Ψηφιακή Επεξεργασία Εικόνας (ΨΕΕ) – MYE037

Εαρινό εξάμηνο 2023-2024

Image Restoration and Reconstruction (Noise Removal)

Άγγελος Γιώτης

a.giotis@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

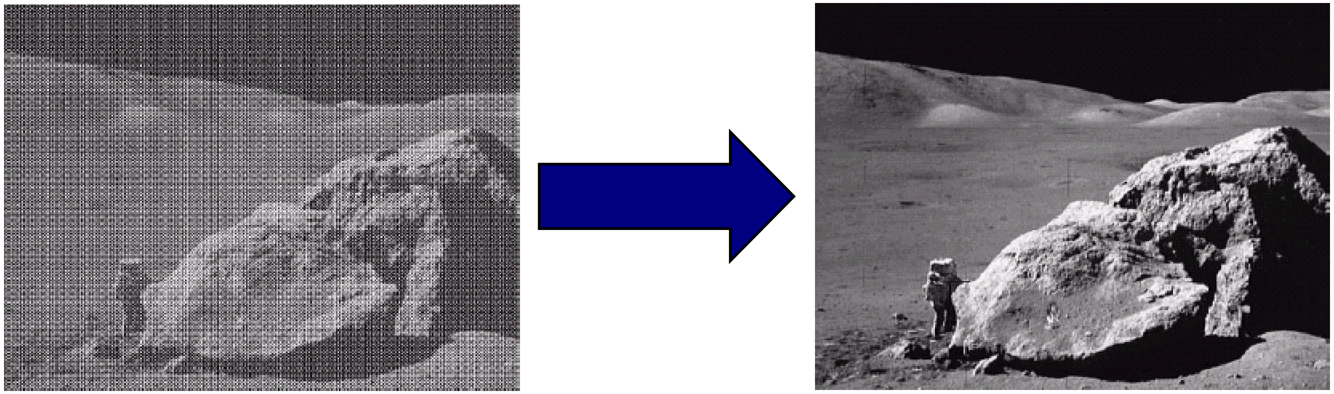
In this lecture we will look at image restoration techniques used for noise removal

- 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 (observation) Model

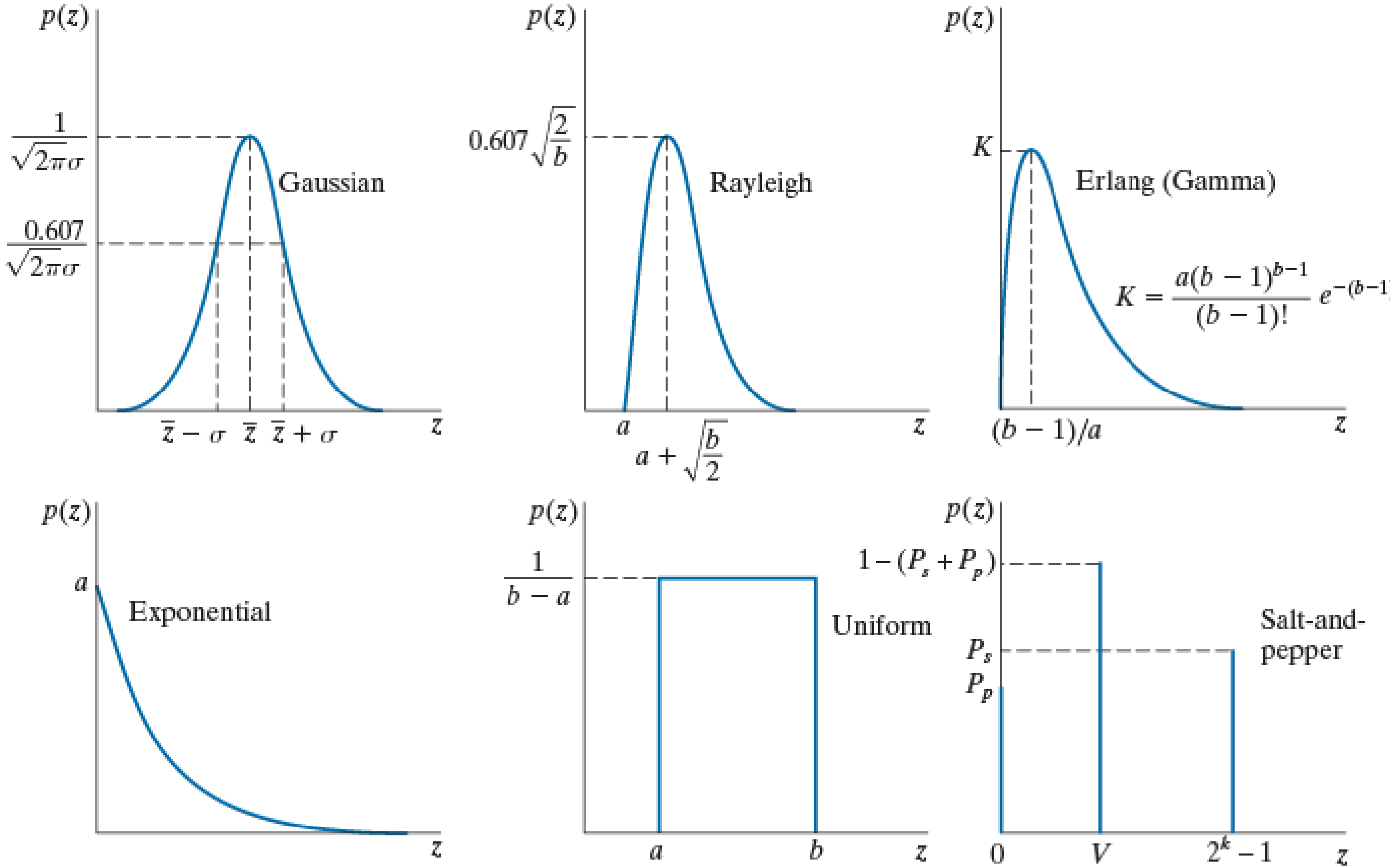
- We can model the observed image with respect to noise as:

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

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

- The model $\mu()$ describes the observation process: it specifies how we believe noise (and possibly other factors) affect the image.
- If we can estimate $\mu()$ we can figure out how to restore the image
- Many different ways to define η and μ

Noise Models (cont...)



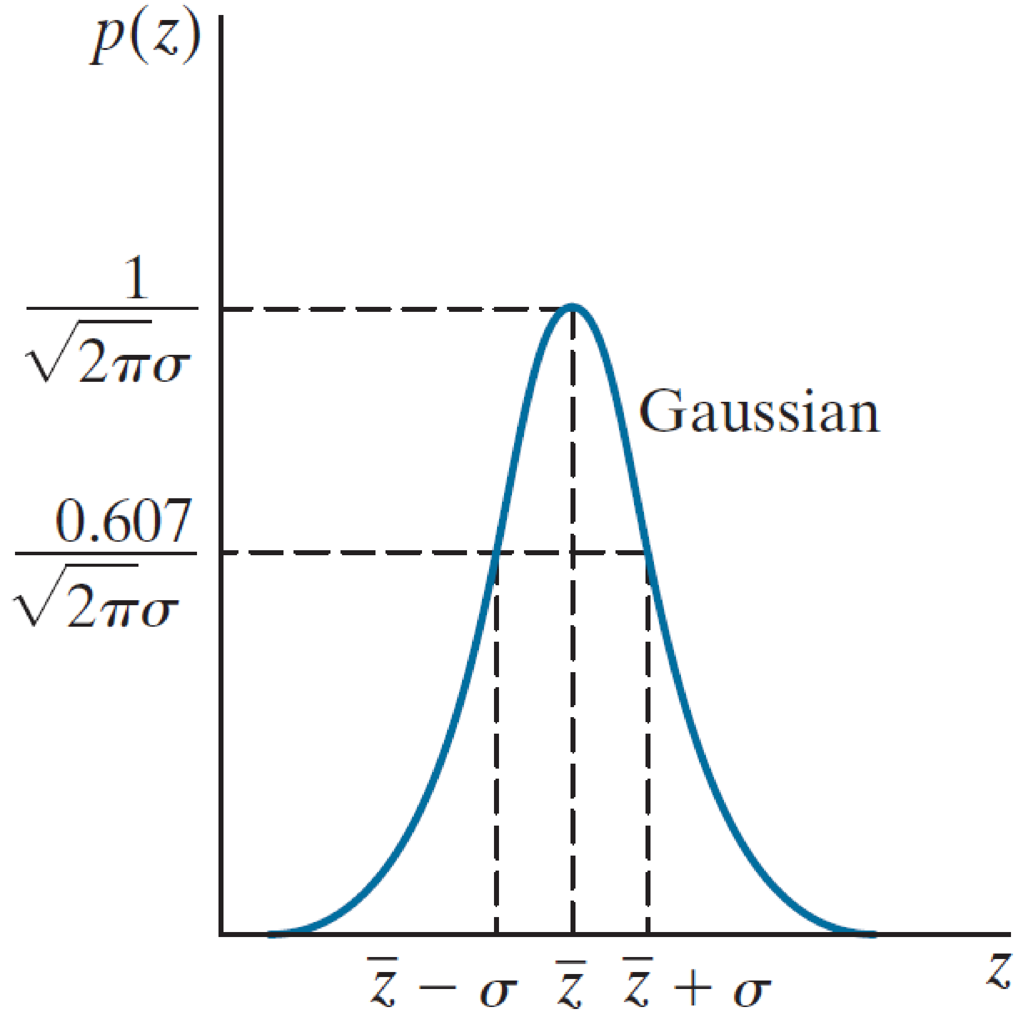
Noise Models (cont...)

There are many ways to define random variable $z = \eta(x, y)$.

The probability density function of a Gaussian z:

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

$$-\infty < z < \infty$$



Noise Models (cont...)

There are many ways to define random variable $z = \eta(x, y)$.

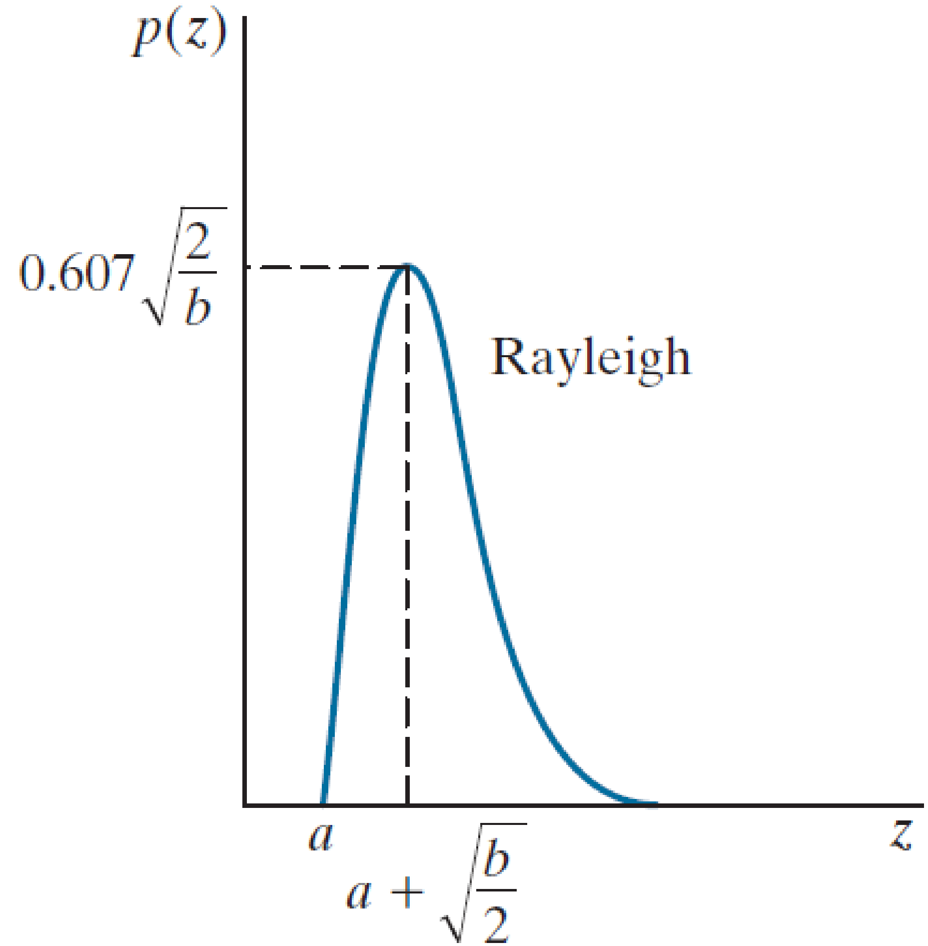
The probability density function of a Rayleigh z:

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

The mean and variance of z:

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

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



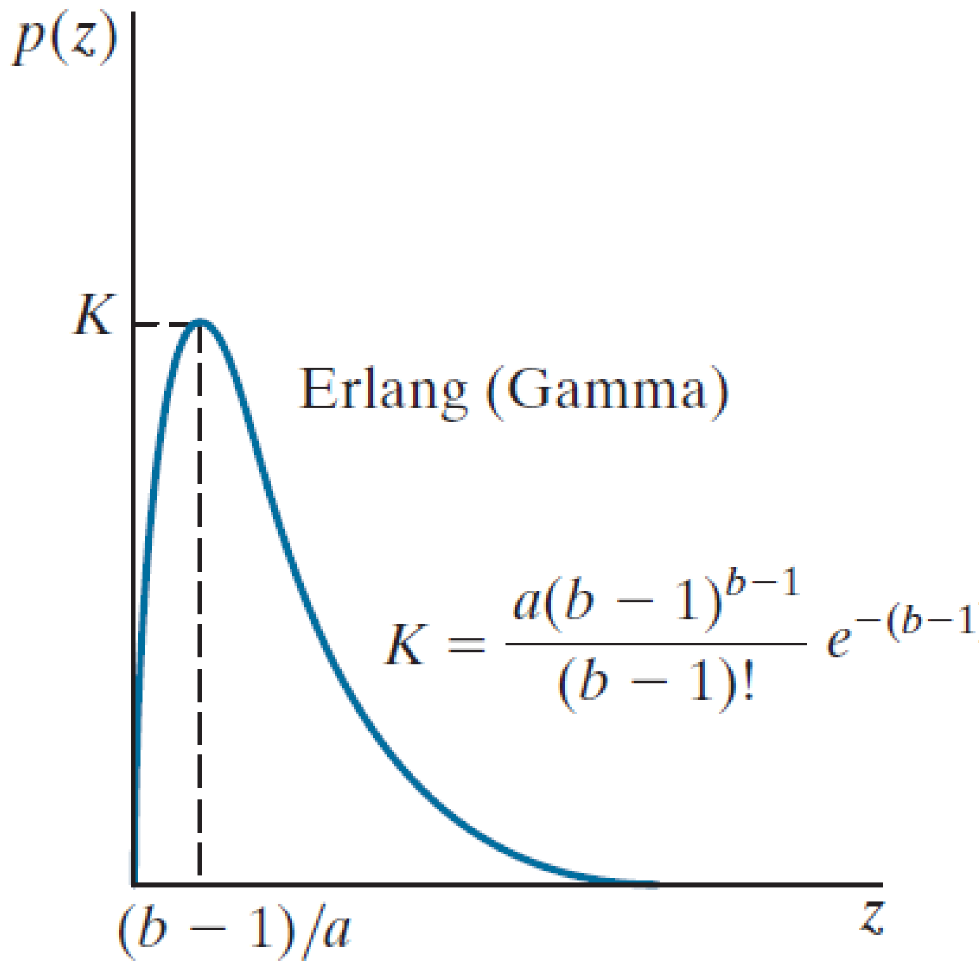
Noise Models (cont...)

- There are many ways to define random variable $z = \eta(x, y)$.
- The probability density function of a Gamma (or Erlang) z:

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

The mean and variance of z:

$$\bar{z} = \frac{b}{a} \qquad \sigma^2 = \frac{b}{a^2}$$



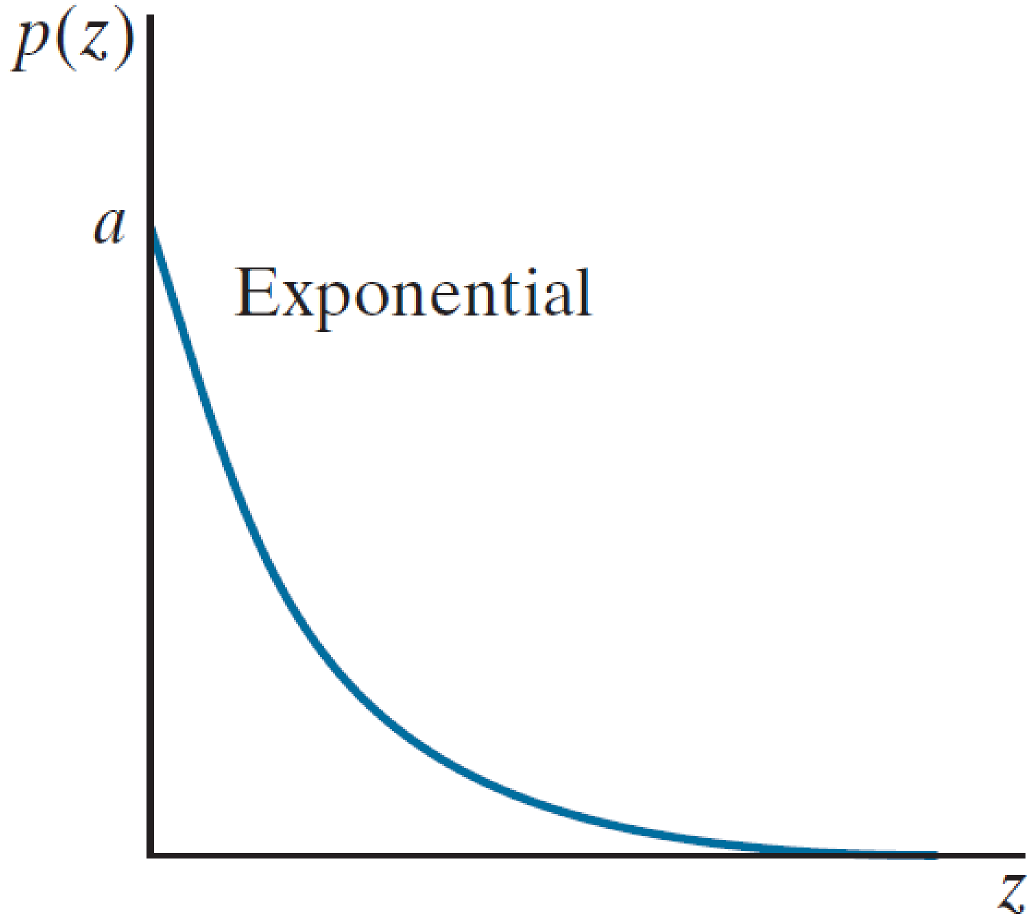
Noise Models (cont...)

- There are many ways to define random variable $z = \eta(x, y)$.
- The probability density function of an exponential z:

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

The mean and variance of z:

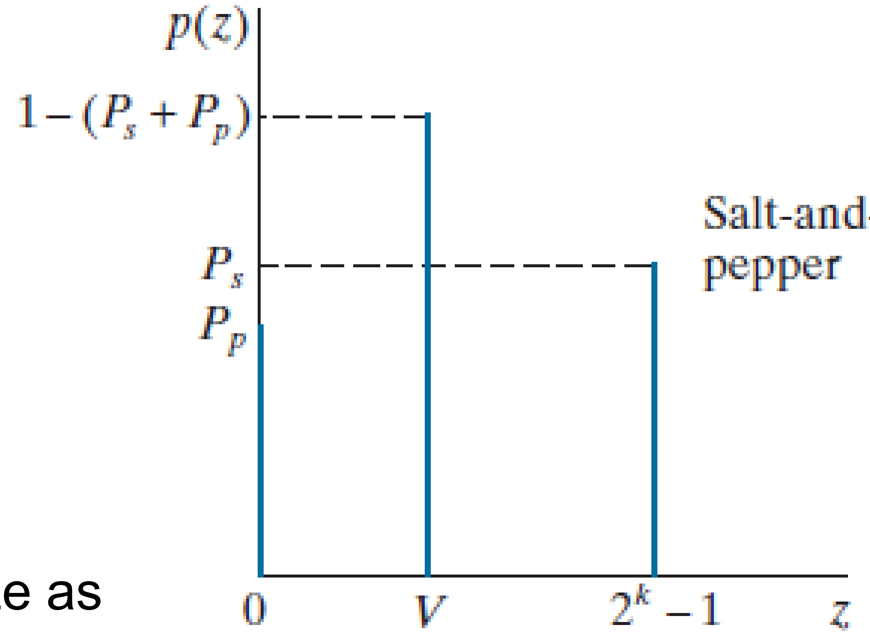
$$\bar{z} = \frac{1}{a} \qquad \sigma^2 = \frac{1}{a^2}$$



Noise Models (Salt and pepper)

- The probability density function of an impulse z (or salt-and-pepper noise):

$$p(z) = \begin{cases} P_s & \text{for } z = 2^k - 1 \\ P_p & \text{for } z = 0 \\ 1 - (P_s + P_p) & \text{for } z = V \end{cases}$$



Given an image *f(x, y)*, of the same size as *η(x, y)*, we corrupt it with salt-and-pepper noise by assigning a 0 to all locations in *f* where a 0 occurs in *η*. Similarly, we assign a value of $2^k - 1$ to all location in *f* where that value appears in *η*. Finally, we leave unchanged all locations in *f* where V occurs in *η*.

If neither P_s nor P_p is zero, and especially if they are equal, noise values will be white ($2^k - 1$) or black (0), and will resemble salt and pepper granules distributed randomly over the image

Standard Additive noise model

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

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

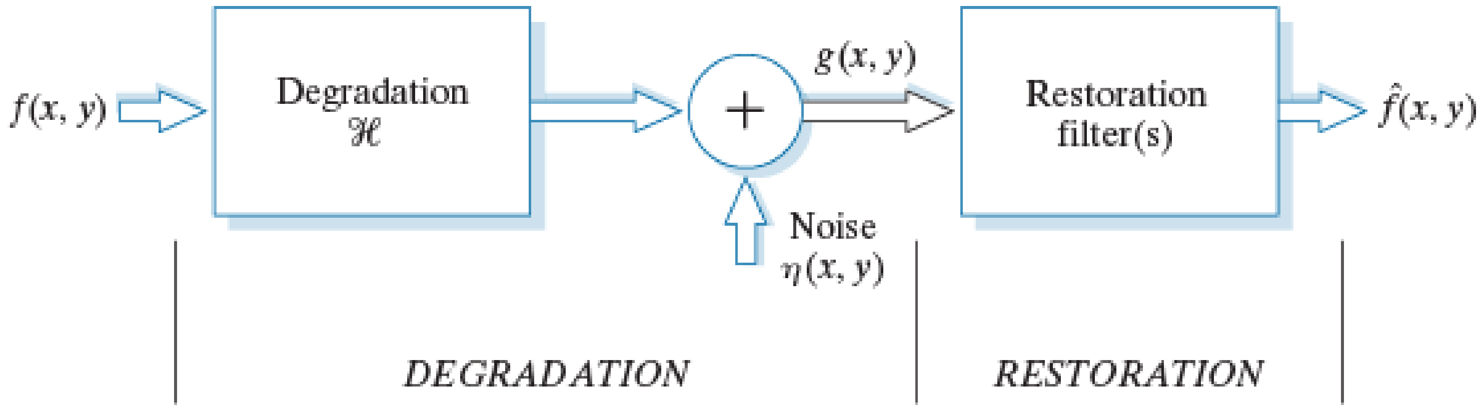
- For noise, we assume that for each point n in the image:

$E[\eta_n] = 0$ and $E[\eta_n^2] = \sigma_\eta^2$

whereas noise is **uncorrelated** between different points m, n:

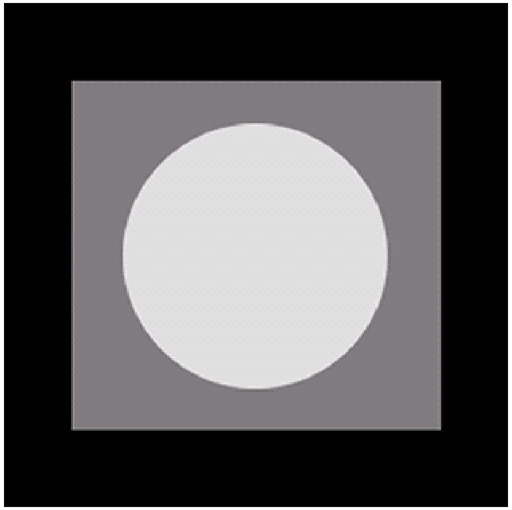
$E[\eta_m \eta_n] = 0$

- This is known as the "**white Gaussian noise**" assumption.

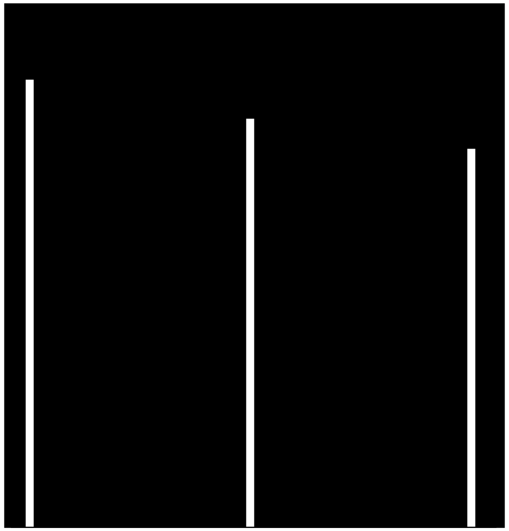


Noise Example

- 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

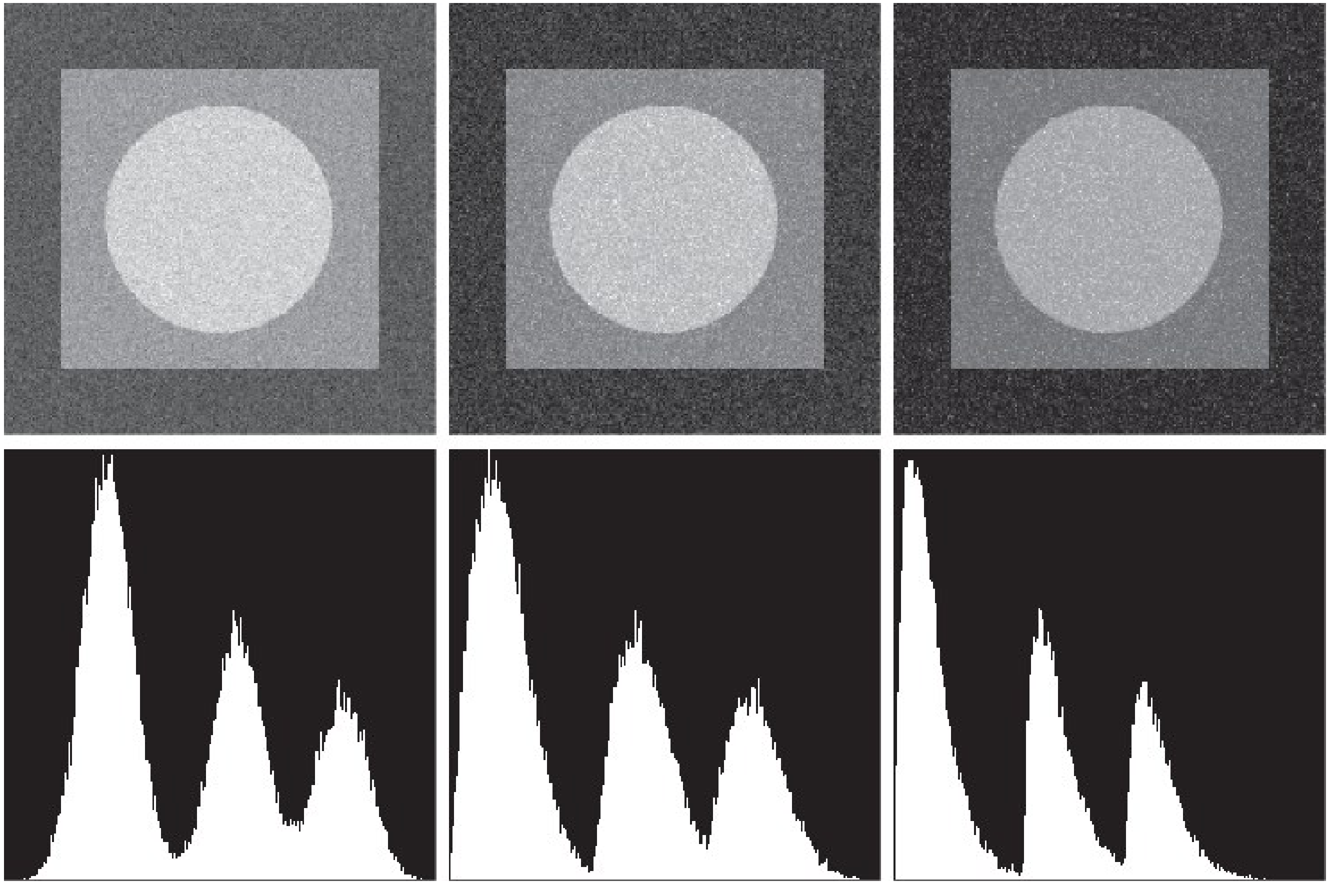


Image



Histogram

Noise Example (cont…)



a b c
d e f

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

Noise Example (cont…)

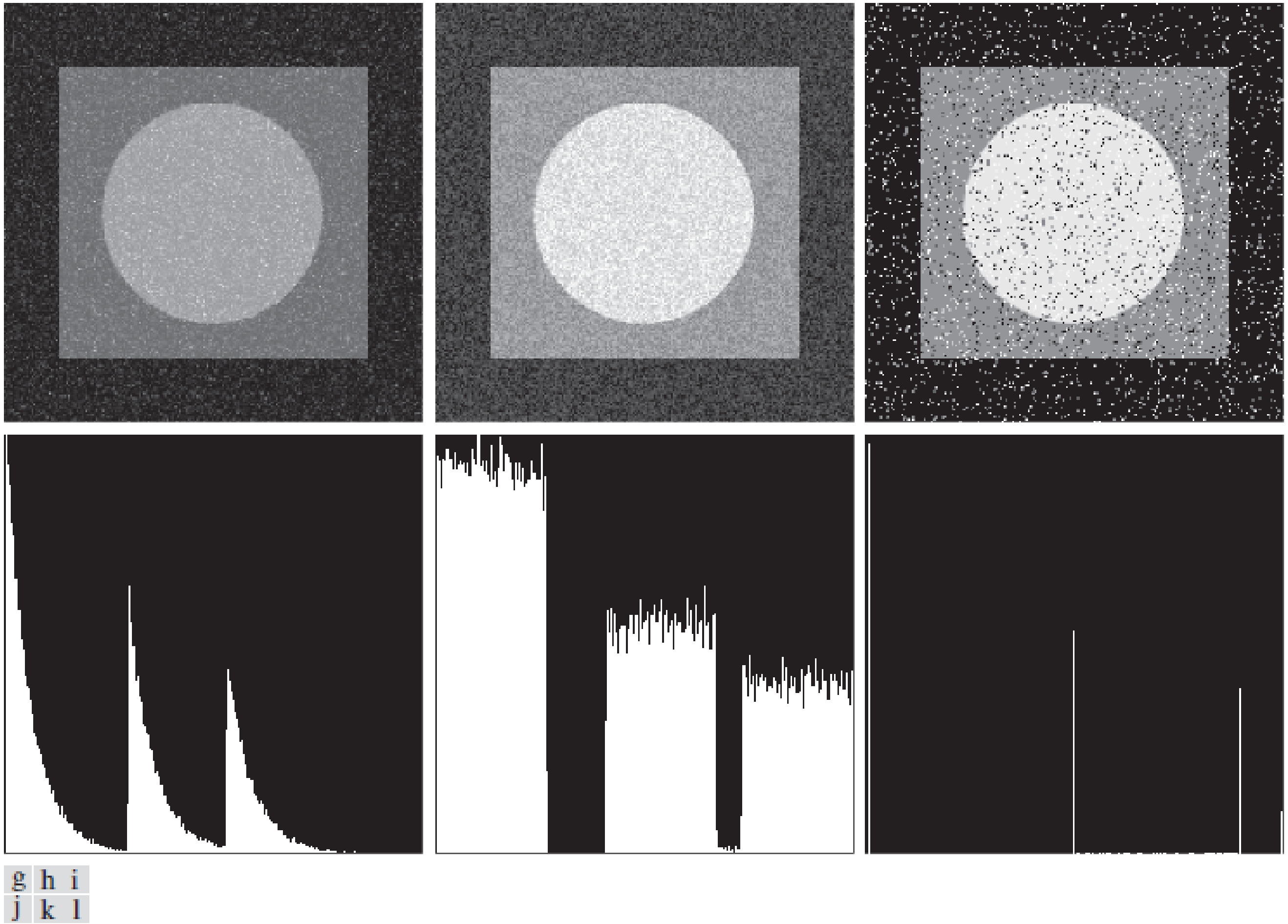


FIGURE 5.4 (*continued*) Images and histograms resulting from adding exponential, uniform, and salt-and-pepper noise to the image in Fig. 5.3. In the salt-and-pepper histogram, the peaks in the origin (zero intensity) and at the far end of the scale are shown displaced slightly so that they do not blend with the page background.

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.

Example Restoration using combination of observations

- Suppose:

$$\hat{f}(x,y) = \bar{g}(x,y) = \frac{1}{K}\sum_{i=1}^{K} g_i(x,y)$$

- Which is our estimation for the actual image
- Note that pixel intensities for each point in the 2D image can be seen as random variables
- We can demonstrate that, by calculating the expected value and the standard deviation of the pixel intensities of the estimated image, it will have less noise than that in the original image.

Example Restoration using combination of observations

- Expected value:

$$E\left[\bar{g}(x,y)\right] = E\left[\frac{1}{K}\sum_{i=1}^{K} g_i(x,y)\right] = \frac{1}{K}E\left[\sum_{i=1}^{K} g_i(x,y)\right]$$

$$= \frac{1}{K}E\left[\sum_{i=1}^{K} f(x,y) + \eta_i(x,y)\right]$$

$$= \frac{1}{K}E\left[\sum_{i=1}^{K} f(x,y)\right] + \frac{1}{K}E\left[\sum_{i=1}^{K} \eta_i(x,y)\right]$$

$$= \frac{1}{K}Kf(x,y) + \frac{1}{K}K0 \;= f(x,y)$$

$$\mathrm{E}[f(x,y) + \cdots + f(x,y)] =$$
$$= E[Kf(x,y)] = KE[f(x,y)]$$
$$= K\, f(x,y)$$

Example Restoration using combination of observations

- Similarly, the variance of the new image is:

$$\sigma_{\hat{f}}^2 = \sigma_{\bar{g}}^2 = E[\bar{g}^2] - E[\bar{g}]^2 = \frac{1}{\mathrm{K}} \sigma_\eta^2$$

- Remember white Gaussian noise assumption: $E[\eta_n^2] = \sigma_\eta^2$ (do the rest as HW)
- As *K* increases, the variability of the brightness of each image element decreases, resulting in approaching the ideal noise-free image f(x, y).
- Note that images (f and η) must be aligned (registered)

Example Restoration using smoothing

- If we only have one observation/image available, what can we do?
- Suppose we apply a smoothing filter h. We can represent the process as a convolution with the filter:

$$\hat{f} = h * (f + \eta) = h * f + h * \eta$$

- Again, we compute the same statistical moments. That is, we calculate the **expected value**:

$$\mathrm{E}[\hat{f}] = \mathrm{E}[h * f + h * \eta] = h * E[f] = h * f$$

Example Restoration using smoothing

- And the **variance** (and standard deviation), for which we assume:

$$\hat{f} = h * f + h * \eta = \bar{f} + \bar{\eta}$$

- $$\sigma_{\hat{f}}^2 = E[\hat{f}^2] - E[\hat{f}]^2$$
  $$= E\left[(\bar{f} + \bar{\eta})^2\right] - (\bar{f})^2 = E[(\bar{\eta})^2]$$

Example Restoration using smoothing

- Assuming that h is a mean filter, for pixel n, we have:

$$\bar{\eta}(n) = (h * \eta)(n) = \frac{1}{N} \sum_{k \in \Gamma(n)} \eta(k)$$

- Hence:

$$\mathrm{E}[\bar{\eta}(n)^2] = \mathrm{E}[\left(\frac{1}{N} \sum_{k \in \Gamma(n)} \eta(k) \right)^2]$$

Example Restoration using smoothing

- Hence:

$$\mathrm{E}\left[\left(\frac{1}{N}\sum_{k\in\Gamma(n)}\eta(k)\right)^2\right]=\frac{1}{N^2}\sum_{k\in\Gamma(n)}\mathrm{E}[\eta(k)^2]$$

$$+\frac{2}{N^2}\sum_{l\in\Gamma(n)}\sum_{\substack{m\in\Gamma(n)\\ m\neq l}}\mathrm{E}[\eta(n-l)\eta(n-m)]$$

$$=\frac{1}{N^2}\sum_{k\in\Gamma(n)}\mathrm{E}[\eta(k)^2]=\frac{1}{N^2}\sum_{k\in\Gamma(n)}\sigma_\eta^2$$

Example Restoration using smoothing

- Consequently:

$$\sigma_{\hat{f}}^2 = \frac{1}{N^2} \sum_{k \in \Gamma(n)} \sigma_\eta^2 = \frac{1}{N^2} \mathrm{N} \sigma_\eta^2 = \frac{1}{\mathrm{N}} \sigma_\eta^2$$

- Positive: The effect of noise is reduced in the estimated image.
- Negative: Unfortunately, since we are using a smoothing filter, along with the noise, we also lose potentially significant details, such as edges.

Restoration in the presence of noise only (continue..)

- 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:

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

- Works well for salt noise, but fails for pepper noise.
- Also does well for other kinds of noise such as Gaussian noise.

Restoration in the presence of noise only (cont.)

Contraharmonic Mean:

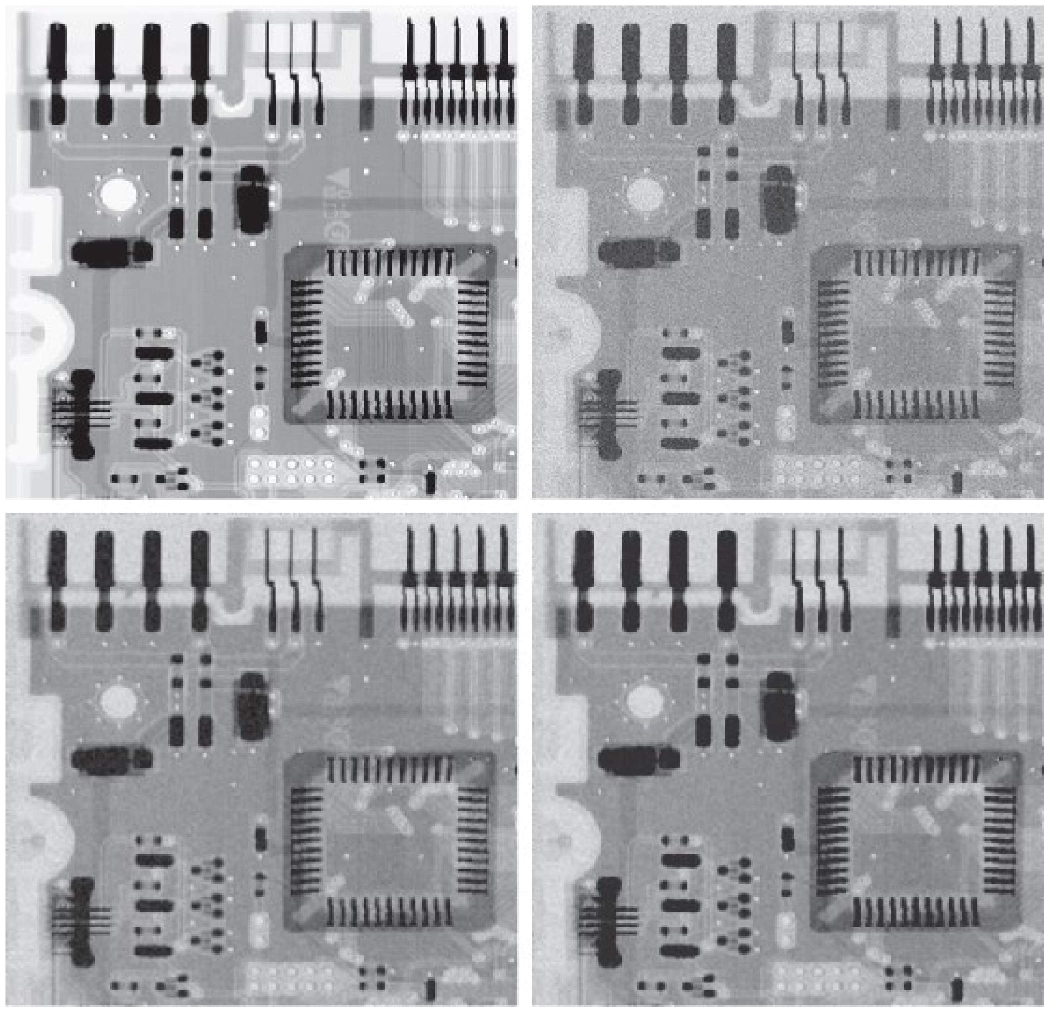
$$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}}$$

- 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

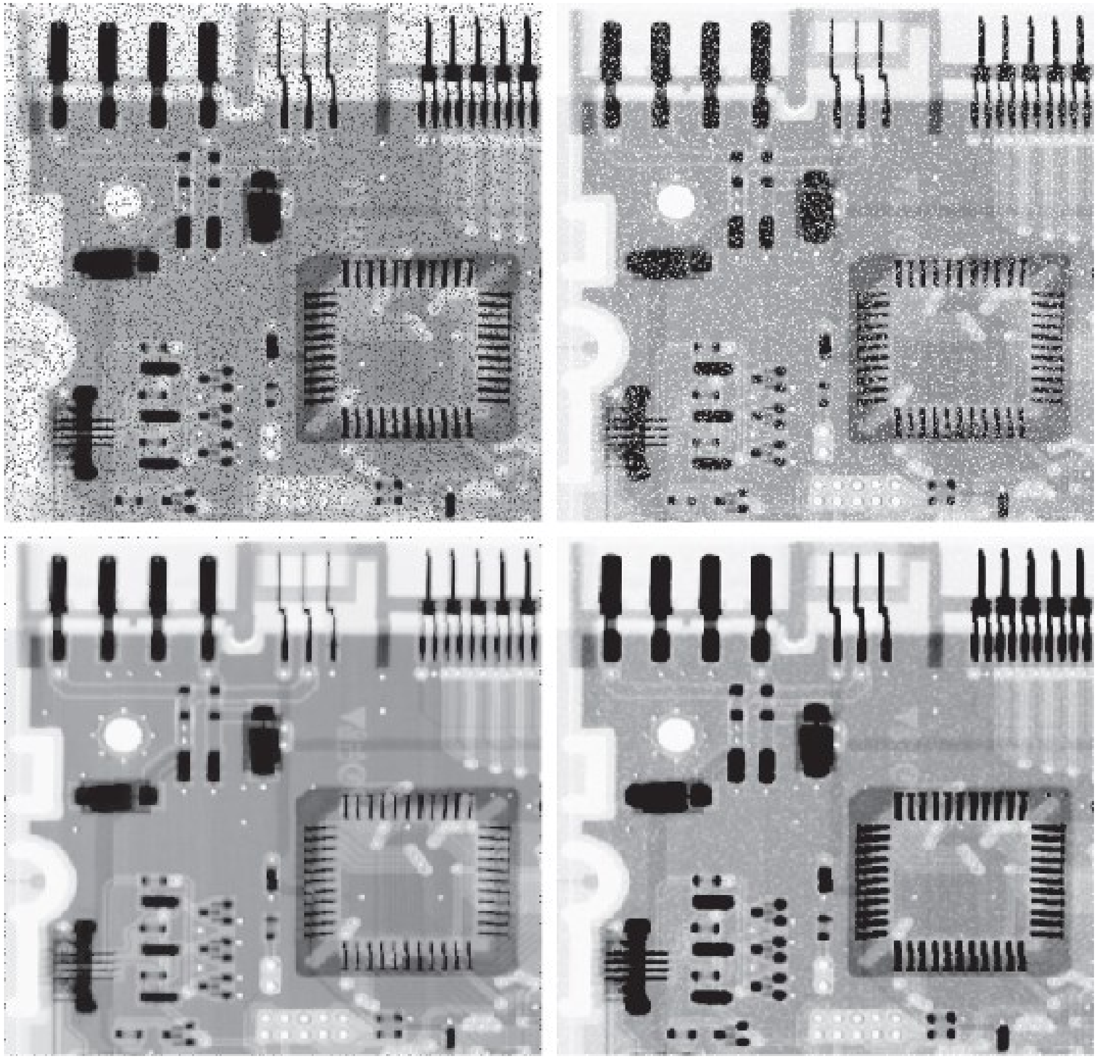
Image corrupted by Gaussian noise



3x3 Arithmetic Mean Filter

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

Noise Removal Examples (cont…)

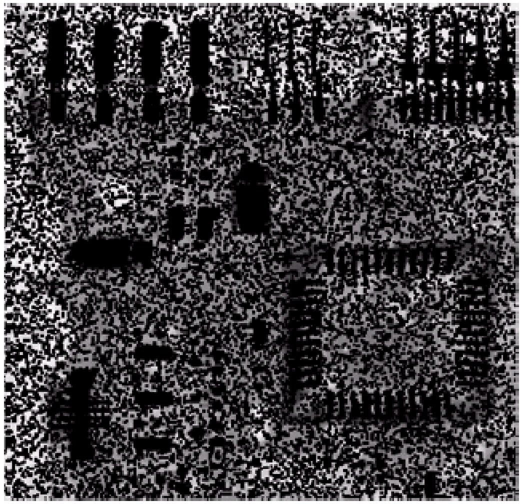


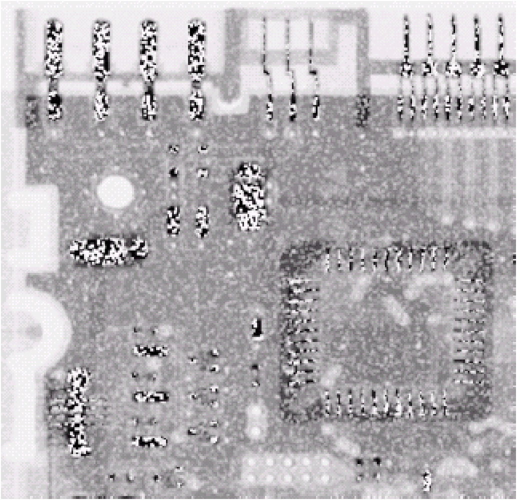
a b
c d

FIGURE 5.8 (a) Image corrupted by pepper noise with a probability of 0.1. (b) Image corrupted by salt noise with the same probability. (c) Result of filtering (a) with a 3×3 contraharmonic filter $Q = 1.5$. (d) Result of filtering (b) 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

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

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:

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

- Good for random Gaussian and uniform noise.

Alpha-Trimmed Mean Filter

Alpha-Trimmed Mean Filter:

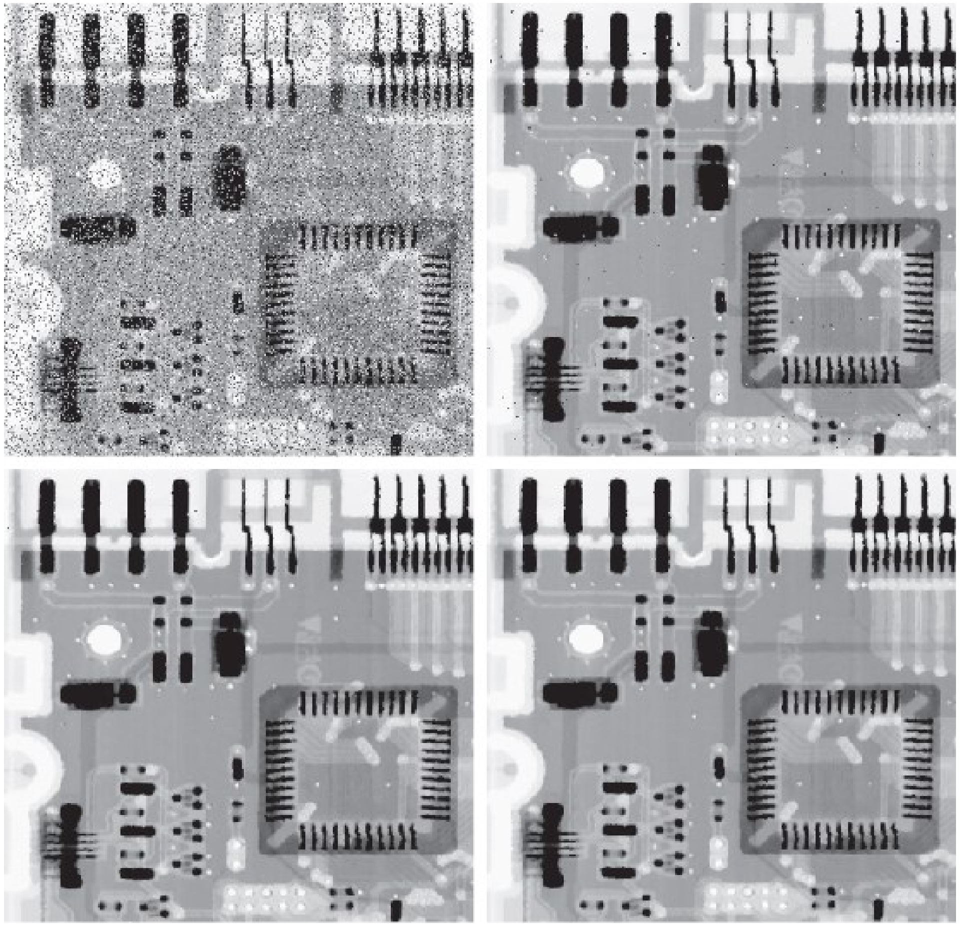
$$\hat{f}(x,y) = \frac{1}{mn-d} \sum_{(s,t)\in S_{xy}} g_r(s,t)$$

- 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 0.1

1 pass with a 3x3 median



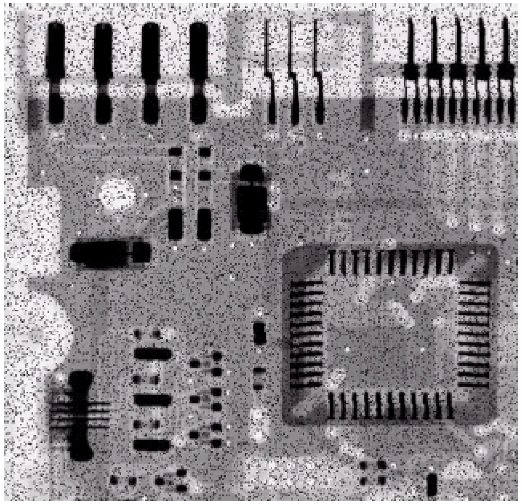
2 passes with a 3x3 median

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



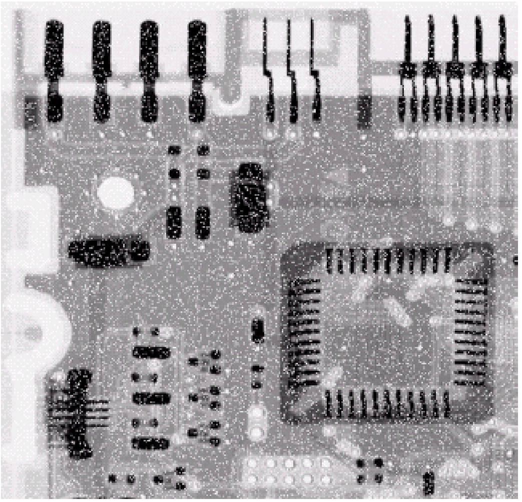
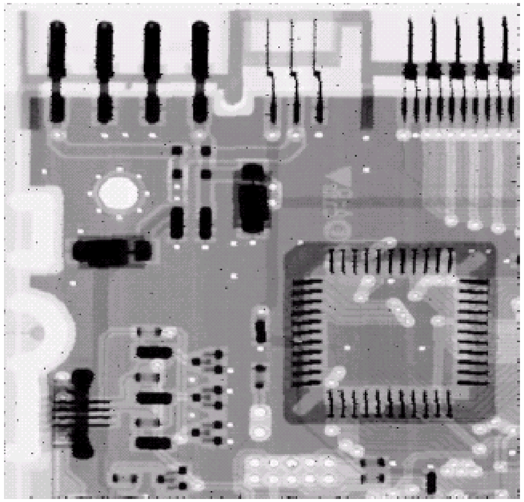
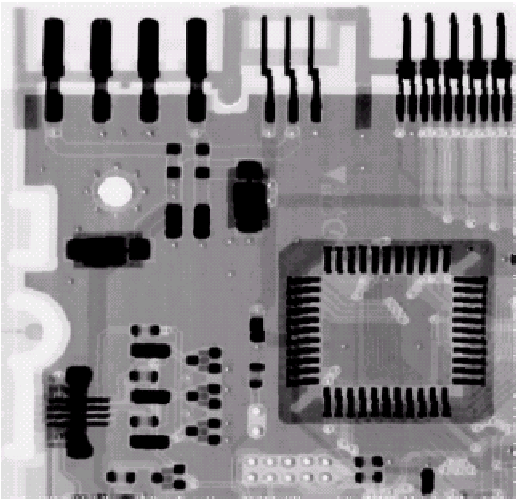


Image corrupted by Salt noise

Filtering above with a 3x3 Max Filter

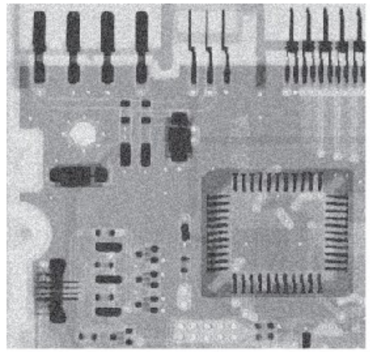




Filtering above with a 3x3 Min Filter

Noise Removal Examples (cont…)

Image corrupted by uniform noise



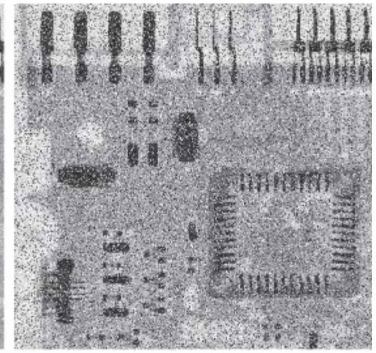
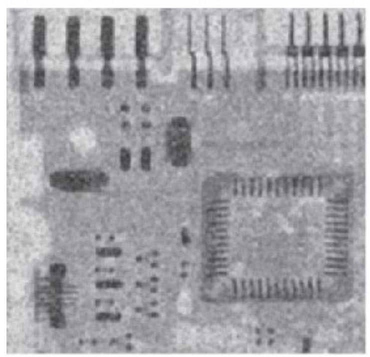
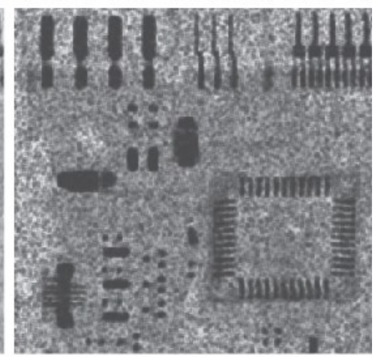


Image further corrupted by Salt and Pepper noise

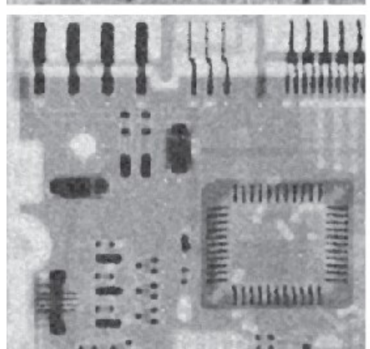
Filtering by a 5x5 Arithmetic Mean Filter

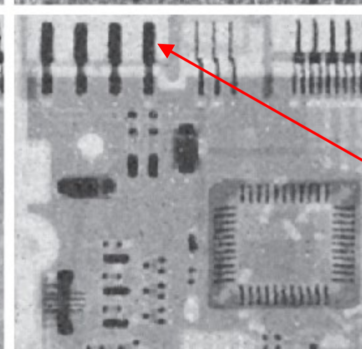




Filtering by a 5x5 Geometric Mean Filter

Filtering by a 5x5 Median Filter





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

More smoothing than the simple median

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 statistical filter**
  - **adaptive median filter.**

Adaptive Statistical Filter

- Applied in a neighbourhood S_{xy} around (x, y).
- $g(x, y)$ is the degraded image pixel.
- σ_η is the noise standard deviation
- $\bar{z}_{S_{xy}}$ is the mean value in S_{xy}
- σ_{xy} is the standard deviation in S_{xy}

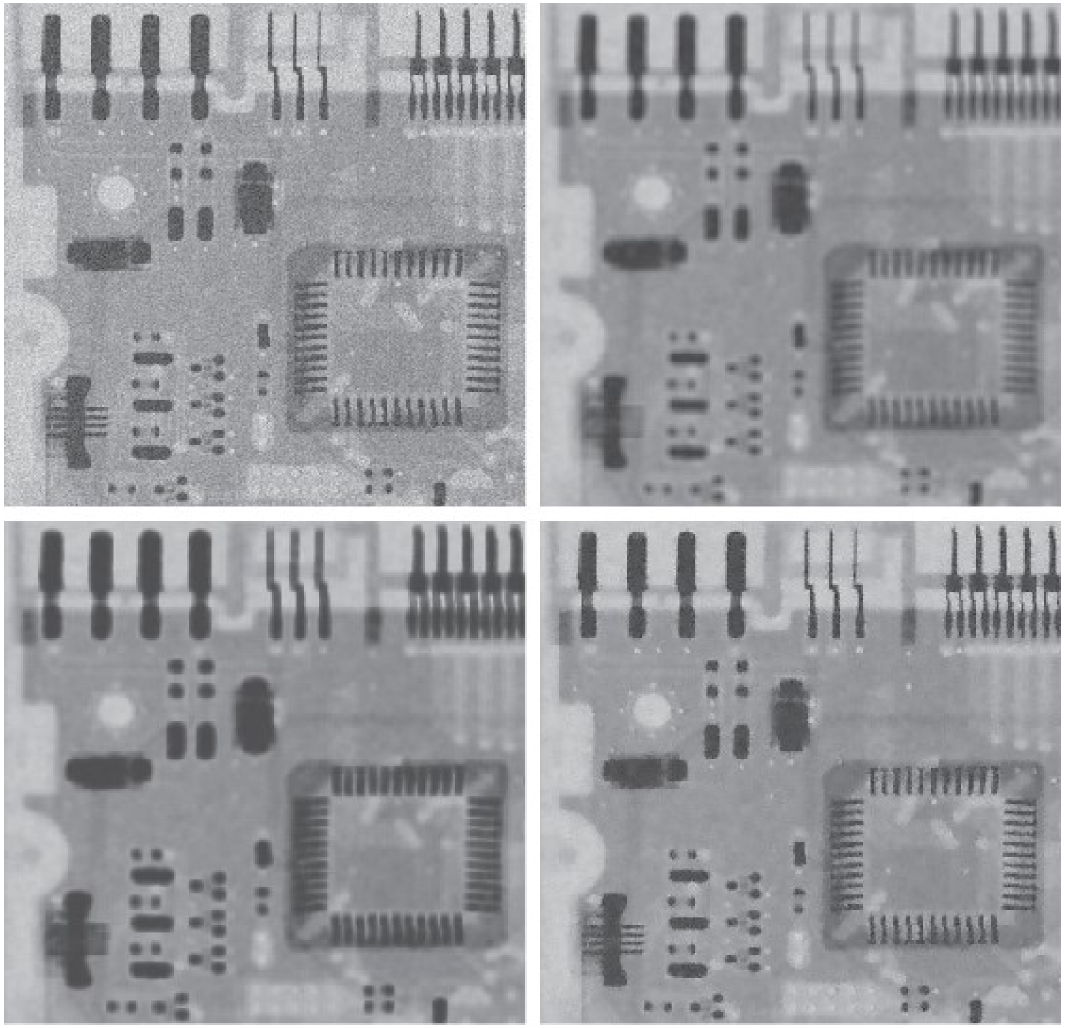
$$\hat{f}(x, y) = g(x, y) - \frac{\sigma_\eta^2}{\sigma_{xy}^2}\left[g(x, y) - \bar{z}_{S_{xy}}\right]$$

Adaptive Statistical Filter (cont.)

$$\hat{f}(x,y) = g(x,y) - \frac{\sigma_\eta^2}{\sigma_{xy}^2}\left[g(x,y) - \bar{z}_{S_{xy}}\right]$$

- When the local standard deviation is higher than the noise standard deviation the filter returns a value close to $g(x,y)$ (e.g. edges tht should be kept).
- When the local standard deviation is close to the noise standard deviation the filter returns a value close to the local average.
- We must estimate σ_η.

Adaptive Statistical Filter (cont.)



a b
c d

FIGURE 5.13
(a) Image corrupted by additive Gaussian noise of zero mean and a variance of 1000. (b) Result of arithmetic mean filtering. (c) Result of geometric mean filtering. (d) Result of adaptive noise-reduction filtering. All filters used were of size 7×7.

Similar results but better contrast

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 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 centered 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:

$$A1 = z_{med} - z_{min}$$
$$A2 = z_{med} - z_{max}$$
$$\text{If } A1 > 0 \text{ and } A2 < 0, \text{ Go to stage B}$$
$$\text{Else increase the window size}$$
$$\text{If window size} \leq S_{max} \text{ repeat stage A}$$
$$\text{Else output } z_{med}$$

- 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}

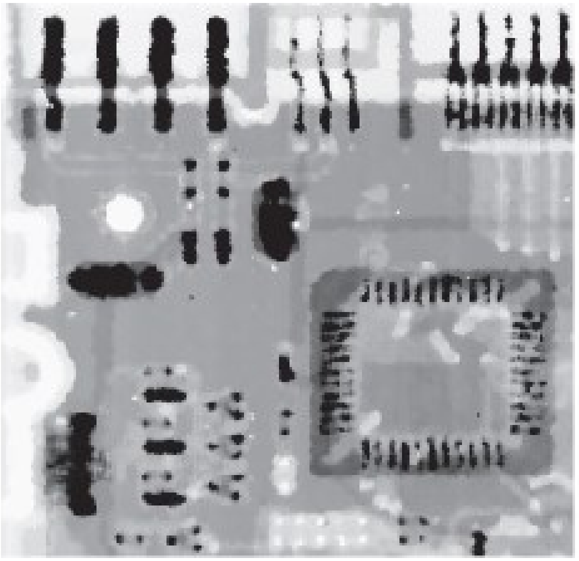
Stage B determines if the pixel value at (x, y), that is z_{xy}, is an impulse or not (black or white).

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

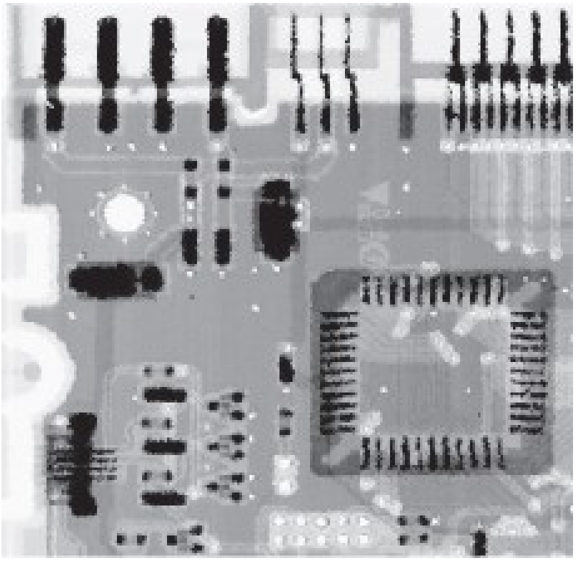
If it is an impulse the algorithm outputs the median z_{med}.

Adaptive Median 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

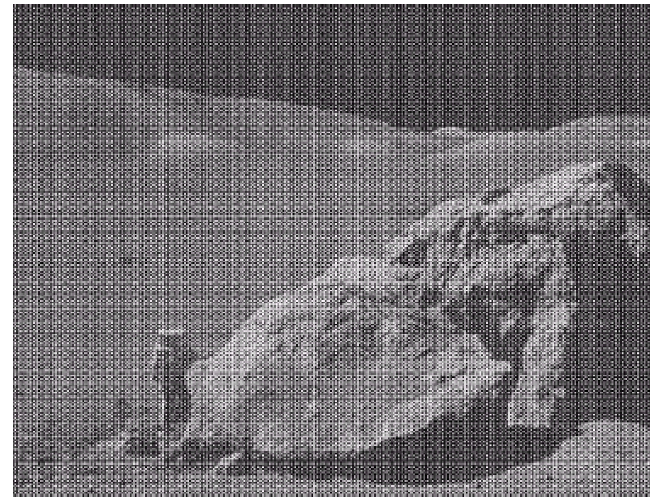


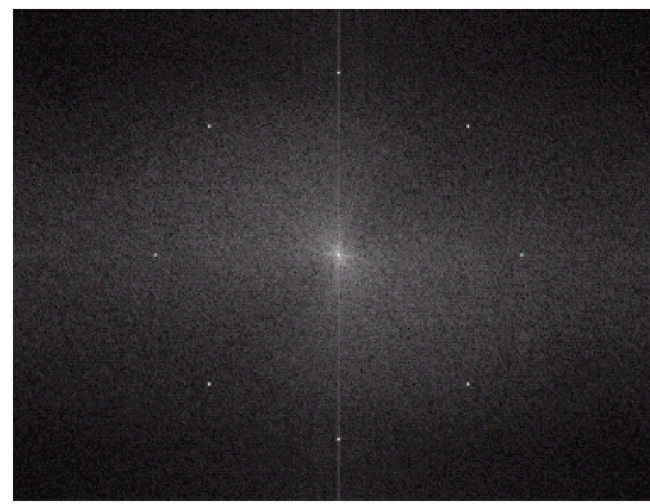
Result of adaptive median filtering with $S_{max} = 7$

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.





Notch Filters

- Removing periodic noise form an image involves removing a particular range of frequencies from that image.
- Rejects frequencies in a predefined neighbourhood around a center frequency.

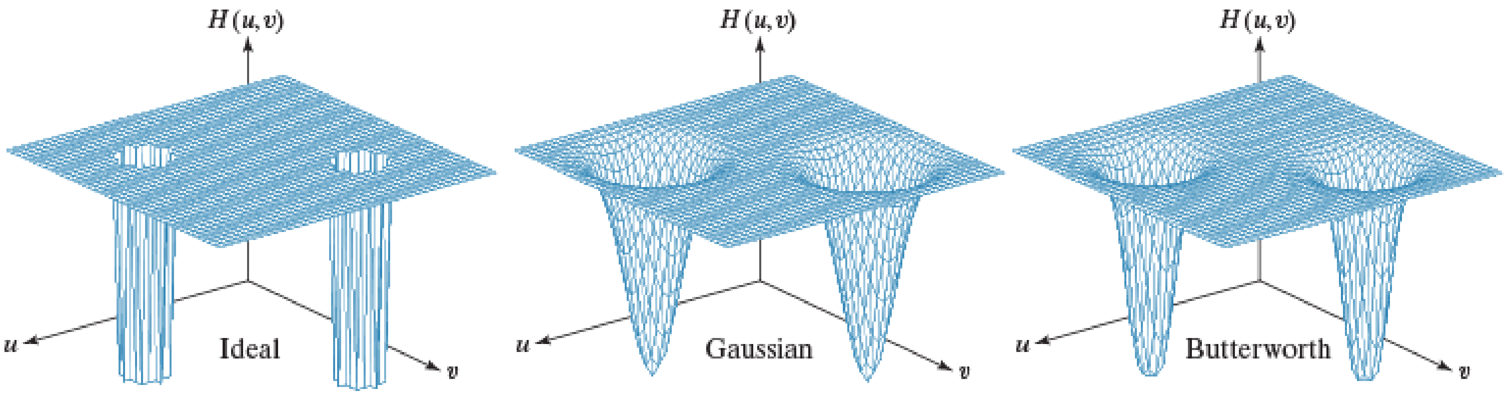


FIGURE 5.15 Perspective plots of (a) ideal, (b) Gaussian, and (c) Butterworth notch reject filter transfer functions.

Bandreject/stop filters (ζωνοφρακτικά φίλτρα)

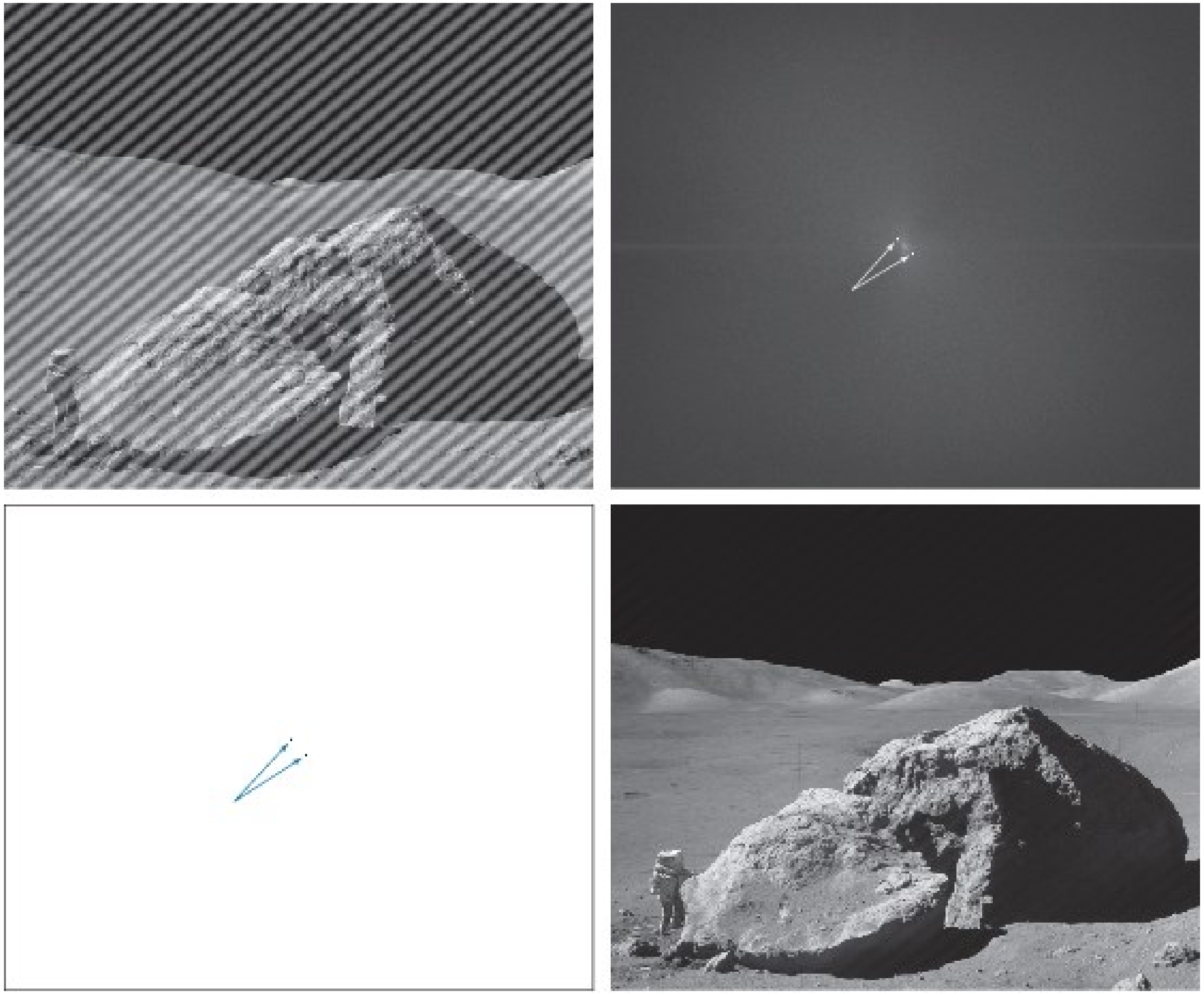
- Removal of periodic noise involves eliminating a specific range of frequencies from the image.
- This is typically achieved using bandstop filters. Conversely, bandpass filters are used to isolate specific frequency ranges.
- An ideal band reject filter is as follows:

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

Notch Filter Example

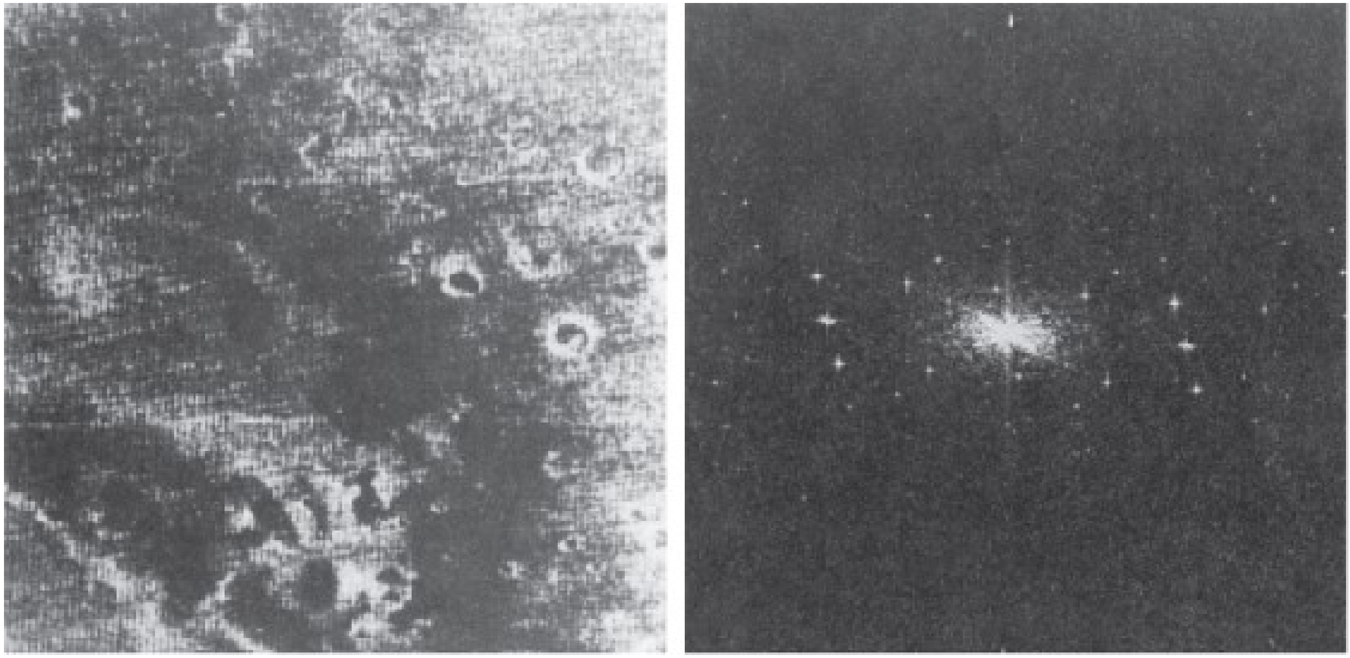
a b
c d

FIGURE 5.16
(a) Image corrupted by sinusoidal interference. (b) Spectrum showing the bursts of energy caused by the interference. (The bursts were enlarged for display purposes.) (c) Notch filter (the radius of the circles is 2 pixels) used to eliminate the energy bursts. (The thin borders are not part of the data.) (d) Result of notch reject filtering. (Original image courtesy of NASA.)



Optimum Notch Filtering (Αποκατάσταση μέσω ζυγισμένου φίλτρου εγκοπής)

- 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.)

Optimum Notch Filtering (cont.)

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

a b

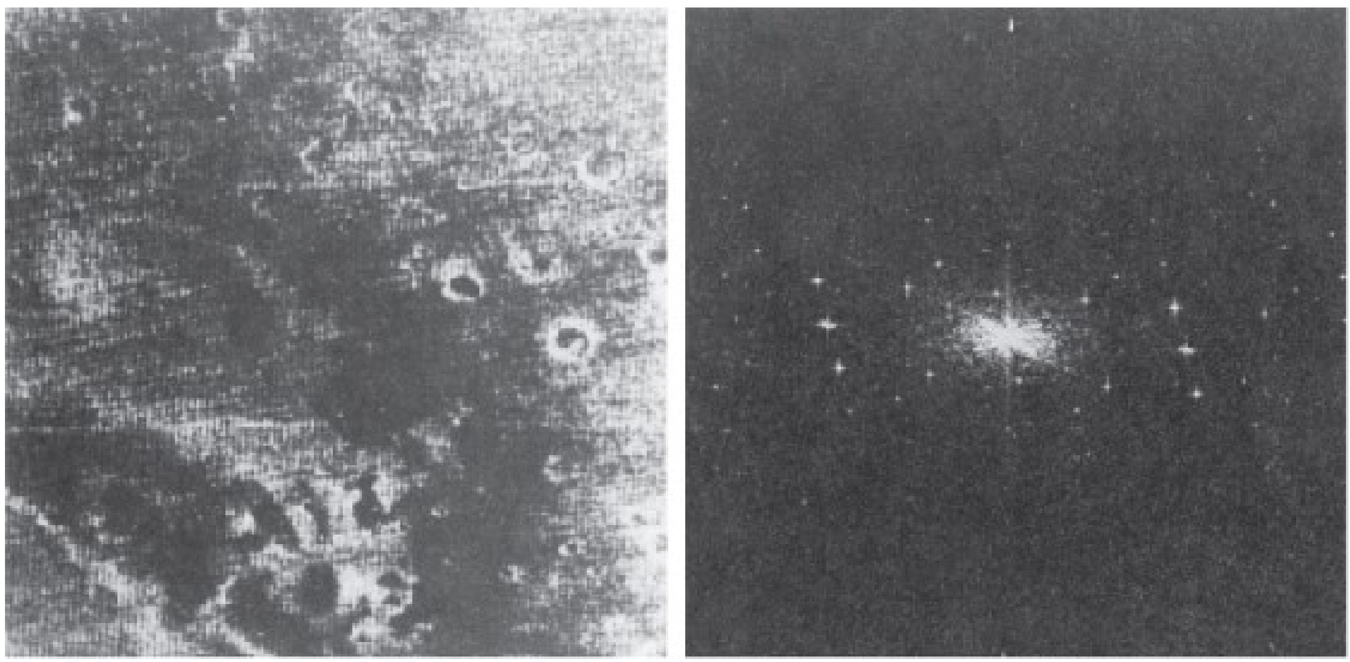


FIGURE 5.20 (a) Image of the Martian terrain taken by Mariner 6. (b) Fourier spectrum showing periodic interference. (Courtesy of NASA.)

Optimum Notch Filtering (cont.)

- A noise estimate in the DFT domain:

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

- In the spatial domain:

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

- Image estimate:

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

Optimum Notch Filtering (cont.)

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

- How we select a proper w;
- Compute the weight minimizing the variance over a local neighbourhood of the estimated image centered at (m,n):

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

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 $\hat{f}(m,n)$ in $\sigma^2(m,n)$: yields:

Optimum Notch Filtering (cont.)

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

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

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

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

Optimum Notch Filtering (cont.)

- To minimize the variance:

$$\frac{\partial \sigma^2(m,n)}{\partial w(m,n)} = 0$$

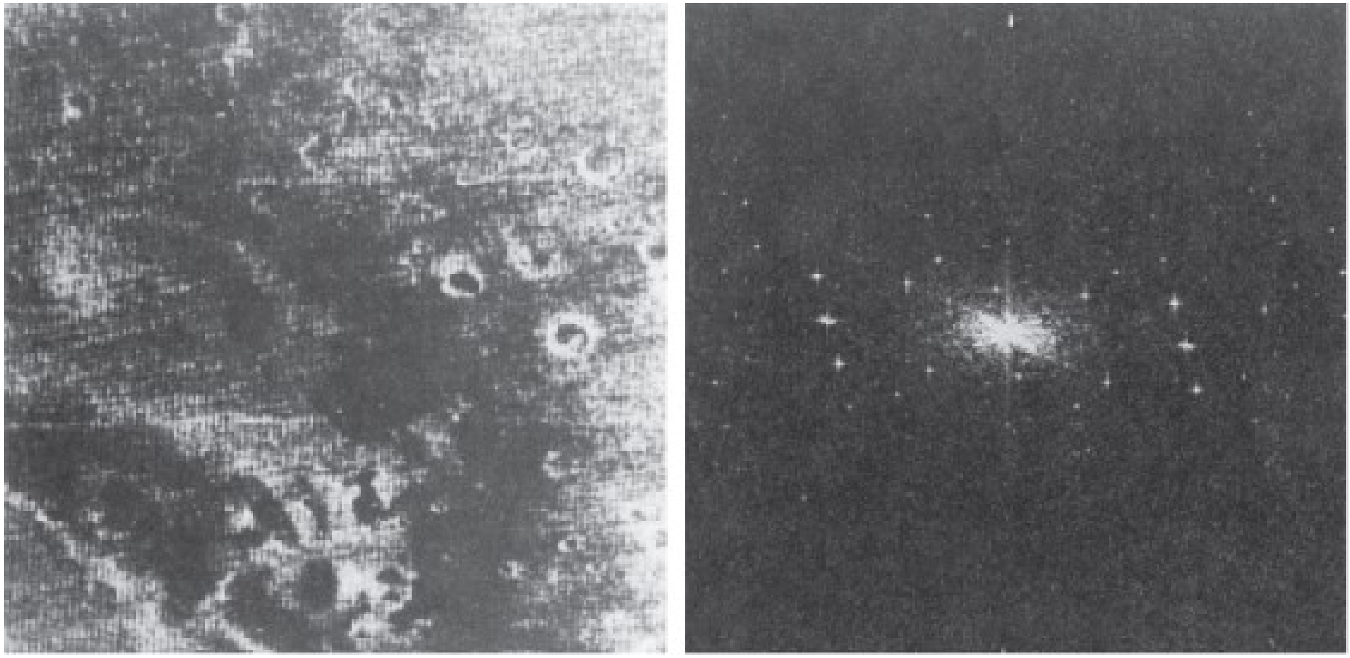
yielding the closed-form solution:

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

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

Optimum Notch Filtering (cont.)

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

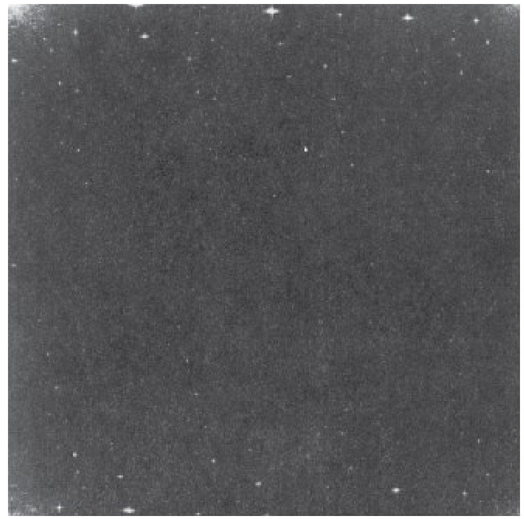


a b

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

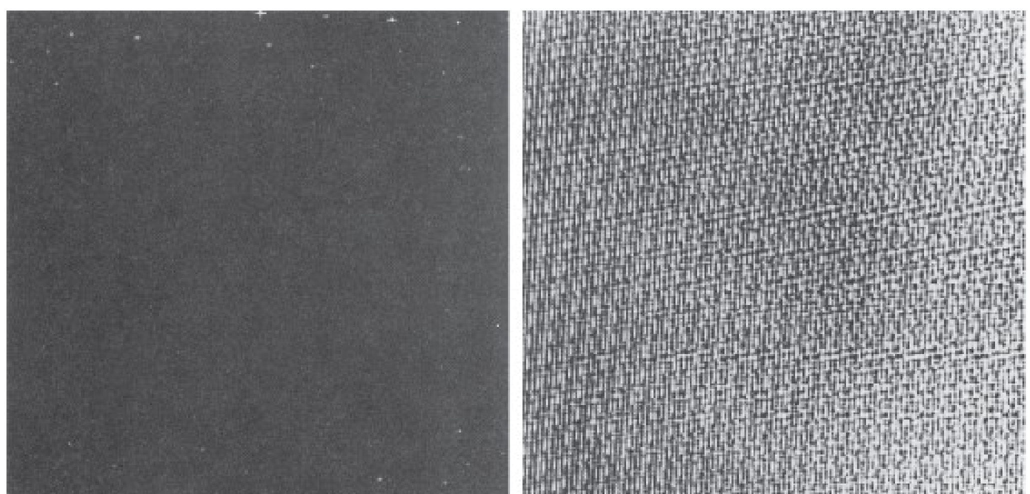
Optimum Notch Filtering (cont.)

FIGURE 5.21 Uncentered Fourier spectrum of the image in Fig. 5.20(a). (Courtesy of NASA.)



a b

FIGURE 5.22 (a) Fourier spectrum of $N(u, v)$, and (b) corresponding spatial noise interference pattern, $\eta(x, y)$. (Courtesy of NASA.)



Optimum Notch Filtering (cont.)

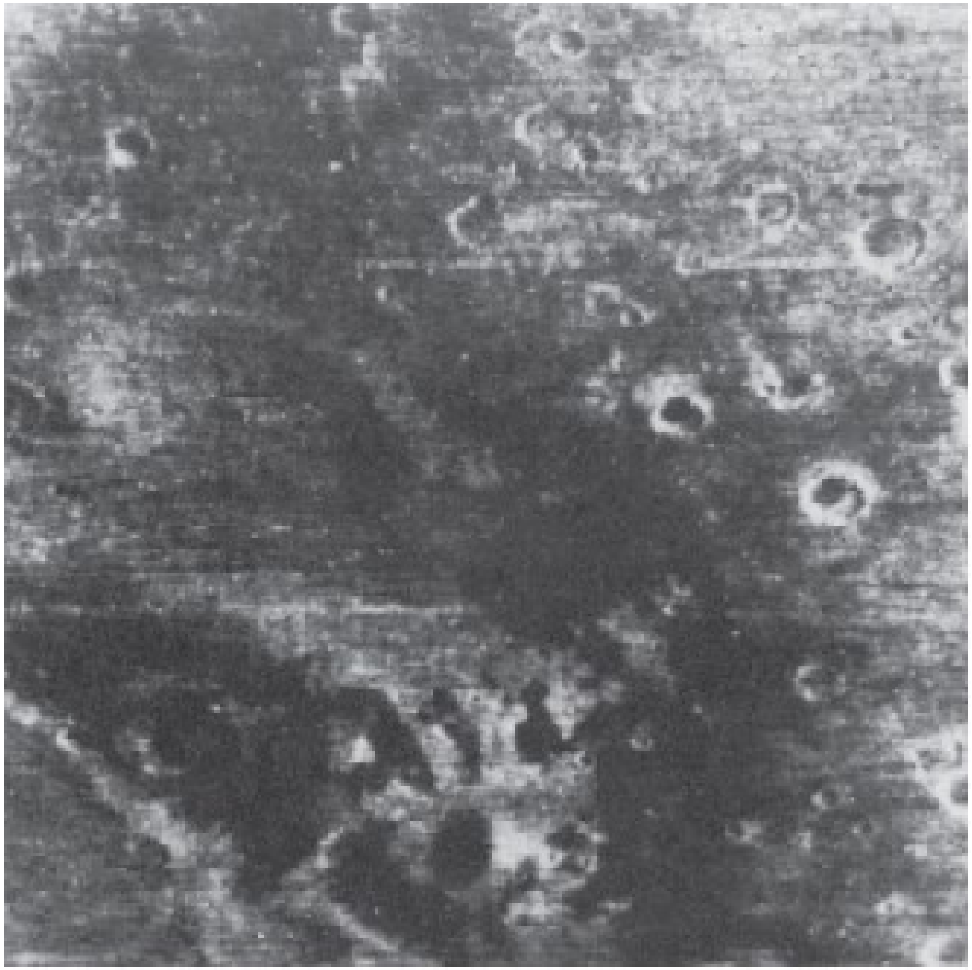


FIGURE 5.23 Restored image. (Courtesy of NASA.)

Synopsis

We examined:

- **Image restoration in the presence of noise**
- **Noise models**
- Noise removal by **filtering** in the **spatial** and **frequency** domain
- In the next lesson we will talk about **linear image restoration with no noise**

- Upcoming Project announcement – in the next few days