Digital Image Processing

Image Restoration and Reconstruction (Noise Removal)

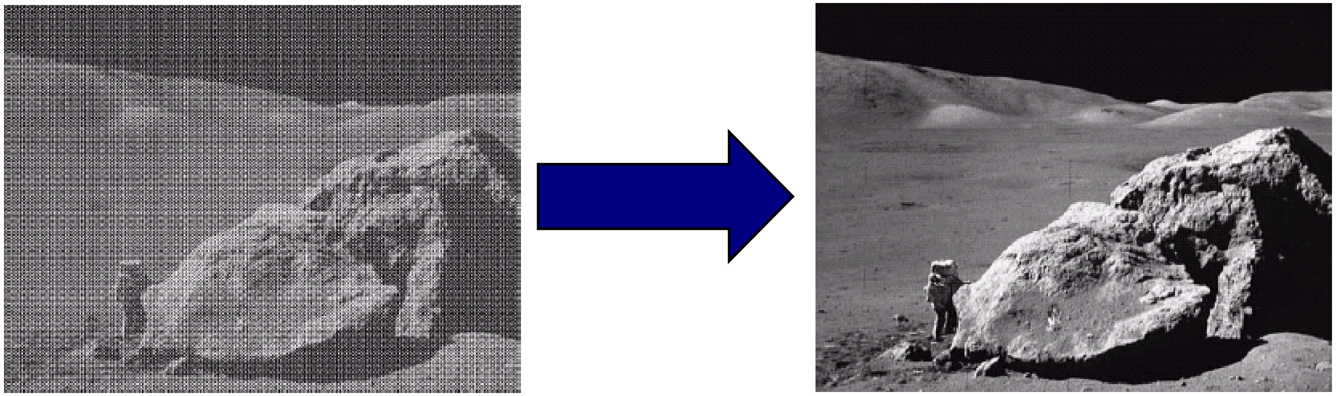
Topics to cover

- 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

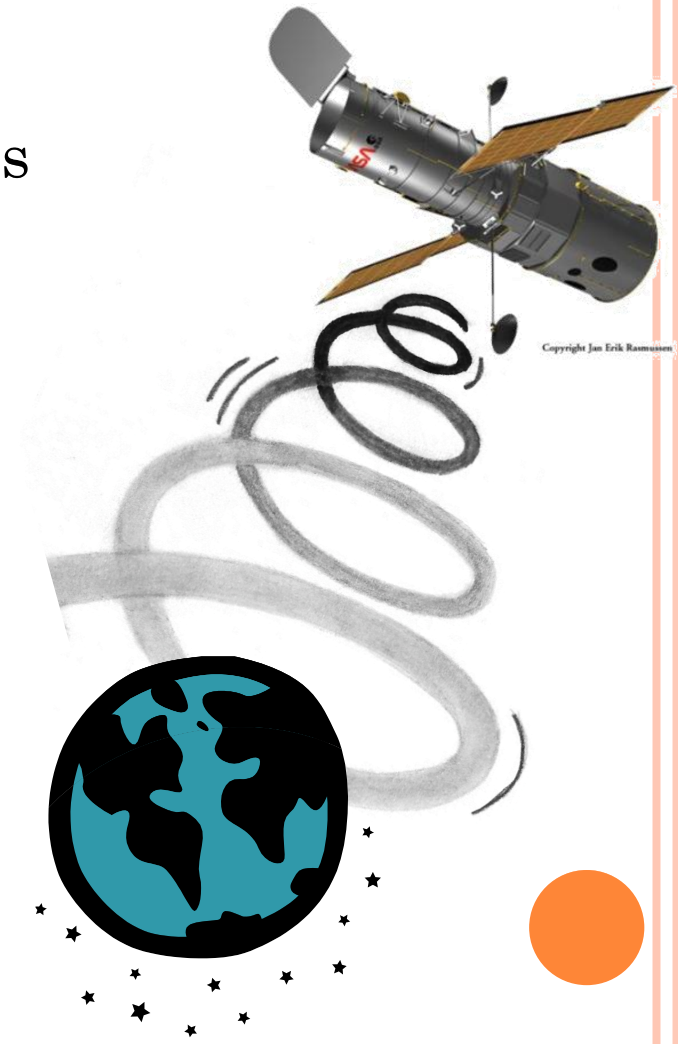


IMAGE BLURRED BY ATMOSPHERIC TURBULENCE & WITH ADDITIVE NOISE

a b
c d

FIGURE 5.25 Illustration of the atmospheric turbulence model. (a) Negligible turbulence. (b) Severe turbulence, $k = 0.0025$. (c) Mild turbulence, $k = 0.001$. (d) Low turbulence, $k = 0.00025$. (Original image courtesy of NASA.)

$$H(u,v) = e^{-c(u^2+v^2)^{\frac{5}{6}}}$$

Inverse v/s Pseudo-inverse filtering

a b
c d

FIGURE 5.27 Restoring Fig. 5.25(b) with Eq. (5.7-1). (a) Result of using the full filter. (b) Result with *H* cut off outside a radius of 40; (c) outside a radius of 70; and (d) outside a radius of 85.

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

Block Diagram

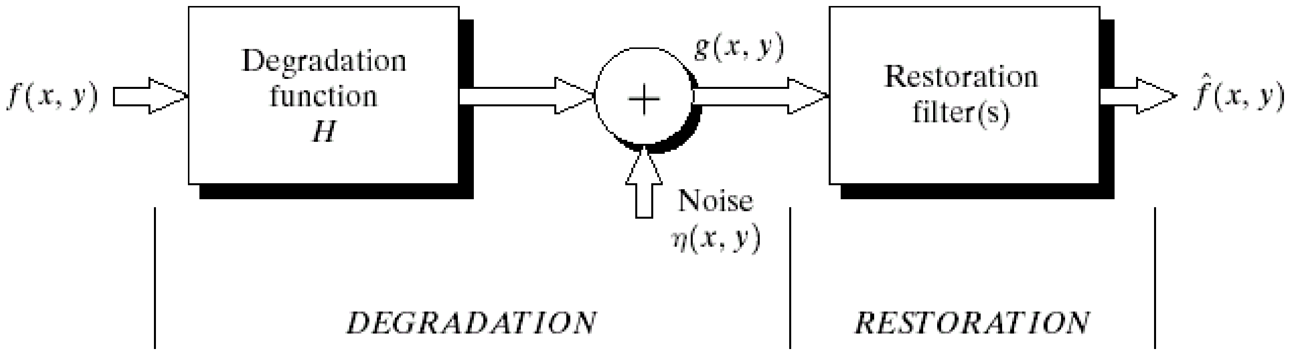
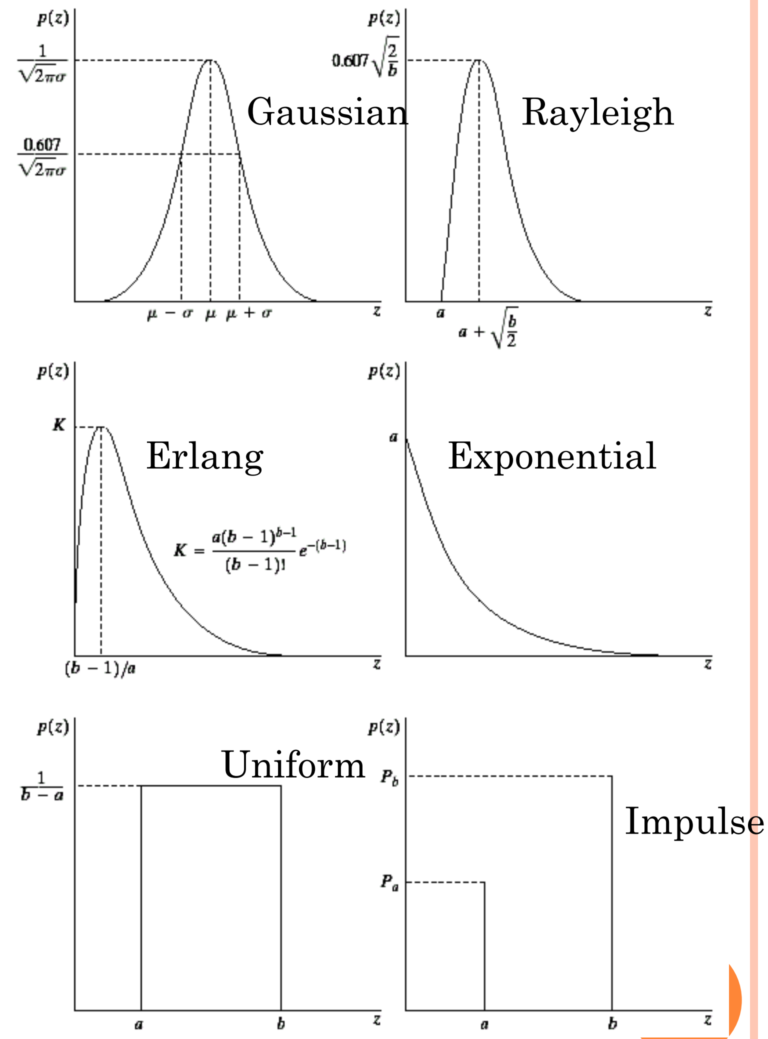


FIGURE 5.1 A model of the image degradation/restoration process.

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

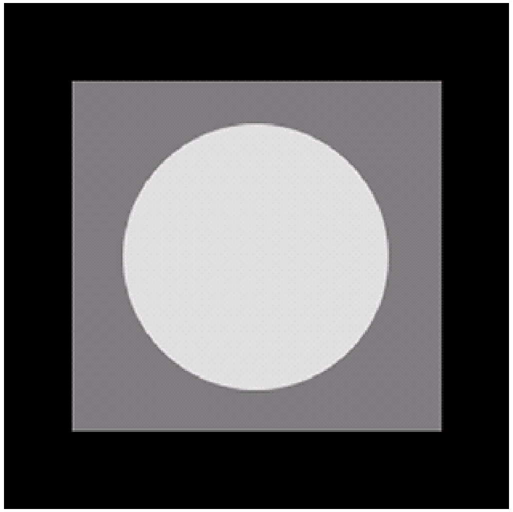


Images taken from Gonzalez & Woods, Digital Image Processing (2002)

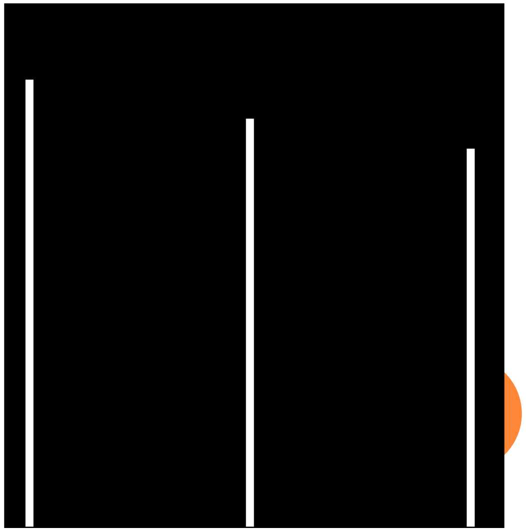
NOISE EXAMPLE

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

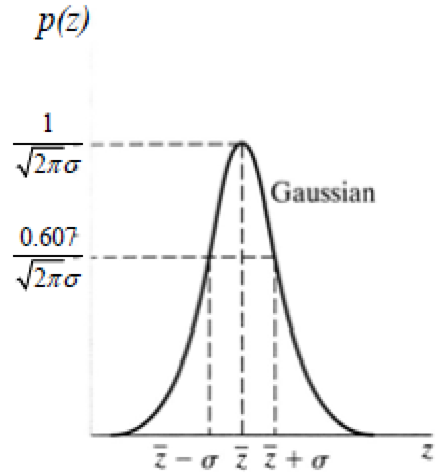
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Noise pdfs

1. **Gaussian** (normal) noise is very attractive from a mathematical point of view since its DFT is another Gaussian process.
   pdf of Gaussian r.v. is

$$p(z) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(z-\bar{z})^2}{2\sigma^2}}$$

Here z represents intensity, $\check{z}$ is the mean (average) value of z, and σ is its standard deviation. σ^2 is the variance of z.
Electronic circuit noise, sensor noise due to low illumination or high temperature.



Noise pdfs

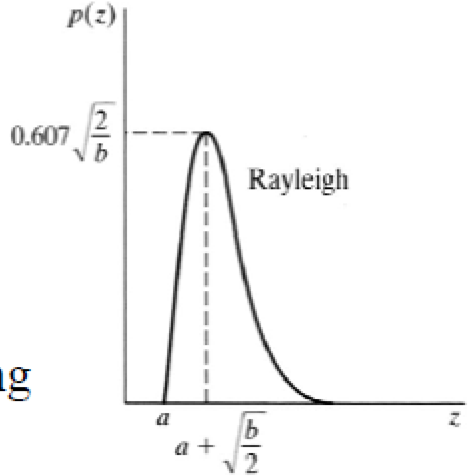
2. Rayleigh noise is specified as

$$p(z) = \begin{cases} \frac{2}{b}(z-a)e^{-\frac{(z-a)^2}{b}} & z \geq a \\ 0 & z < a \end{cases}$$

The mean and variance are given by

$$\bar{z} = a + \sqrt{\pi b/4}$$

$$\sigma^2 = \frac{b(4-\pi)}{4}$$

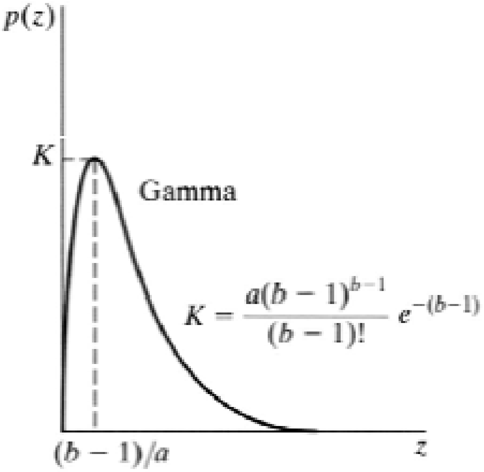


The Rayleigh density is useful for approximating skewed histograms. Used in range imaging.

Noise pdfs

3. Erlang noise is specified as

$$p(z)=\begin{cases}\dfrac{a^b z^{b-1}}{(b-1)!}e^{-az} & z\geq 0\\ 0 & z<0\end{cases}$$



Here $a>0$ and b is a positive integer. The mean and variance are given by

$$\bar{z}=b/a$$

$$\sigma^2=b/a^2$$

When the denominator is the gamma function, the pdf describes the gamma distribution.

Noise pdfs

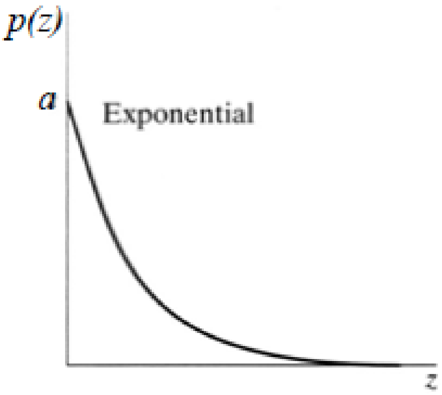
4. Exponential noise is specified as

$$p(z) = \begin{cases} ae^{-az} & z \geq 0 \\ 0 & z < 0 \end{cases}$$

Here $a > 0$. The mean and variance are given by

$$\bar{z} = 1/a$$

$$\sigma^2 = 1/a^2$$

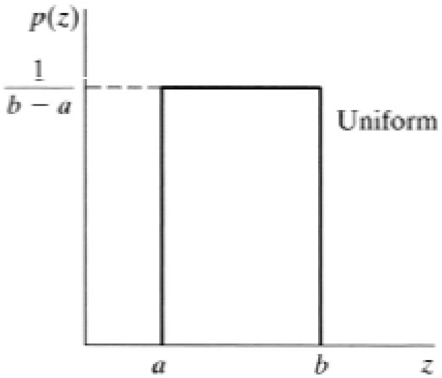


Exponential pdf is a special case of Erlang pdf with b =1.
Used in laser imaging.

Noise pdfs

5. Uniform noise is specified as

$$p(z)=\begin{cases}\dfrac{1}{b-a} & a\le z\le b\\[4pt] 0 & otherwise\end{cases}$$



The mean and variance are given by

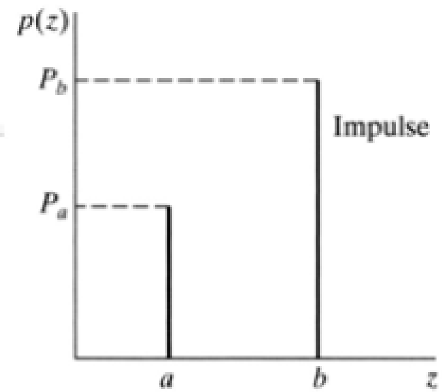
$$\bar{z}=\frac{a+b}{2}$$

$$\sigma^2=\frac{(b-a)^2}{12}$$

Noise pdfs

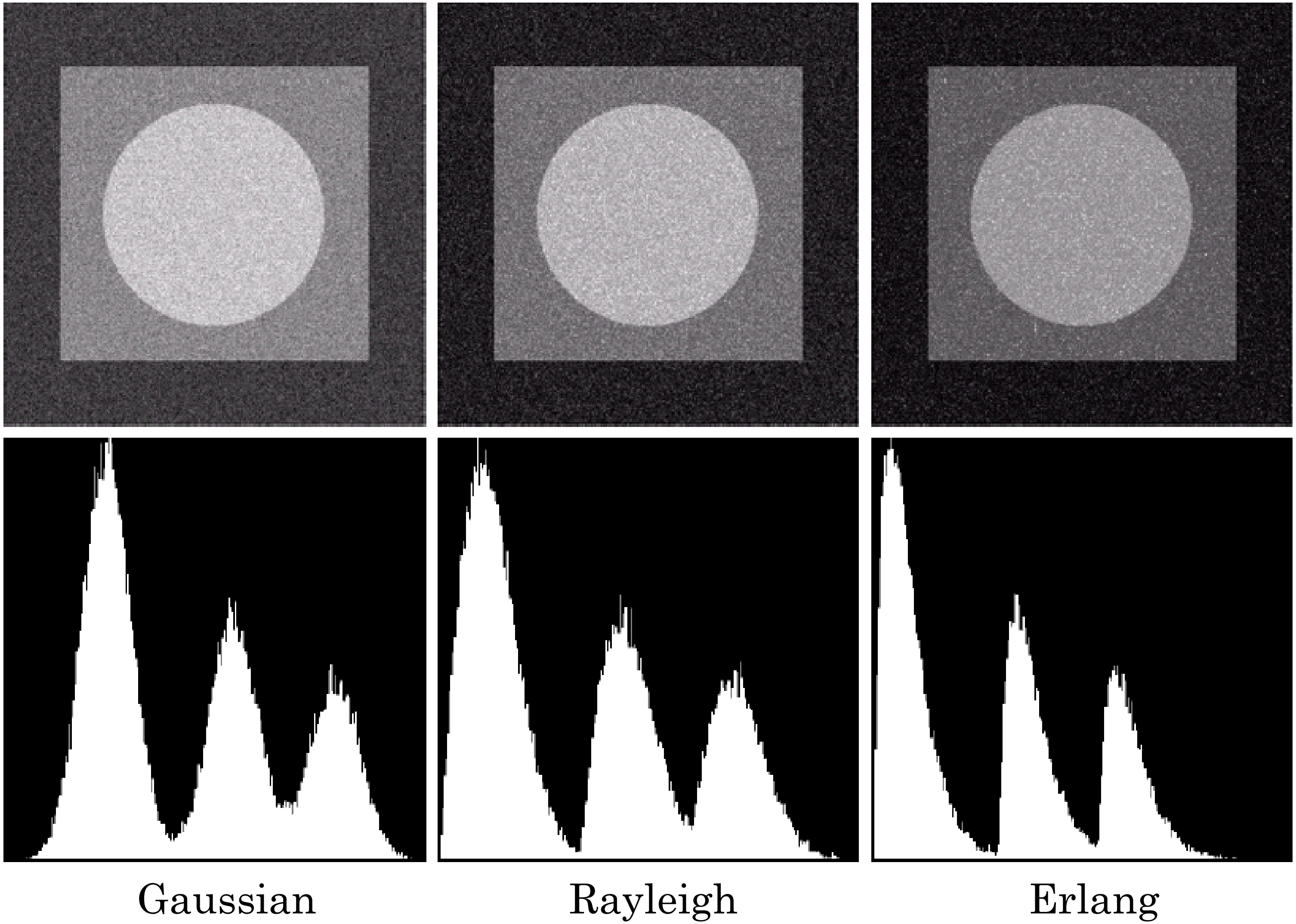
6. Impulse (salt-and-pepper) noise (bipolar) is specified as

$$p(z) = \begin{cases} P_a & z = a \\ P_b & z = b \\ 0 & \text{otherwise} \end{cases}$$



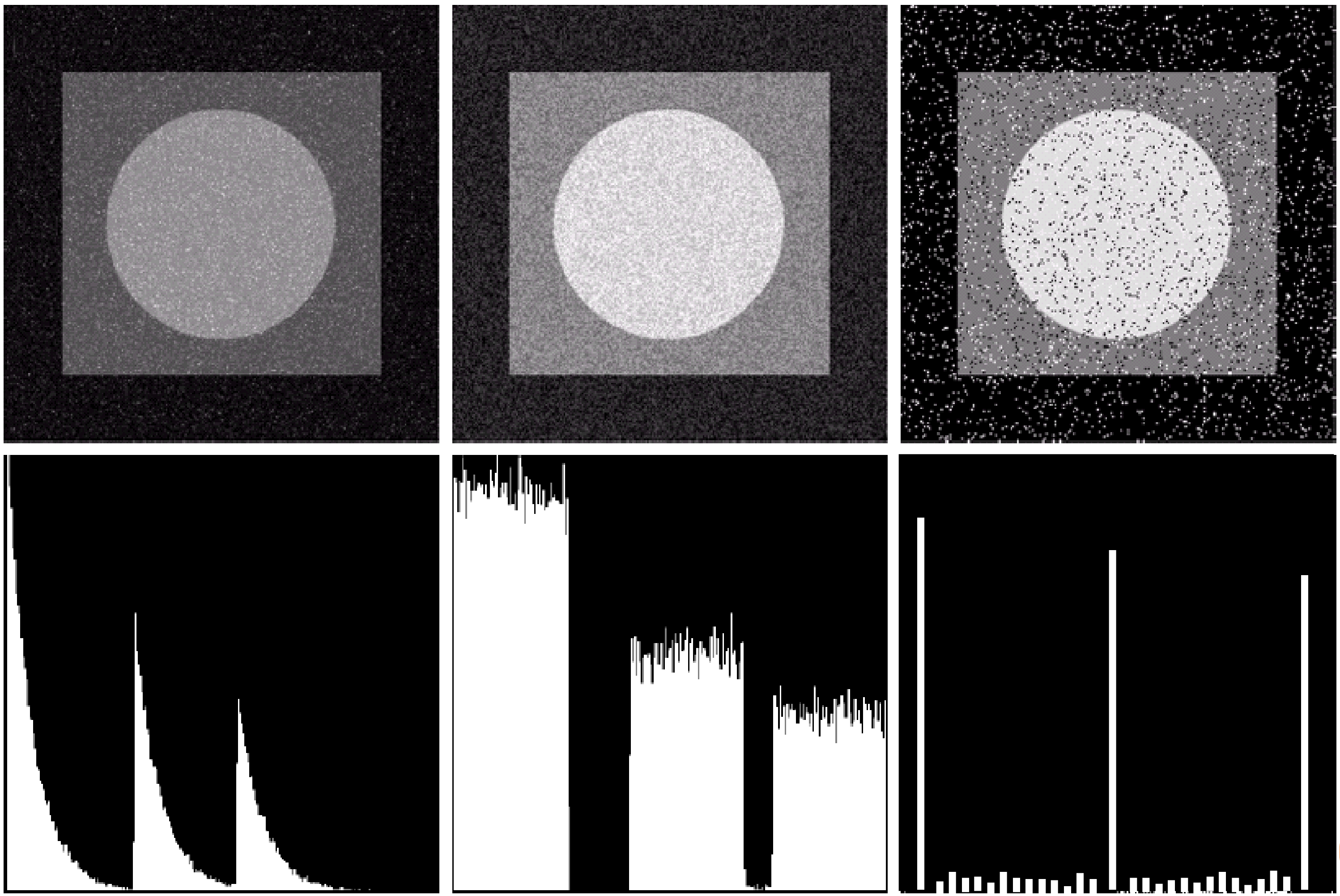
If $b>a$, intensity b will appear as a light dot on the image and a appears as a dark dot. If either P_a or P_b is zero, the noise is called *unipolar*. Frequently, a and b are ***saturated*** values, resulting in positive impulses being white and negative impulses being black. This noise shows up when quick transitions – such as faulty switching – take place.

NOISE EXAMPLE (CONT...)



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

Noise Example (cont…)



Exponential Uniform Impulse

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

Estimation of noise parameters

The simplest use of the data from the image strips is to calculate the mean and variance of intensity levels:

$$\bar{z} = \sum_{i=0}^{L-1} z_i p_s(z_i)$$

$$\sigma^2 = \sum_{i=0}^{L-1} (z_i - \bar{z}) p_s(z_i)$$

If the shape of pdf is Gaussian, the mean and variance completely specify it. Otherwise, these estimates are needed to derive a and b.

For impulse noise, we need to estimate actual probabilities P_a and P_b from the histogram.

Restoration in the presence of noise only – Spatial filtering

When the only degradation in the image is noise:

$$g(x, y) = f(x, y) + \eta(x, y)$$
$$G(u, v) = F(u, v) + N(u, v)$$

the noise terms are unknown, so, subtracting them from $g(x,y)$ or $G(u,v)$ is not a realistic option. Therefore, spatial filtering is a good candidate when only additive random noise is present.

Restoration using Spatial Filtering

- Mean Filters
- Order Statistic Filters
- Adaptive Filters

Restoration using Spatial Filtering

Mean Filters:

- Arithmetic mean filter
- Geometric Mean
- Harmonic Mean
- Contraharmonic Mean

Arithmetic Mean filter

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 by introducing the blur in the image.

Other Means (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.

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.

Positive values of Q eliminate pepper noise.

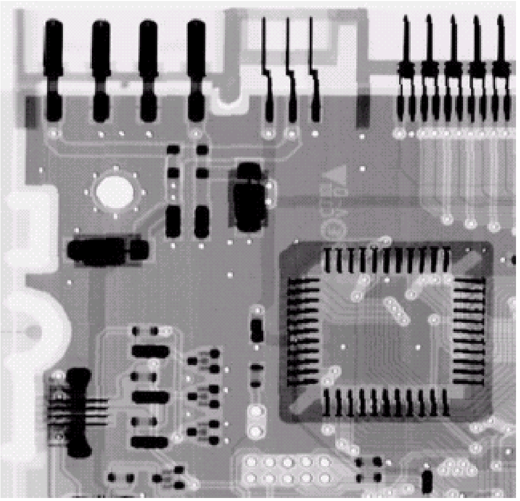
Negative values of Q eliminate salt noise.

It cannot eliminate both simultaneously.

Noise Removal Examples

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

Original image



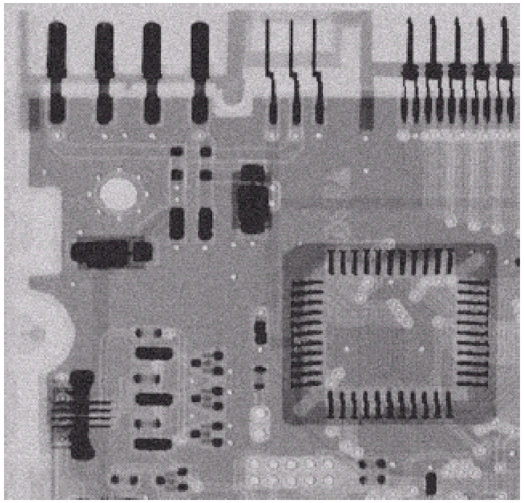
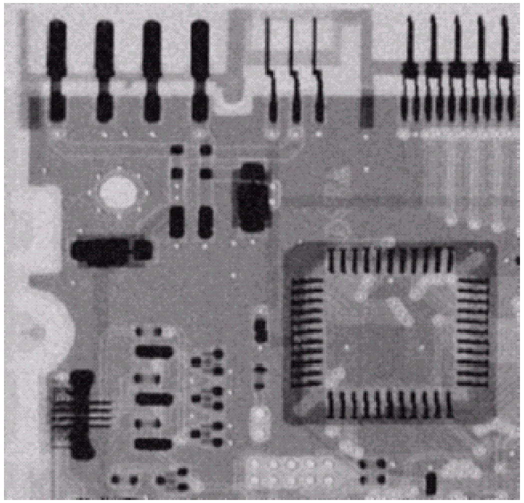
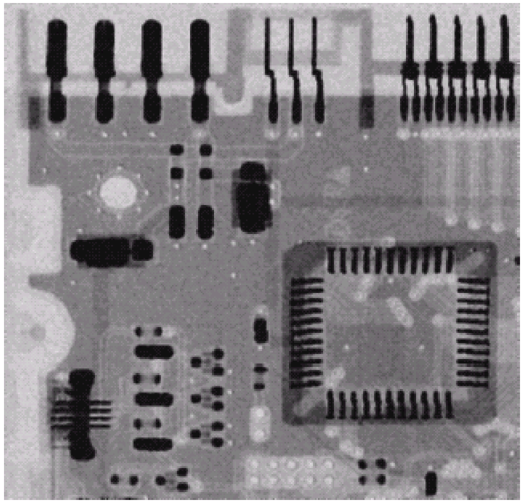


Image corrupted by Gaussian noise

3x3 Arithmetic Mean Filter



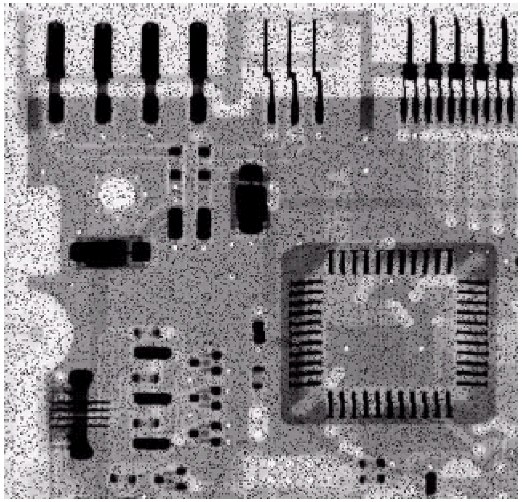


3x3 Geometric Mean Filter (less blurring than AMF, the image is sharper)

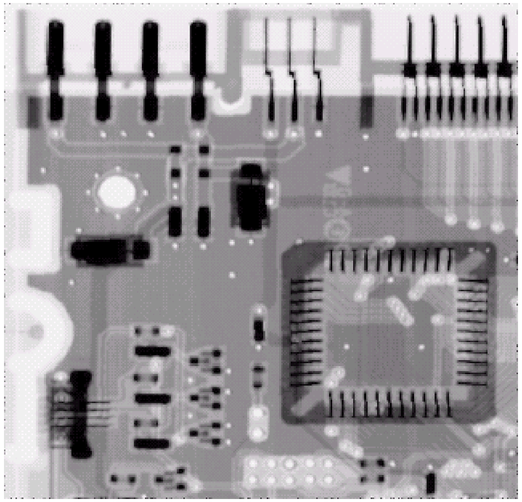
Noise Removal Examples (cont…)

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

Image corrupted by pepper noise at 0.1

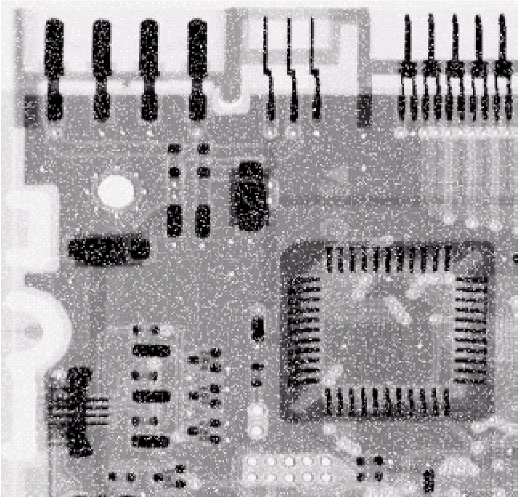


Filtering with a 3x3 Contraharmonic Filter with Q=1.5

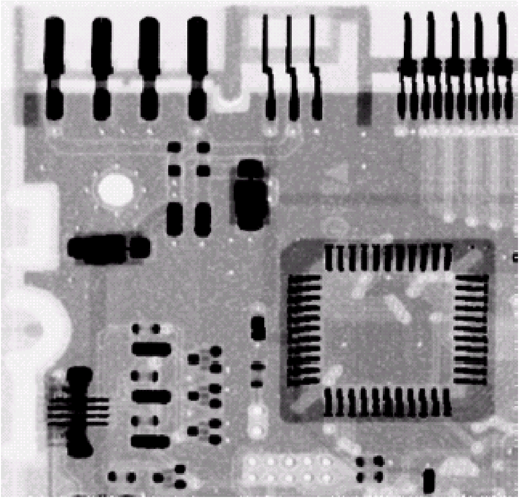


Noise Removal Examples (cont…)

Image corrupted by salt noise at 0.1



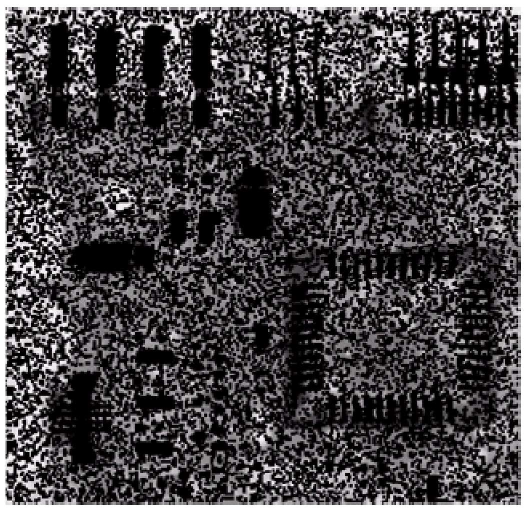
Filtering with a 3x3 Contraharmonic Filter with Q=-1.5



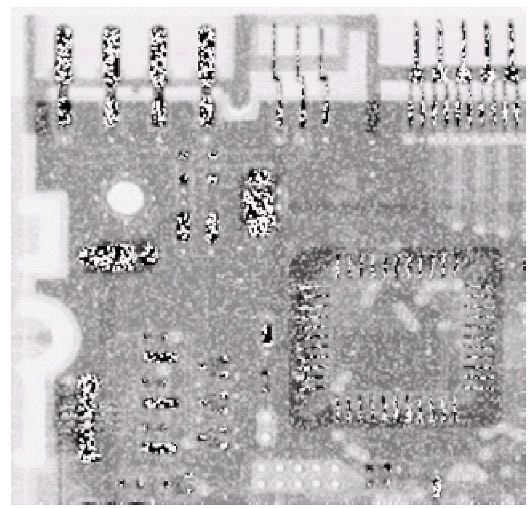
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

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

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

a) **Median filter**: replaces the pixel value by the median of the intensity levels in the neighborhood of that pixel:

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

Median filters provide excellent results for certain types of noise with considerably less blurring than linear smoothing filters of the same size. These filters are very effective against both bipolar and unipolar noise contaminations (if $P_a < 0.2$ and $P_b < 0.2$). The same filter can be applied more than once to yield better results.

Noise Removal Examples

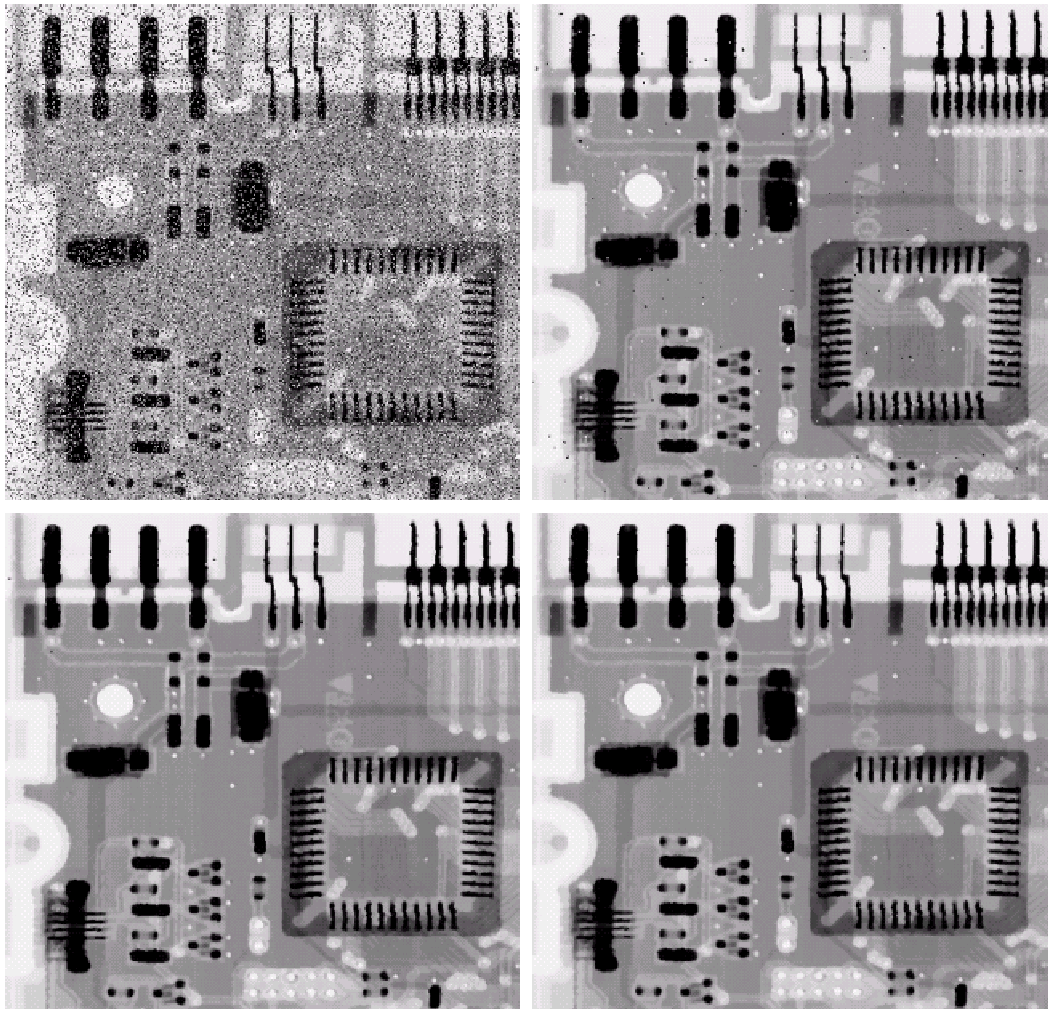


Image corrupted by Salt And Pepper noise at 0.2

Result of 1 pass with a 3x3 Median Filter

Result of 2 passes with a 3x3 Median Filter

Result of 3 passes with a 3*3 Median Filter

Repeated passes remove the noise better but also blur the image

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

b) **Max and Min filters**:

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

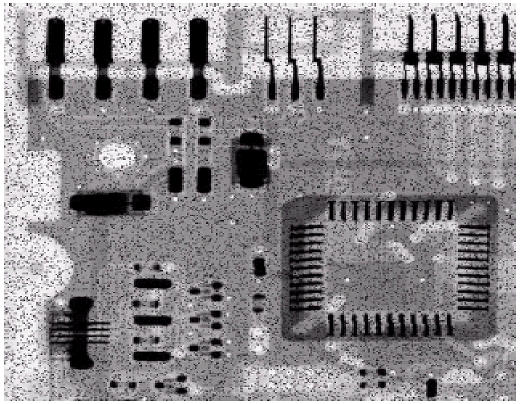
This filter is useful for finding the brightest points in an image; therefore, its effective against pepper noise.

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

This filter is useful for finding the darkest points in an image; therefore, its effective against salt noise.

Noise Removal Examples (cont…)

Image corrupted by Pepper noise



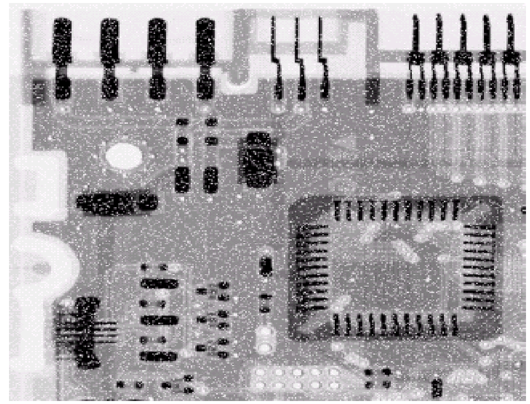
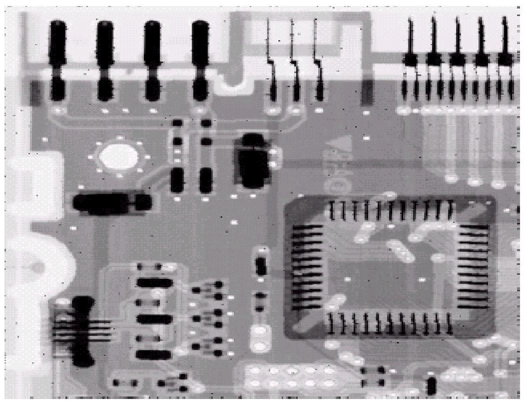
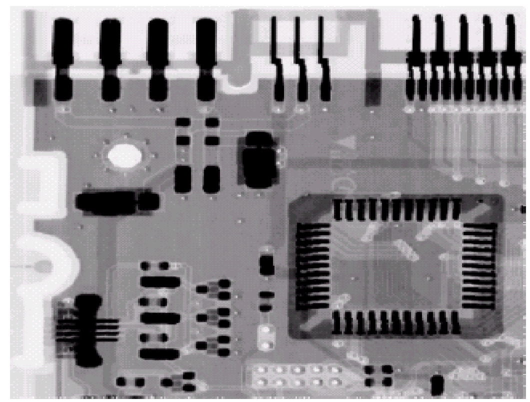


Image corrupted by Salt noise

Filtering above with a 3x3 Max Filter





Filtering above with a 3x3 Min Filter

1. Max filter will reduce some black pixels which are near to dark image object.
2. Min Filter will make white pixel also to dark pixel near the dark image object.

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

c) **Midpoint filter**: computes the midpoint between the maximum and minimum values of intensities:

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

This filter is a combination of order statistics and averaging and works best for Gaussian and uniform noise contaminations.

d) **Alpha-trimmed mean filter**: if we delete $d/2$ highest intensity values and $d/2$ lowest intensity values, denote the rest as $g_r(x,y)$; a filter that averages what is left is alpha-trimmed mean filter:

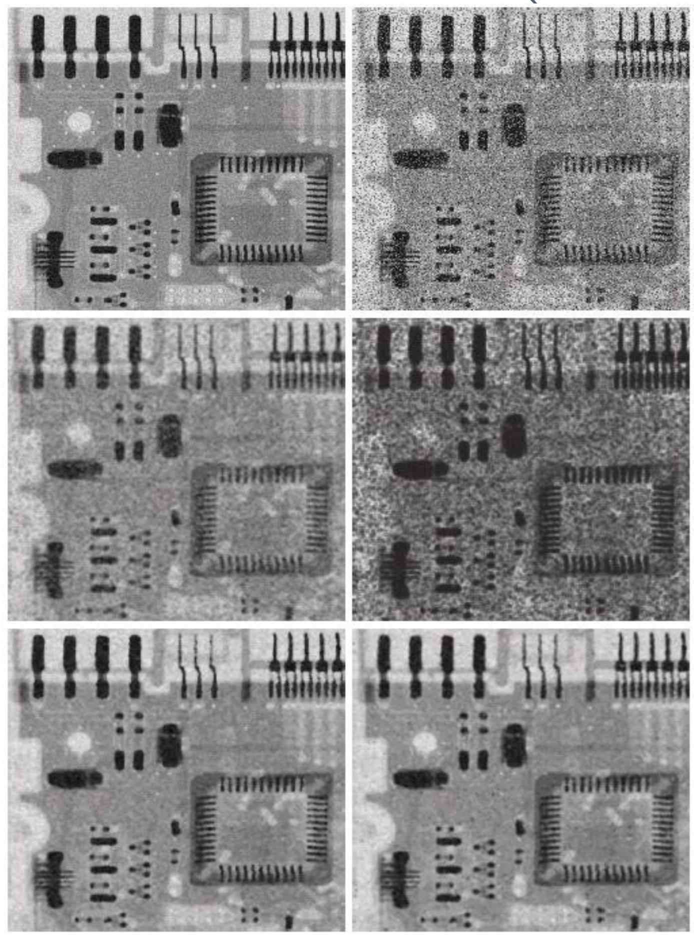
$$\hat{f}(x,y) = \frac{1}{mn-d} \sum_{(s,t)\in S_{xy}} g_r(s,t)$$

d can range from 0 to mn-1. When $d = 0$, this filter reduces to the arithmetic mean filter, when $d = mn$-1, this filter reduces to a median filter. For other values of d, the filter is useful against contaminations with noise of multiple types, such as a combination of salt-and-pepper and Gaussian noise.

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)

Note: For higher order of 'd' in ATM filter it approaches the result of median filter.

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

Adaptive Filters

The behavior of adaptive filters changes based on statistical characteristics of the image inside the filter region S_{xy}.

a) **Adaptive local noise reduction filter**: since the mean gives a measure for average intensity in the region and variance characterizes contrast in that region, they both are reasonable parameters to base an adaptive filter.

Considering a local region S_{xy} of a noisy version $g(x,y)$ of the image $f(x,y)$ with the overall noise variance σ_η^2, local mean of pixels in the region m_L and their local variance σ_L^2, the following rules are to be implemented:

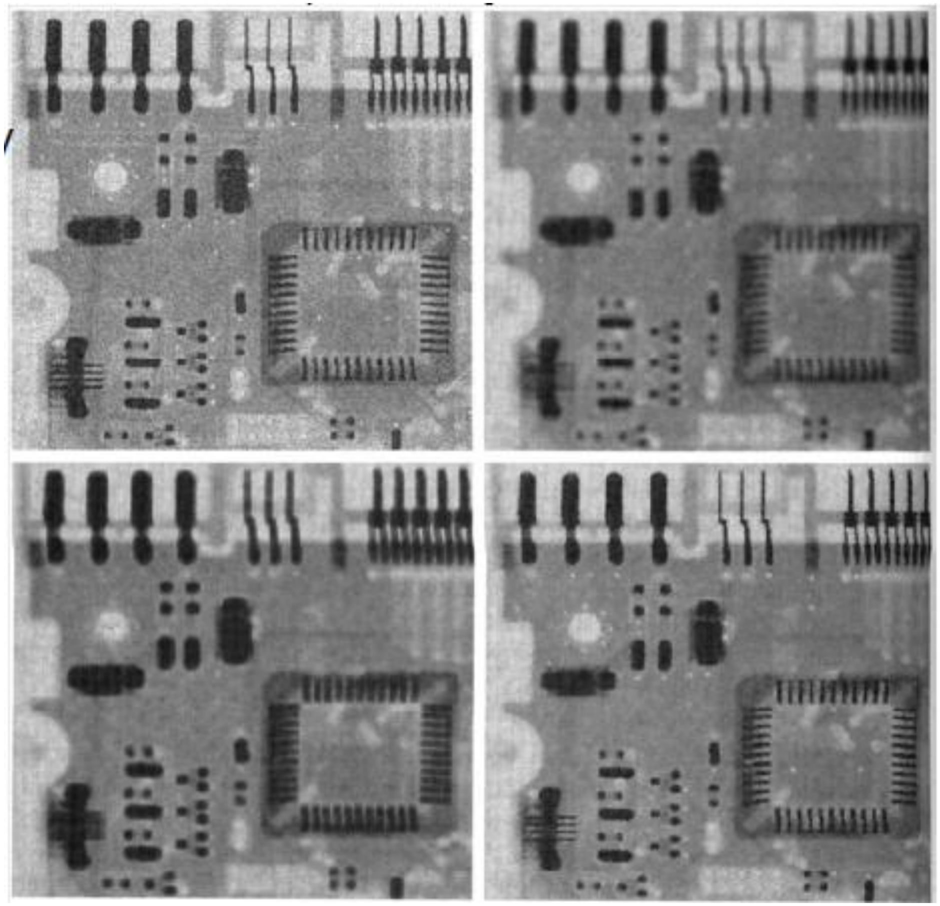
1. If σ_η^2 is zero, the filter returns the value of $g(x,y)$;
2. If $\sigma_L^2 > \sigma_\eta^2$ (typical for edges that needs to be preserved) the filter returns the value close to $g(x,y)$;
3. If two variances are equal, the filter returns the arithmetic mean value for pixels in the region S_{xy}.

Therefore, the adaptive filter is:

$$\hat{f}(x,y) = g(s,t) - \frac{\sigma_\eta^2}{\sigma_L^2}\left[g(s,t) - m_L\right]$$

The only parameter that needs to be known in advance is the overall noise variance σ_η^2; all other parameters are estimates for the selected area.

Image corrupted by Gaussian noise with zero mean, $\sigma_\eta^2 = 1000$



Filtered by a 7x7 arithmetic mean filter

Filtered by a 7x7 geometric mean filter

Filtered by a 7x7 adaptive noise reduction filter with $\sigma_\eta^2 = 1000$

Advantages

Adaptive filter achieves approximately the same performance in noise cancellation but adds much less blurring than the mean filters.

Adaptive filtering typically yields considerably better results in overall performance at the price of filter complexity.

In the example, the exact value of noise variance was used. If a variance estimate used is too low, noise correction will be smaller than it should be. If the estimate is too high, the output image may lose dynamic range.

Adaptive Median Filtering (cont…)

b) **Adaptive median filter**: can handle impulse noise with larger probabilities than traditional median filter. It operates on a rectangular region S_{xy}, whose ***size is changing***.

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 thickening 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 centred at (x, y)

z_{min} = minimum intensity value in S_{xy}

z_{max} = maximum intensity value in S_{xy}

z_{med} = median of intensity value in S_{xy}

z_{xy} = intensity value 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 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

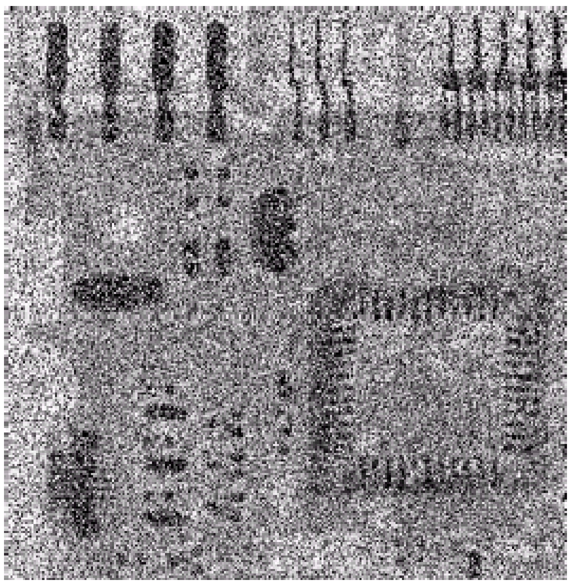
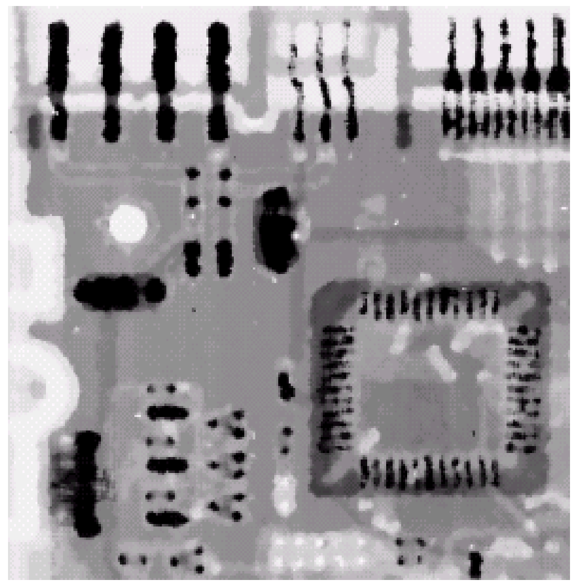
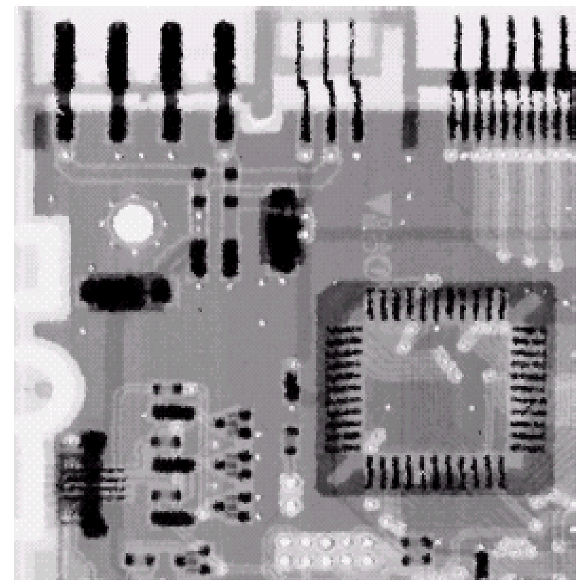


Image corrupted by salt and pepper noise with probabilities $P_a = P_b = 0.25$



Result of filtering with a 7x7 median filter



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

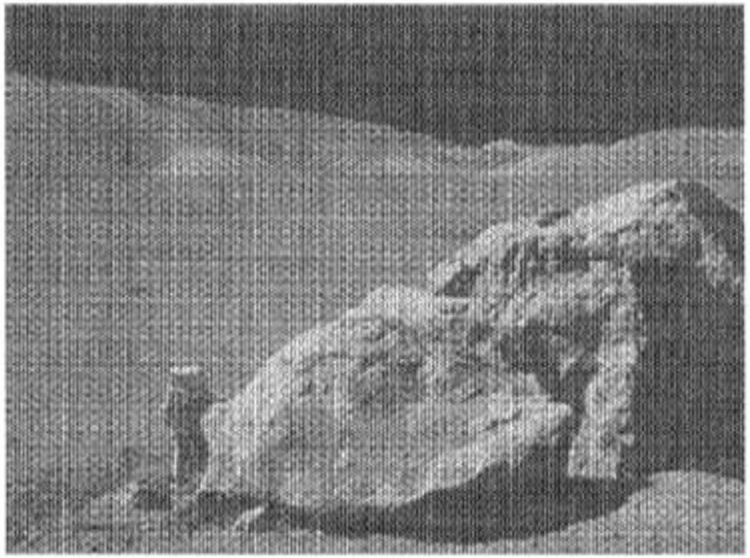
AMF preserves sharpness and details, e.g. the connector fingers.

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

Periodic noise

This noise typically cams from electrical or electromechanical interference during image acquisition. We don't consider any other spatially dependent noise.

If the noise is strong enough, it can be seen in the frequency domain.



Band Reject Filters

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

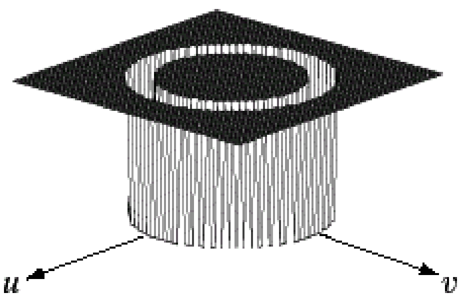
Band reject filters can be used for this purpose

An ideal band reject filter is given as follows:

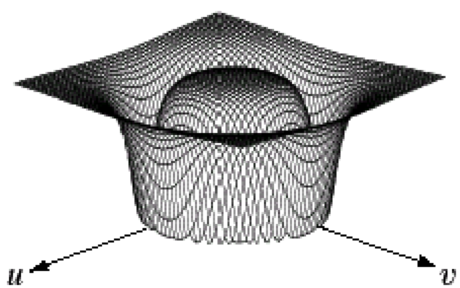
$$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) < D_0 - \frac{W}{2} \\ 0 & \text{if } D_0 - \frac{W}{2} \le D(u,v) \le D_0 + \frac{W}{2} \\ 1 & \text{if } D(u,v) > D_0 + \frac{W}{2} \end{cases}$$

BAND REJECT FILTERS (CONT…)

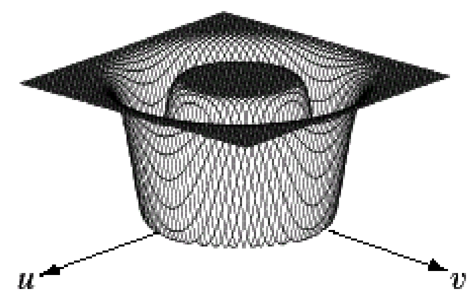
The ideal band reject filter is shown below, along with Butterworth and Gaussian versions of the filter



Ideal Band Reject Filter



Butterworth Band Reject Filter (of order 1)



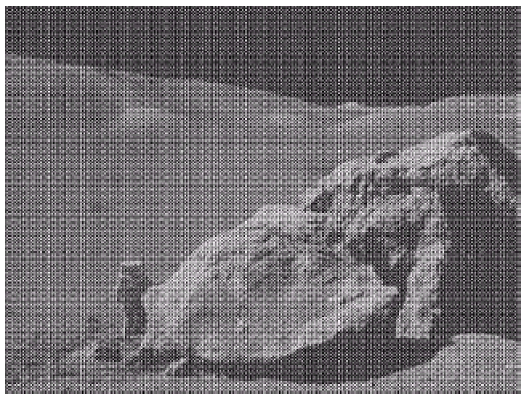
Gaussian Band Reject Filter

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

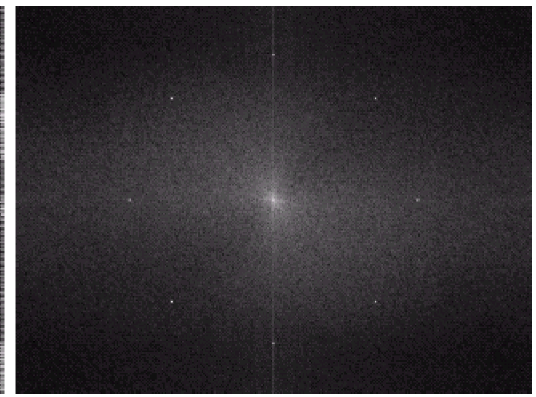
Band Reject Filter Example

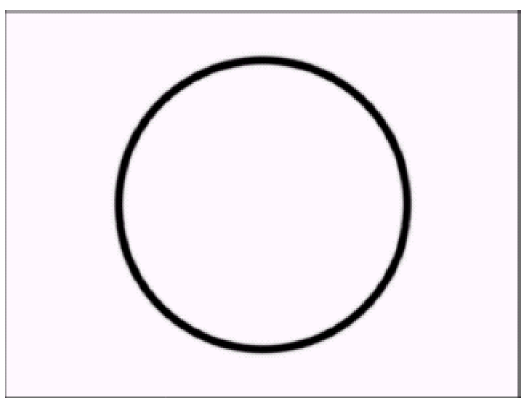
Images taken from Gonzalez & Woods, Digital Image Processing (2002)

Image corrupted by sinusoidal noise



Fourier spectrum of corrupted image





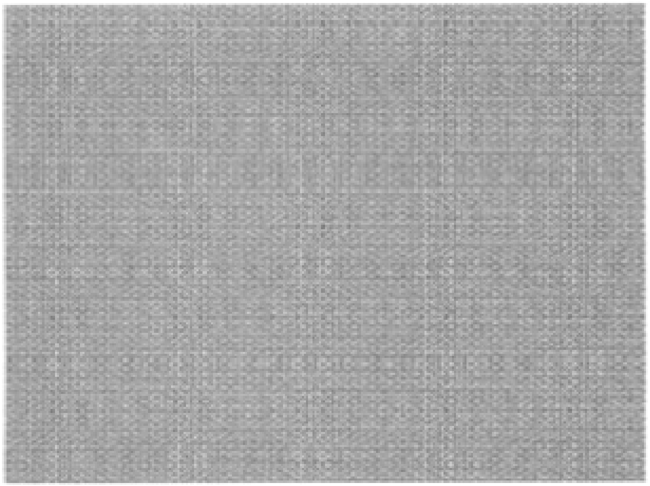
Butterworth band reject filter

Filtered image

Band-pass filters are opposite to band-reject.

$$H_{BP}(u,v) = 1 - H_{BR}(u,v)$$

It is not very common to apply band-pass filtering by itself since it removes too much image details. It is used mostly to isolate the effects on image caused by selected frequency bands; i.e., to isolate and study noise patterns.

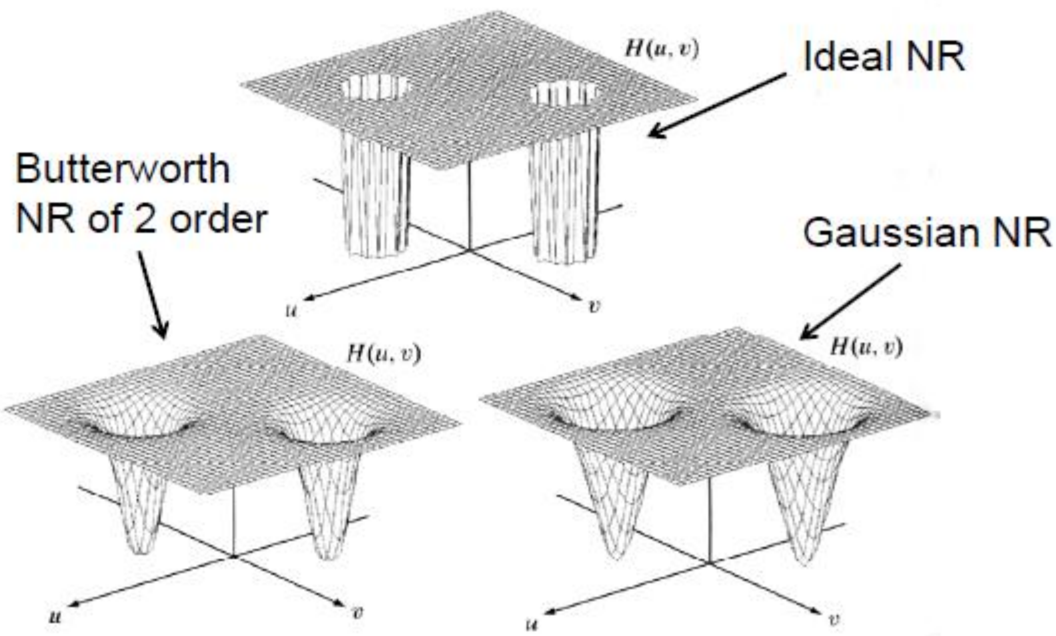


Noise pattern from the previous example

Notch Filters

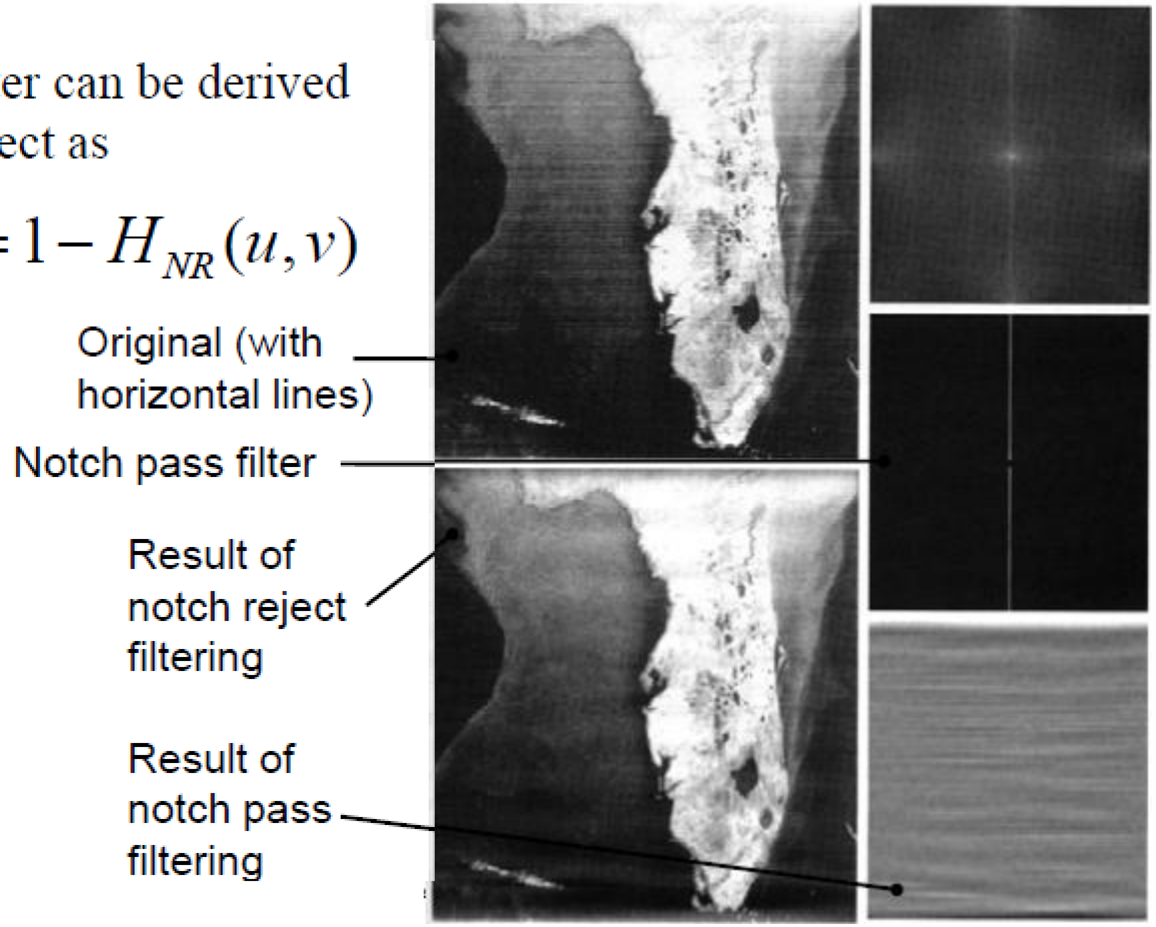
Notch filters reject (or pass) frequencies in predefined neighborhoods.

Because of symmetry of the Fourier transform, notch filters must be symmetric about the origin. The shape of notch filters can be arbitrary.



A notch pass filter can be derived from a notch reject as

$$H_{NP}(u,v) = 1 - H_{NR}(u,v)$$



Estimating the Degradation Function

3 main ways to estimate the degradation function for use in an image restoration:

1. Observation
2. Experimentation
3. Mathematical modeling

Estimation by Image Observation

The degradation is assumed to be *linear* and *position-invariant*

- Look at a small rectangular section of the image containing sample structures, and in which the signal content is strong (e.g. high contrast): subimage $g_s(x,y)$
- Process this subimage to arrive at a result as good as possible: $\hat{f}_s(x,y)$

Assuming the effect of noise is negligible in this area: $H_s(u,v) = \dfrac{G_s(u,v)}{\hat{F}_s(u,v)}$

=> deduce the complete degradation function $H(u,v)$ (position invariance)

Example1: Image with blur. (Consider each small sub images and try to find high contrast and the deblur the portion).
Note: This can be achieved by sharpening the Image.
Example2: Restoring Historical Photographs

Estimation by Experimentation

If an equipment similar to the one used to acquire the degraded image is available:

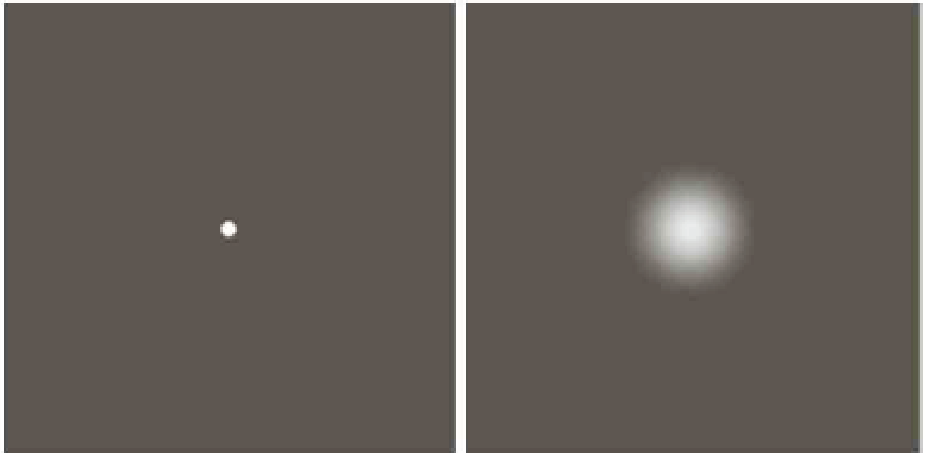
- Find system settings reproducing the most similar degradation as possible
- Obtain an impulse response of the degradation by imaging an impulse (dot of light)

FT of an impulse = constant => $$H(u,v) = \frac{G(u,v)}{A}$$

A = constant describing the strength of the impulse

a b

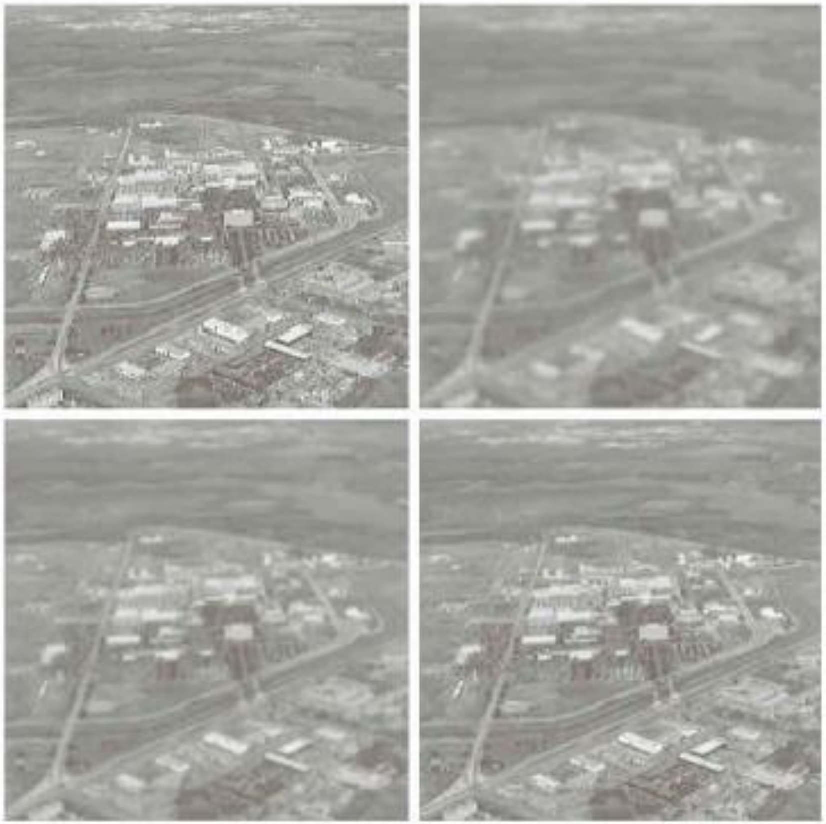
FIGURE 5.24 Degradation estimation by impulse characterization. (a) An impulse of light (shown magnified). (b) Imaged (degraded) impulse.



Estimation by Modeling

Example 1: degradation model proposed by Hufnagel and Stanley [1964] based on the physical characteristics of atmospheric turbulence: $H(u,v) = e^{-k(u^2+v^2)^{5/6}}$

a) Negligible turbulence
b) Severe Turbulence with k=0.0025
c) Mild turbulence k=0.001
d) Low turbulence k=0.00025



Example 2: derive a mathematical model starting form basic principles

Illustration: image blurring by uniform linear motion between the image and the sensor during image acquisition

If T is the duration of exposure the blurred image can be expressed as:

$$g(x,y) = \int_0^T f\left[x - x_0(t), y - y_0(t)\right] dt$$

FT[g(x,y)] => $G(u,v) = F(u,v) \int_0^T e^{-j2\pi[ux_0(t)+vy_0(t)]} dt$

$$H(u,v) = \int_0^T e^{-j2\pi[ux_0(t)+vy_0(t)]} dt \quad \Rightarrow \quad G(u,v) = H(u,v)F(u,v)$$

E.g. if uniform linear motion in the *x*-direction only, at a rate $x_0(t) = at/T$

$$H(u,v) = \frac{T}{\pi ua} \sin(\pi ua) e^{-j\pi ua}$$

NB: $H = 0$ for $u = n/a$

If motion in y as well:

$$H(u,v) = \frac{T}{\pi(ua+vb)} \sin\left[\pi(ua+vb)\right] e^{-j\pi(ua+vb)}$$

a b

FIGURE 5.26 (a) Original image. (b) Result of blurring using the function in Eq. (5.6-11) with $a = b = 0.1$ and $T = 1$.

Inverse filtering

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)} \qquad \text{(array operation)}$$

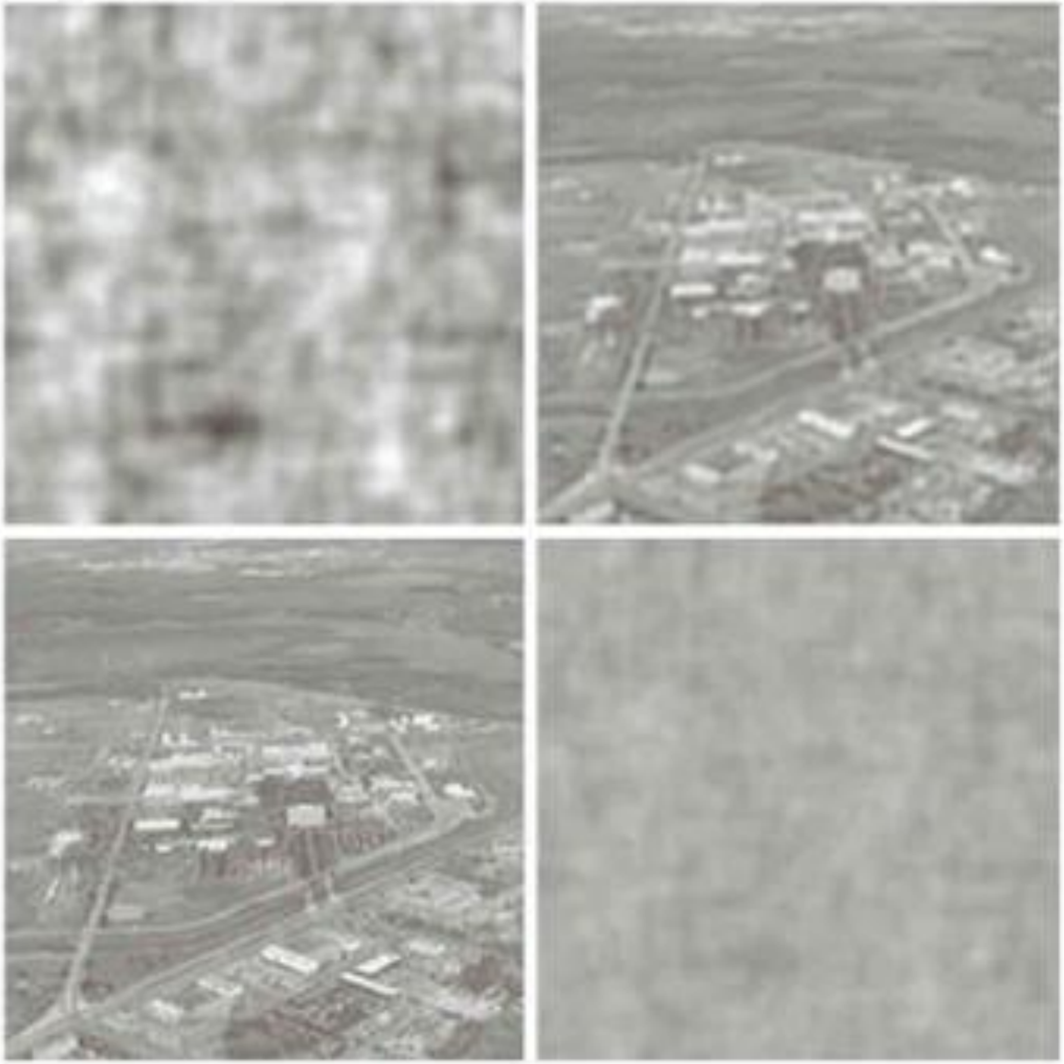
$$G(u,v) = H(u,v)F(u,v) + N(u,v) \qquad \Rightarrow \quad \hat{F}(u,v) = F(u,v) + \frac{N(u,v)}{H(u,v)}$$

$\Rightarrow$ Even if we know *H(u,v)*, we cannot recover the “undegraded” image exactly because *N(u,v)* is not known
$\Rightarrow$ If *H* has zero or very small values, the ration *N/H* could dominate the estimate

One approach to get around this is to limit the filter frequencies to values near the origin

a) Normal Filter
b) Result of H cut off outside the radaii 40
c) Result of H cut off outside the radaii 70
d) Result of H cut off outside the radaii 85

Note: Image dominated with the noise



Minimum Mean Square Error (Wiener) Filtering

Objective: find an estimate $\hat{f}$ of the uncorrupted image such that the mean square error between them is minimized: $e^2 = E\{(f - \hat{f})^2\}$

Assumptions:

- Noise and image are uncorrelated
- One or the other has zero mean
- The intensity levels in the estimate are a linear function of the levels in the degraded image

The minimum of the error function *e* is given by:

$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + S_\eta(u,v)/S_f(u,v)}\right] G(u,v)$$

$S_\eta(u,v) = |N(u,v)|^2 =$ Power spectrum of the noise (autocorrelation of noise)

$S_f(u,v) = |F(u,v)|^2 =$ Power spectrum of the undegraded image

Note: When Noise is zero, then power spectrum vanishes and Wiener filter results as inverse filter

Minimum Mean Square Error (Wiener) Filtering

When the two spectrums are not known or cannot be estimated, approximate to:

$$\hat{F}(u,v) = \left[\frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + K}\right] G(u,v)$$

Where K is a specified constant

Hint: Considering the Signal to Noise ratio in the image (SNR)

a b c

a) Full filter b)inverse filter c)Wiener filter

	Image + motion blur + add. noise	Result of Inverse filtering	Result of Wiener filtering
Reduced noise variance			
Reduced noise variance			

a b c
d e f
g h i

FIGURE 5.29 (a) 8-bit image corrupted by motion blur and additive noise. (b) Result of inverse filtering. (c) Result of Wiener filtering. (d)–(f) Same sequence, but with noise variance one order of magnitude less. (g)–(i) Same sequence, but noise variance reduced by five orders of magnitude from (a). Note in (h) how the deblurred image is quite visible through a "curtain" of noise.

Geometric mean filter

Generalization of the Wiener filter:

$$\hat{F}(u,v) = \left[\frac{H^*(u,v)}{|H(u,v)|^2}\right]^{\alpha} \left[\frac{H^*(u,v)}{|H(u,v)|^2 + \beta\left[\frac{S_\eta(u,v)}{S_f(u,v)}\right]}\right]^{1-\alpha} G(u,v)$$

α and β being positive real constants

α = 1 => inverse filter
α = 0 => *parametric Wiener filter* (standard Wiener filter when β = 1)
α = 1/2 => actual geometric mean
α = 1/2 and β = 1 => *spectrum equalization filter*

Constrained Least Squares Filtering

In vector-matrix form: $\boldsymbol{g} = \boldsymbol{H}\boldsymbol{f} + \boldsymbol{\eta}$

g**, **f**, **η vectors of dimension MNx1
H matrix of dimension MNxMN => very large !

Issue: Sensitivity of H to noise
$\Rightarrow$ Optimality of restoration based on a measure of smoothness: e.g. Laplacian
$\Rightarrow$ Find the minimum of a criterion function C:

$$C = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[\nabla^2 f(x,y)\right]^2$$

subject to the constraint: $||\boldsymbol{g} - \boldsymbol{H}\hat{\boldsymbol{f}}||^2 = ||\boldsymbol{\eta}||^2$ (Euclidean vector norm)

Constrained Least Squares Filtering

Frequency domain solution:

$$\hat{F}(u,v) = \left[\frac{H^*(u,v)}{|H(u,v)|^2 + \gamma |P(u,v)|^2} \right] G(u,v)$$

With:

- γ = parameter to adjust so that the constraint is satisfied
- $P(u,v)$ = Fourier Transform of the function:

$$p(x,y) = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

NB: $\gamma = 0$ => Inverse Filtering

Constrained Least Squares Filtering

Results adjusting γ interactively

a b c

FIGURE 5.30 Results of constrained least squares filtering. Compare (a), (b), and (c) with the Wiener filtering results in Figs. 5.29(c), (f), and (i), respectively.

Wiener Filtering

Adjusting γ so that the constraint is satisfied
(an algorithm)

Goal: find γ so that: $\|\boldsymbol{g} - \boldsymbol{H}\hat{\boldsymbol{f}}\|^2 = \|\boldsymbol{\eta}\|^2 \pm a$ (1) (a = accuracy factor)

1. Specify an initial value of γ
2. Compute the corresponding residual $\|\boldsymbol{r}\|^2 = \|\boldsymbol{g} - \boldsymbol{H}\hat{\boldsymbol{f}}\|^2$
3. Stop if Eq. (1) is satisfied. Otherwise:
   - if $\|\boldsymbol{r}\|^2 < \|\boldsymbol{\eta}\|^2 - a$, increase γ,
   - if $\|\boldsymbol{r}\|^2 > \|\boldsymbol{\eta}\|^2 + a$, decrease γ,
   - Then return to step 2.

Adjusting γ so that the constraint is satisfied

$$||\boldsymbol{r}||^2 = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} r^2(x, y)$$

$$||\boldsymbol{\eta}||^2 = MN\left[\sigma_n^2 + m_n^2\right]$$

$$m_n = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \eta(x, y) \qquad \text{(average)}$$

$$\sigma_n^2 = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} \left[\eta(x, y) - m_n\right]^2 \qquad \text{(variance)}$$

END