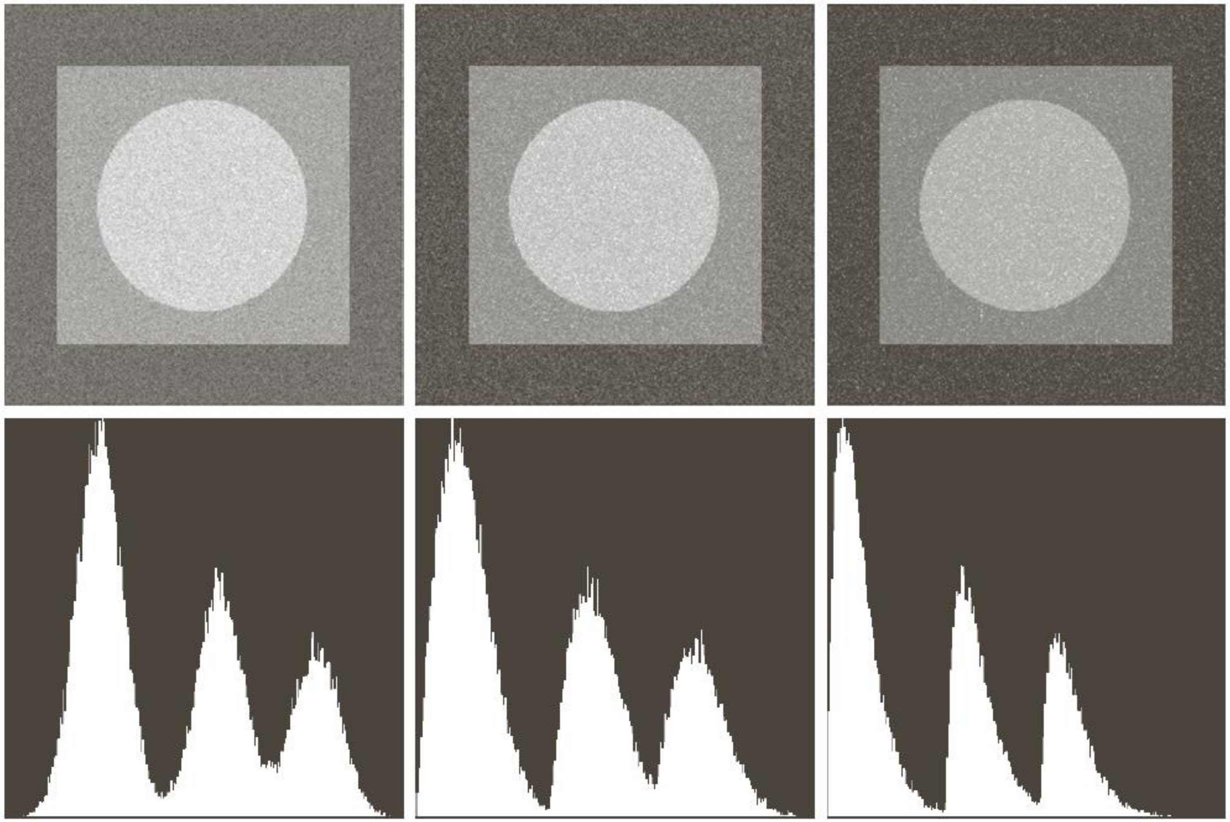


FIGURE 5.4 Images and histograms resulting from adding Gaussian, Rayleigh, and gamma noise to the image in Fig. 5.3.

Noise Example (cont…)

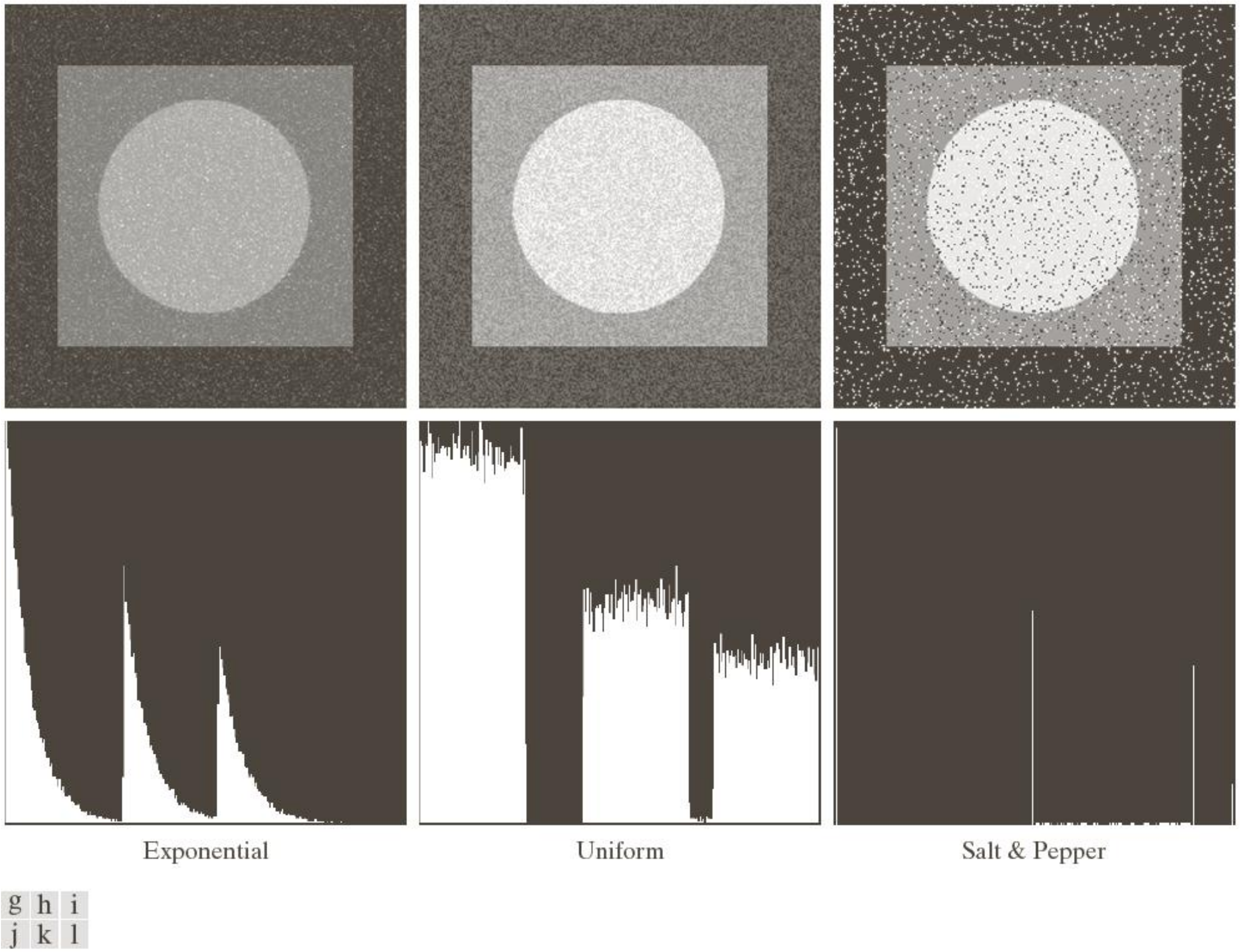


FIGURE 5.4 *(Continued)* Images and histograms resulting from adding exponential, uniform, and salt and pepper noise to the image in Fig. 5.3.

Restoration in the presence of noise only

- 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

It blurs the image.

Restoration in the presence of noise only (cont.)

- There are different kinds of mean filters all of which exhibit slightly different behaviour:
  - Geometric Mean
  - Harmonic Mean
  - Contraharmonic Mean

Restoration in the presence of noise only (cont.)

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.

Restoration in the presence of noise only (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.

Restoration in the presence of noise only (cont.)

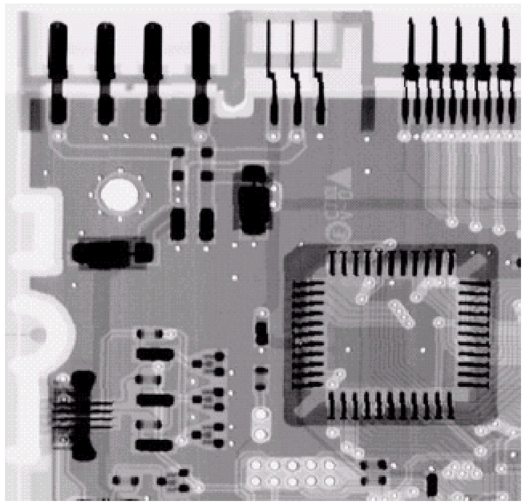
Contraharmonic Mean:

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

- Q is the order of the filter.
- Positive values of Q eliminate pepper noise.
- Negative values of Q eliminate salt noise.
- It cannot eliminate both simultaneously.

Noise Removal Examples

Original image



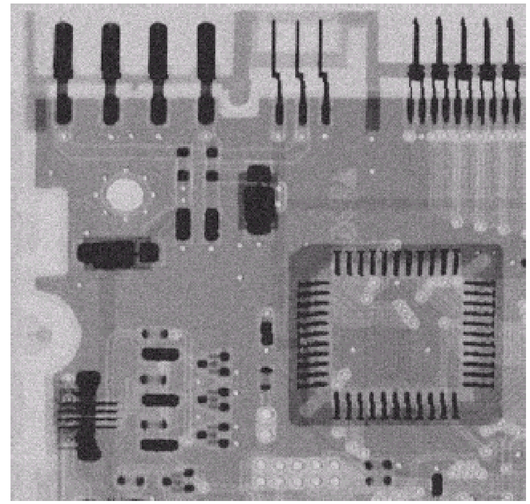
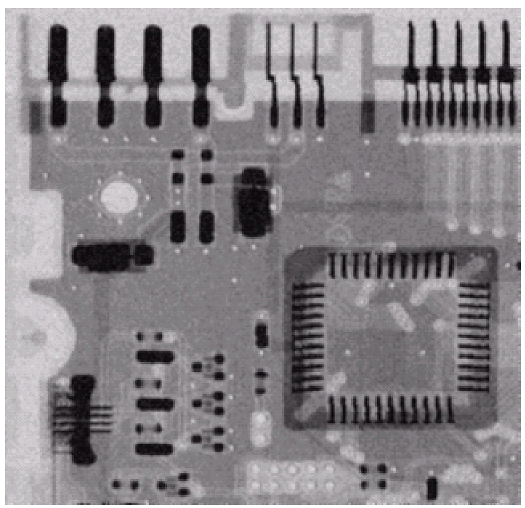
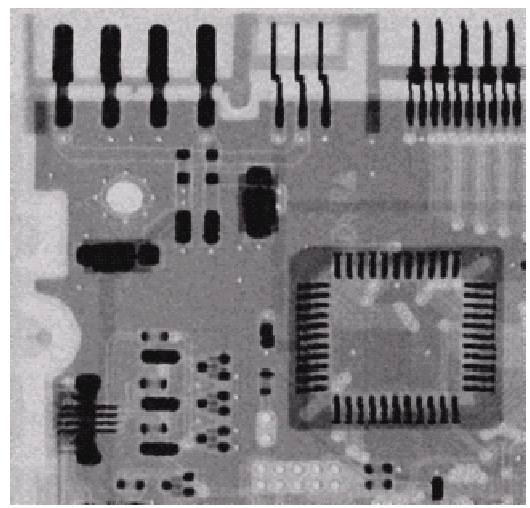


Image corrupted by Gaussian noise

3x3 Arithmetic Mean Filter

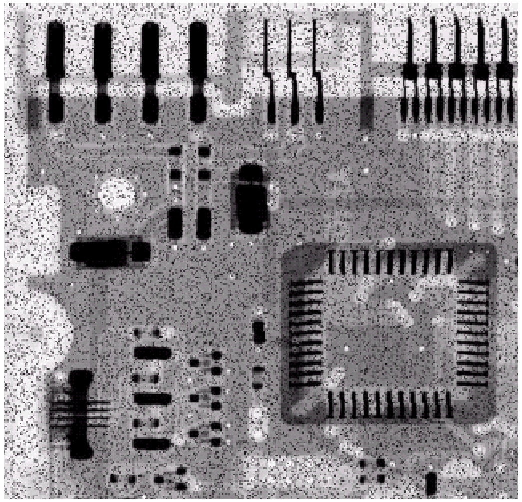




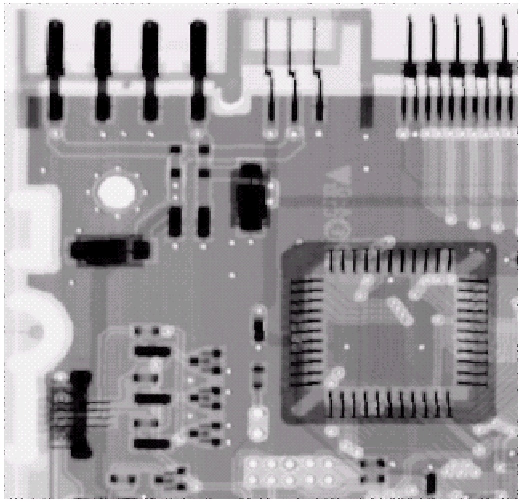
3x3 Geometric Mean Filter (less blurring than AMF, the image is sharper)

Noise Removal Examples (cont…)

Image corrupted by pepper noise at 0.1

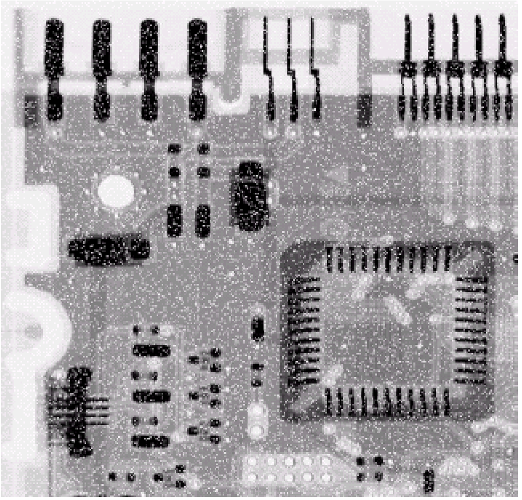


Filtering with a 3x3 Contraharmonic Filter with Q=1.5

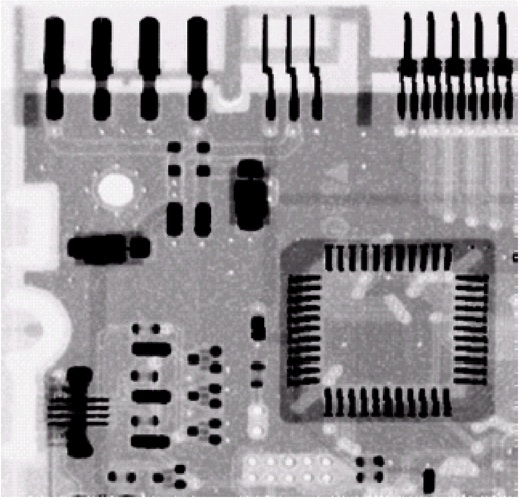


Noise Removal Examples (cont…)

Image corrupted by salt noise at 0.1

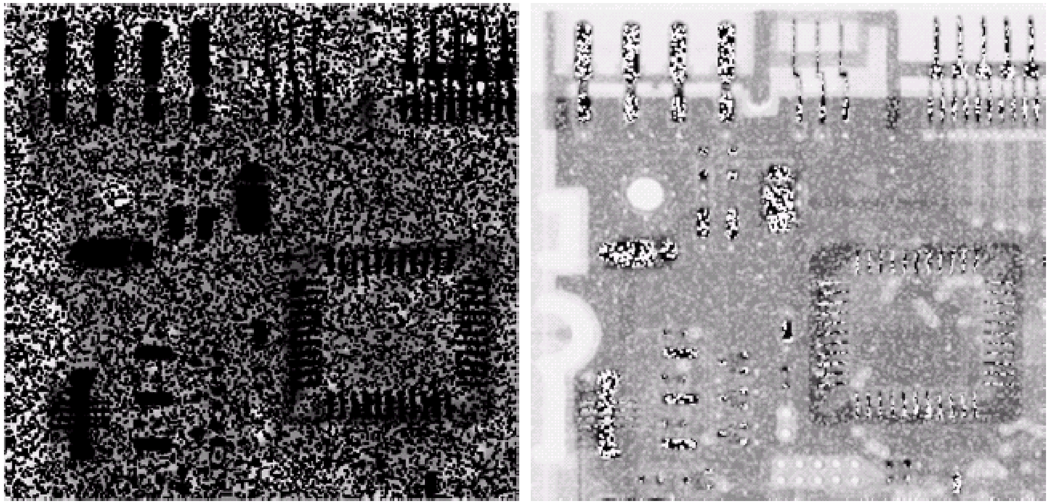


Filtering with a 3x3 Contraharmonic Filter with Q=-1.5



Contraharmonic Filter: Here Be Dragons

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



Pepper noise filtered by a 3x3 CF with Q=-1.5

Salt noise filtered by a 3x3 CF with Q=1.5

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

Order Statistics Filters

- Spatial filters based on ordering the pixel values that make up the neighbourhood defined by the filter support.

- Useful spatial filters include
  - Median filter
  - Max and min filter
  - Midpoint filter
  - Alpha trimmed mean filter

Median Filter:

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

- 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 filter 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:

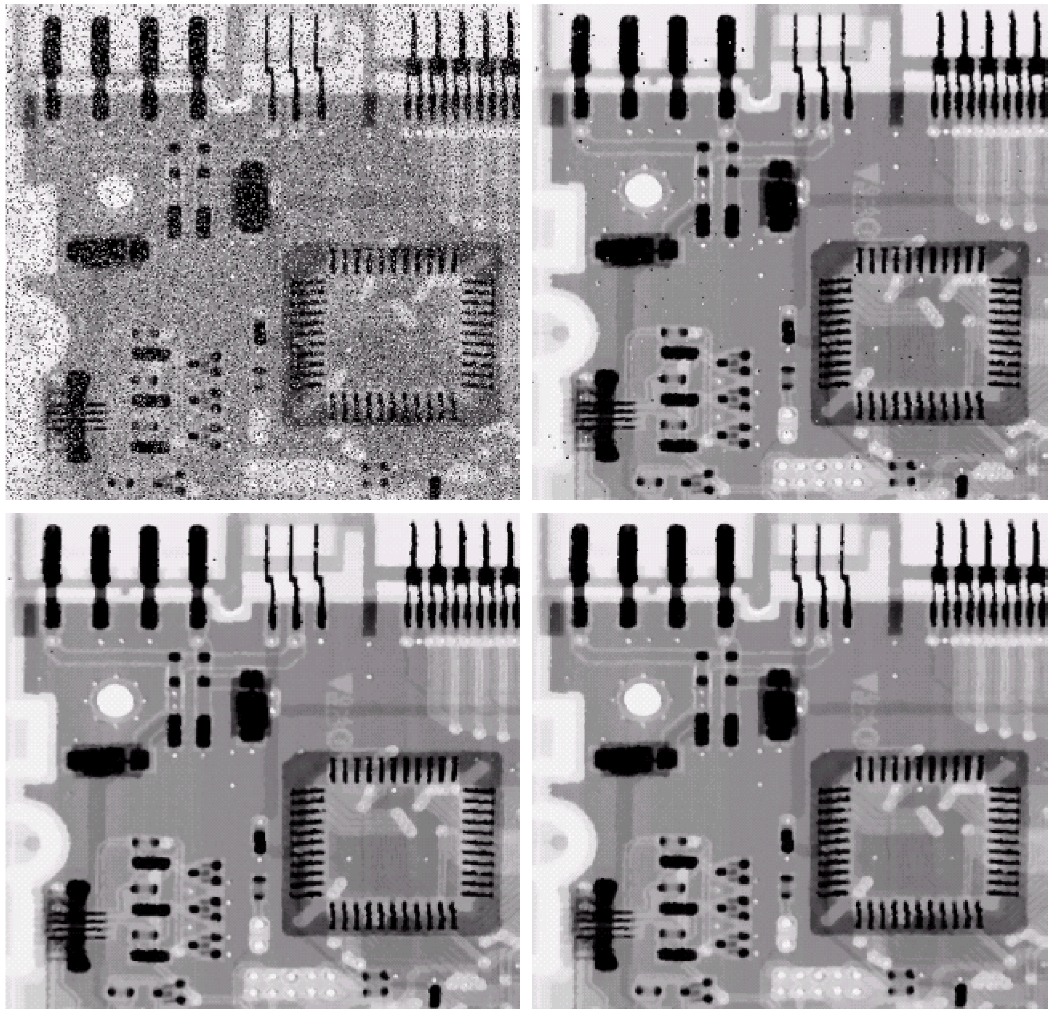
$$\hat{f}(x, y) = \frac{1}{mn - d} \sum_{(s,t) \in S_{xy}} g_r(s, t)$$

- 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

Salt And Pepper at 0.2

1 pass with a 3x3 median



2 passes with a 3x3 median

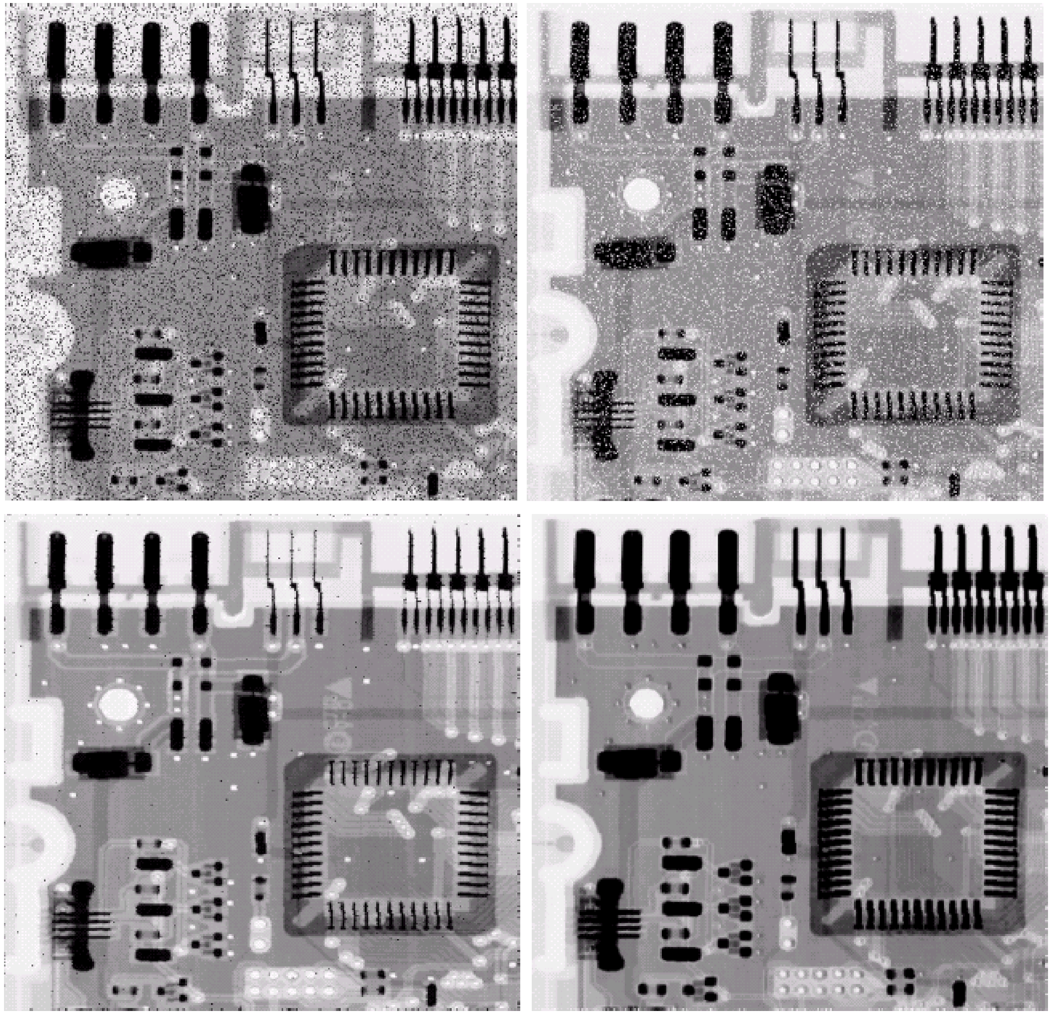
3 passes with a 3x3 median

Repeated passes remove the noise better but also blur the image

Noise Removal Examples (cont…)

Image corrupted by Pepper noise

Image corrupted by Salt noise

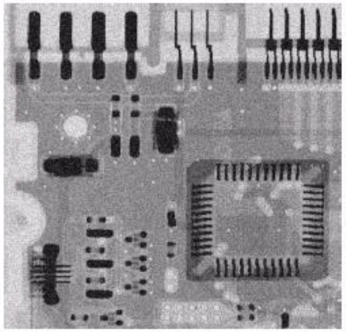


Filtering above with a 3x3 Max Filter

Filtering above with a 3x3 Min Filter

Noise Removal Examples (cont…)

Image corrupted by uniform noise



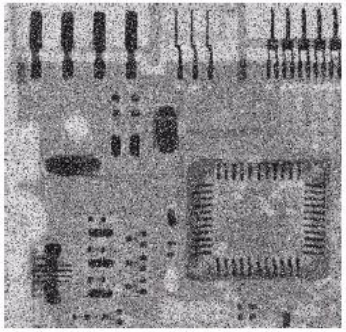
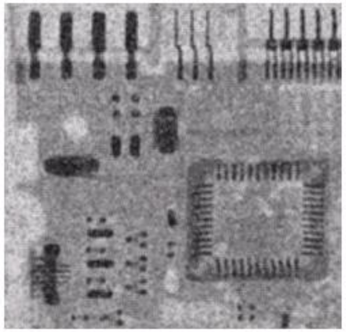
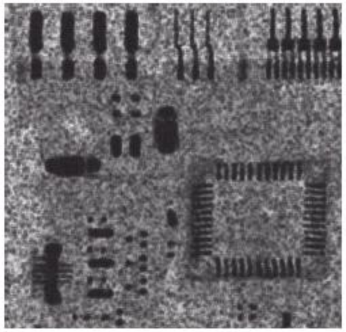


Image further corrupted by Salt and Pepper noise

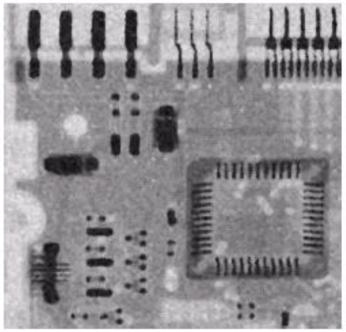
Filtering by a 5x5 Arithmetic Mean Filter

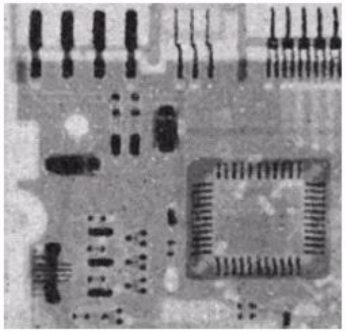




Filtering by a 5x5 Geometric Mean Filter

Filtering by a 5x5 Median Filter





Filtering by a 5x5 Alpha-Trimmed Mean Filter (d=5)

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.

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 (excessive thinning or thickenning of object boundaries).

Adaptive Median Filtering (cont…)

- In the adaptive median filter, the filter size changes depending on the characteristics of the image.
- Notation:
  - S_{xy} = the support of the filter centerd at (x, y)
  - 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…)

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

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

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

Else increase the window size

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

Else output z_{med}

Stage 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…)

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

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

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

Else increase the window size

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

Else output z_{med}

- Stage A determines if the output of the median filter z_{med} is an impulse or not (black or white).
- If it is not an impulse, we go to stage B.
- If it is an impulse the window size is increased until it reaches S_{max} or z_{med} is not an impulse.
- Note that there is no guarantee that z_{med} will not be an impulse. The smaller the the density of the noise is, and, the larger the support S_{max}, we expect not to have an impulse.

Adaptive Median Filtering (cont…)

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

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

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

Else output z_{med}

Stage B determines if the pixel value at (x, y), that is z_{xy}, is an impulse or not (black or white).

If it is not an impulse, the algorithm outputs the unchanged pixel value z_{xy} .

If it is an impulse the algorithm outputs the median z_{med} .

Adaptive Filtering Example

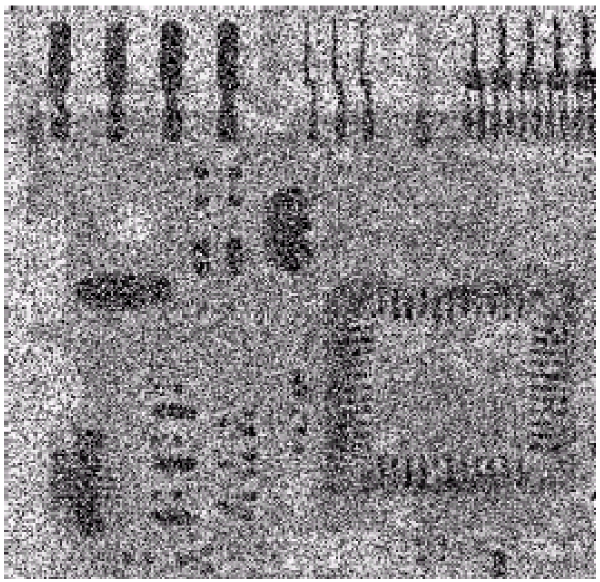
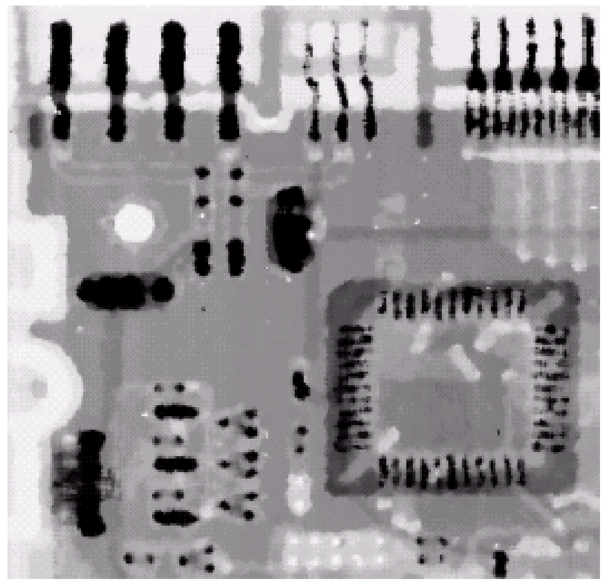
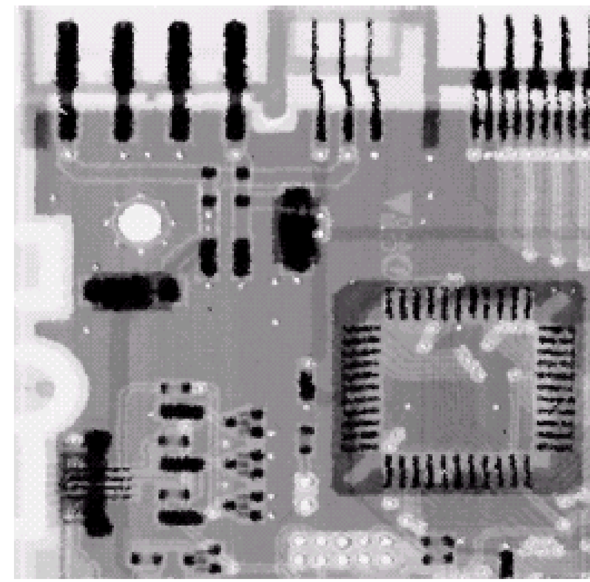


Image corrupted by salt and pepper noise with probabilities $P_a = P_b = 0.25$



Result of filtering with a 7x7 median filter

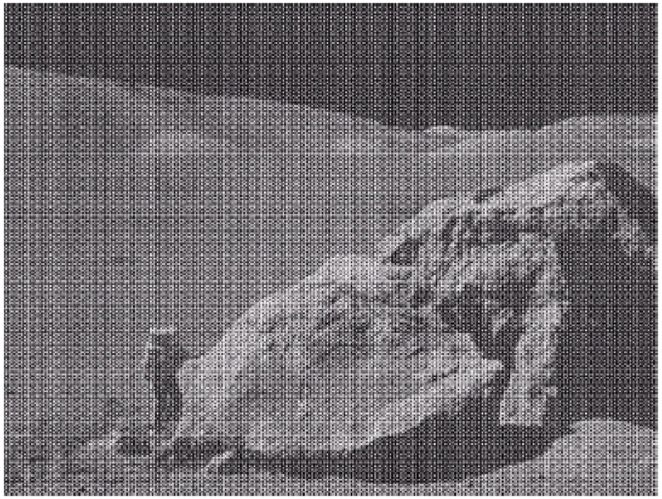


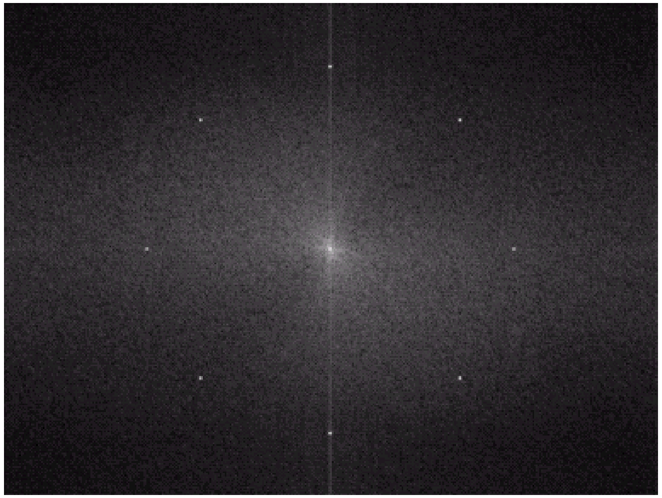
Result of adaptive median filtering with $S_{max} = 7$

AMF preserves sharpness and details, e.g. the connector fingers.

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.



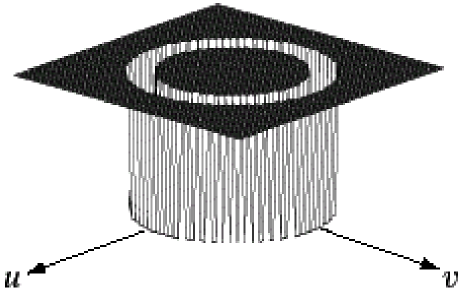


Band Reject Filters

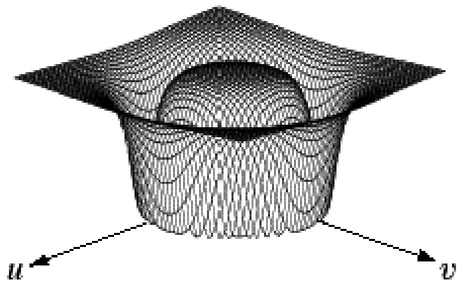
- Removing periodic noise form an image involves removing a particular range of frequencies from that image.
- *Band reject* filters can be used for this purpose
- An ideal band reject filter is given as follows:

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

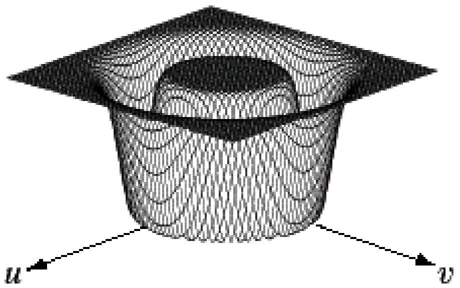
Band Reject Filters (cont…)



Ideal Band Reject Filter



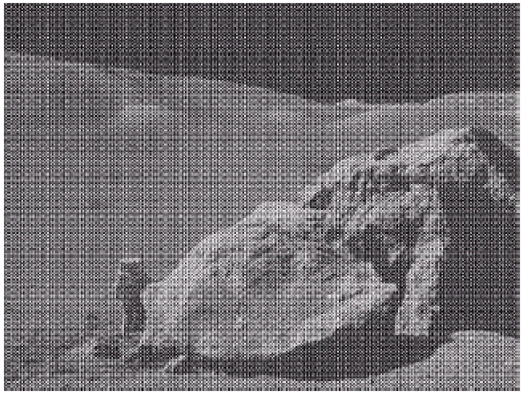
Butterworth Band Reject Filter (of order 1)



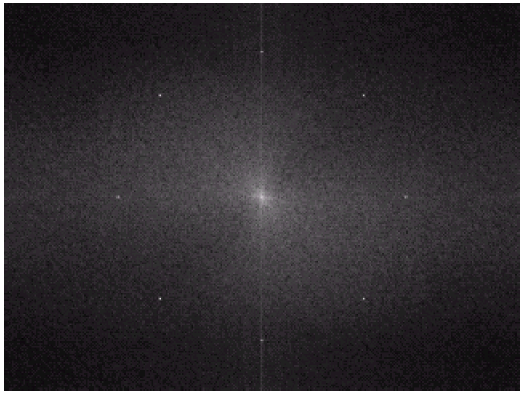
Gaussian Band Reject Filter

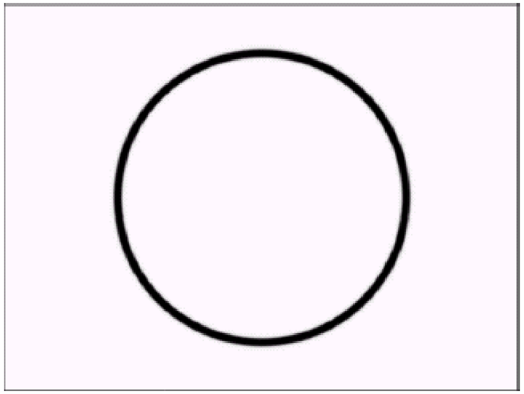
Band Reject Filter Example

Image corrupted by sinusoidal noise



Fourier spectrum of corrupted image





Butterworth band reject filter

Filtered image

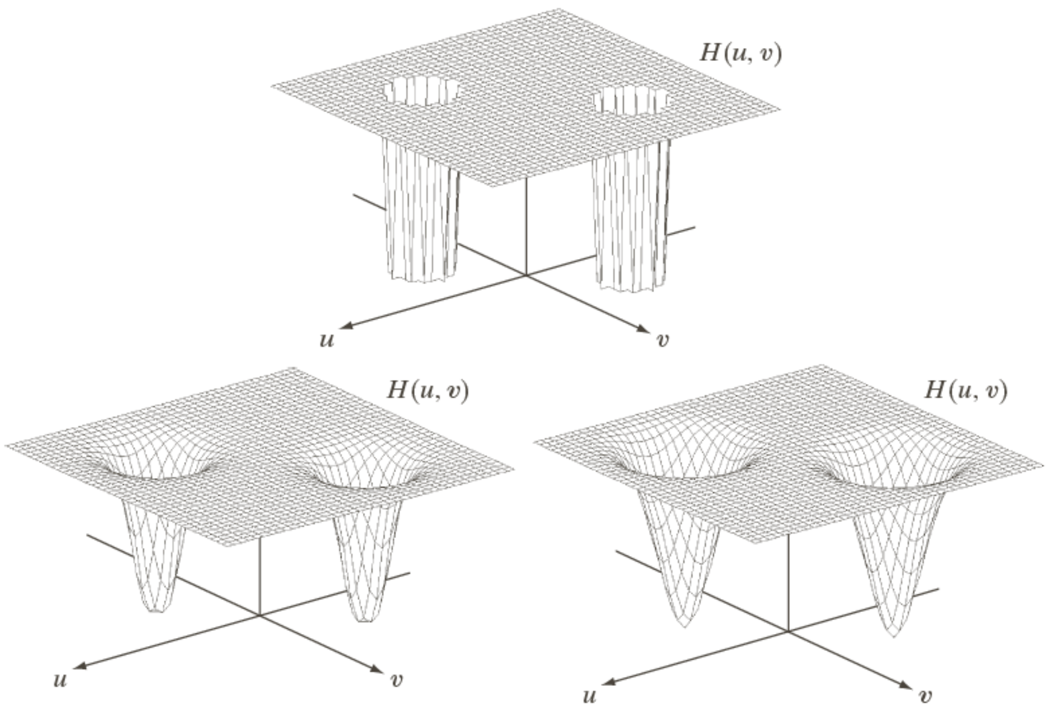
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Notch Filters

- Rejects frequencies in a predefined neighbourhood around a center frequency.

a
b c

FIGURE 5.18 Perspective plots of (a) ideal, (b) Butterworth (of order 2), and (c) Gaussian notch (reject) filters.



Optimum Notch Filtering

- Several interference components (not a single burst).
- Removing completely the star-like components may also remove image information.

a b

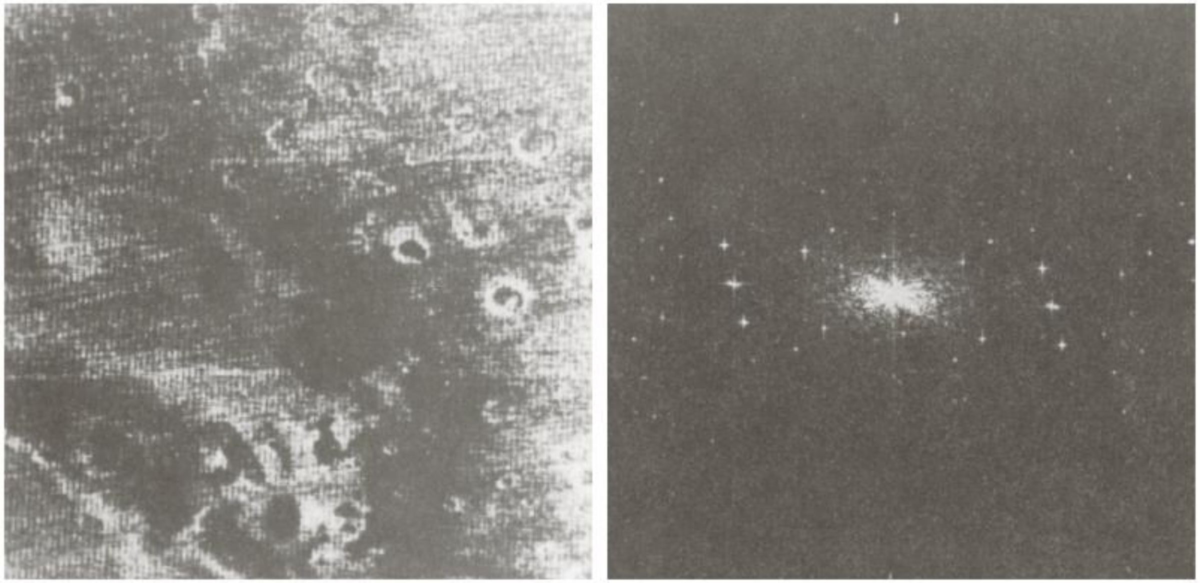
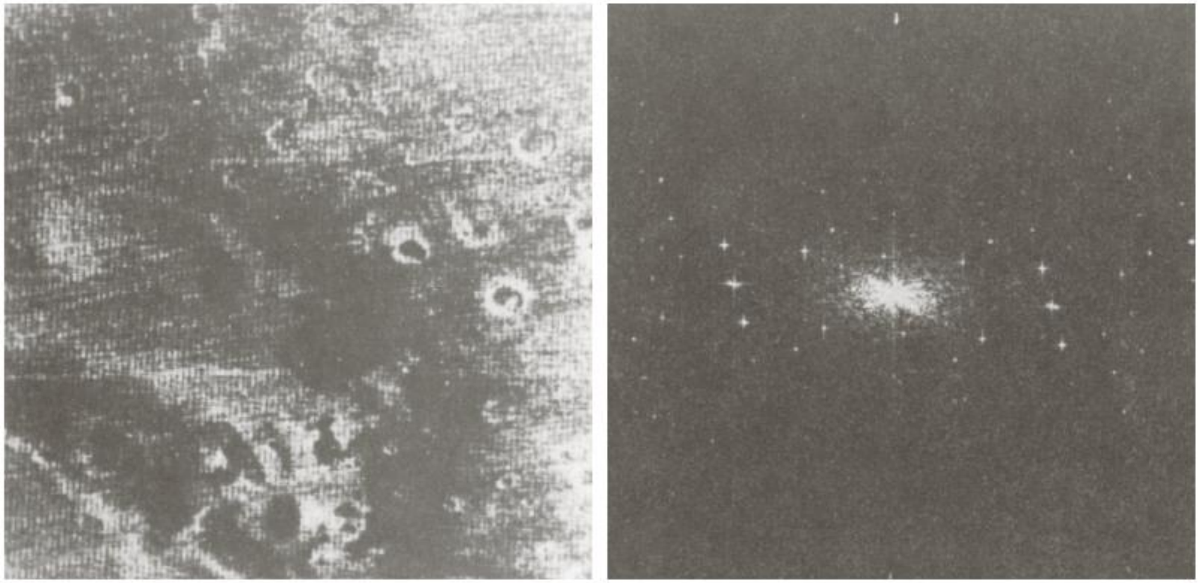


FIGURE 5.20 (a) Image of the Martian terrain taken by *Mariner 6*. (b) Fourier spectrum showing periodic interference. (Courtesy of NASA.)

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

Optimum Notch Filtering (cont.)

- Apply the notch filter to isolate the bursts.
- Remove a portion of the burst.



a b

FIGURE 5.20 (a) Image of the Martian terrain taken by *Mariner 6*. (b) Fourier spectrum showing periodic interference. (Courtesy of NASA.)

Optimum Notch Filtering (cont.)

- A noise estimate in the DFT domain:

$$N(k,l) = H(k,l)G(k,l)$$

- In the spatial domain:

$$\eta(m,n) = \mathfrak{I}^{-1}\left\{H(k,l)G(k,l)\right\}$$

- Image estimate:

$$\hat{f}(m,n) = g(m,n) - w(m,n)\eta(m,n)$$

Optimum Notch Filtering (cont.)

$$\hat{f}(m,n) = g(m,n) - w(m,n)\eta(m,n)$$

- Compute the weight minimizing the variance over a local neighbourhood of the estimated image centered at (m,n):

$$\sigma(m,n) = \frac{1}{(2a+1)(2b+1)} \sum_{k=-a}^{a} \sum_{l=-b}^{b} \left[\hat{f}(m+k,n+l) - \overline{\hat{f}}(m,n) \right]^2$$

with $$\overline{\hat{f}}(m,n) = \frac{1}{(2a+1)(2b+1)} \sum_{k=-a}^{a} \sum_{l=-b}^{b} \hat{f}(m+k,n+l)$$

- Substituting the estimate in $\sigma(m,n)$: yields:

Optimum Notch Filtering (cont.)

$$\sigma(m,n) = \frac{1}{(2a+1)(2b+1)} \sum_{k=-a}^{a} \sum_{l=-b}^{b} \Big\{ \big[g(m+k,n+l) - w(m+k,n+l)\eta(m+k,n+l) \big] - \big[\bar{g}(m,n) - \overline{w(m,n)\eta(m,n)} \big] \Big\}^2$$

- A simplification is to assume that the weight remains constant over the neighbourhood:

$$w(m+k,n+l) = w(m,n), \; -a \le k \le a, -b \le l \le b$$

$$\sigma(m,n) = \frac{1}{(2a+1)(2b+1)} \sum_{k=-a}^{a} \sum_{l=-b}^{b} \Big\{ \big[g(m+k,n+l) - w(m,n)\eta(m+k,n+l) \big] - \big[\bar{g}(m,n) - w(m,n)\bar{\eta}(m,n) \big] \Big\}^2$$

Optimum Notch Filtering (cont.)

- To minimize the variance:

$$\frac{\partial \sigma(m,n)}{\partial w(m,n)} = 0$$

yielding the closed-form solution:

$$w(m,n) = \frac{\overline{g(m,n)\eta(m,n)} - \bar{g}(m,n)\bar{\eta}(m,n)}{\overline{\eta^2}(m,n) - \bar{\eta}^2(m,n)}$$

- More elaborated result is obtained for non-constant weight $w(m,n)$ at each pixel.