CHAPTER 5

IMAGE RESTORATION

CHAPTER 5: IMAGE RESTORATION

- **The ultimate goal is to improve an image in some predefined sense.**
- **Restoration**
  - **The given image is degraded.**
  - **Degradation is modeled.**
  - **Inverse process is applied.**
- **No focus on sensor, digitizer, and display degradations.**
- **Some restoration techniques are best formulated in the spatial domain while others are better suited for the frequency domain.**
  - **Spatial domain: additive noise.**
  - **Frequency domain: blurring**

A MODEL OF IMAGE DEGRADATION/RESTORATION PROCESS

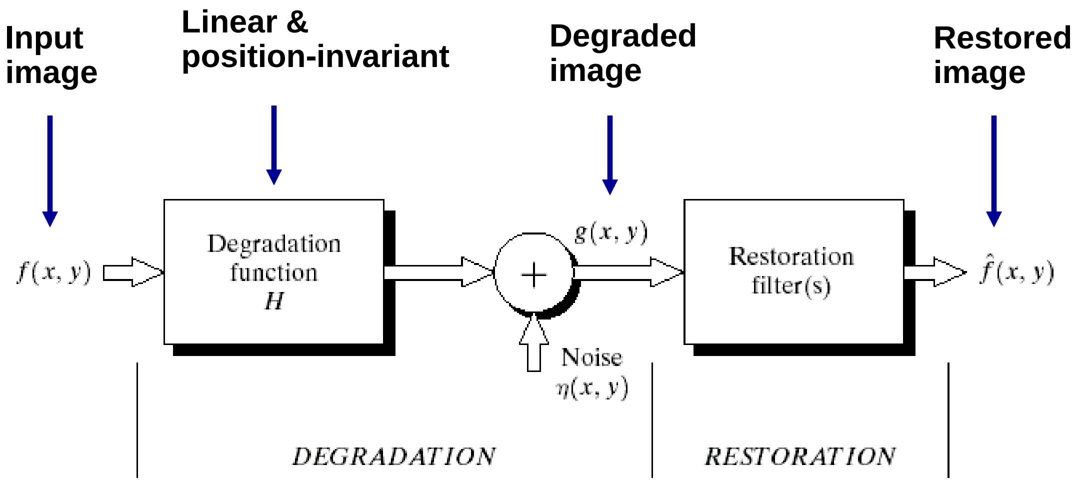


FIGURE 5.1 A model of the image degradation/restoration process.

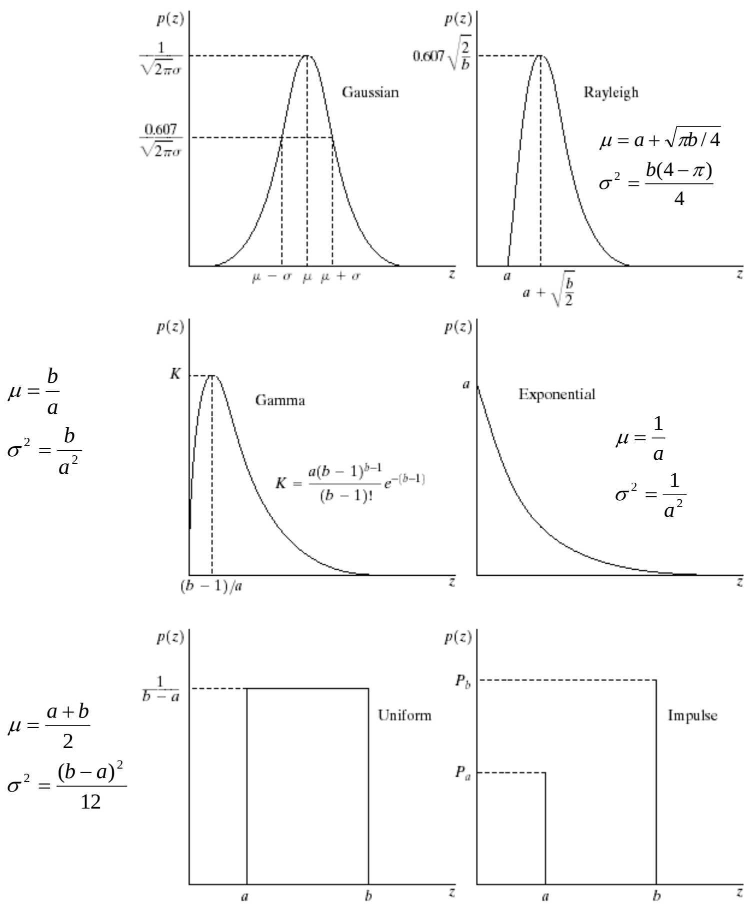
$$g(x,y) = h(x,y) * f(x,y) + \eta(x,y)$$

$$G(u,v) = H(u,v)F(u,v) + N(u,v)$$

NOISE MODELS

- **Principle sources of noise in digital images arise during acquisition and transmission.**
  - **Acquisition: performance of image sensors is affected by a variety of factors (environmental conditions, etc.) and by the quality of the sensing elements.**
  - **Transmission: images are corrupted due to interference in the channel.**
- **Spatial properties of noise**
- **Frequency properties of noise**
  - **White noise: Fourier spectrum of noise is constant.**
- **Assumption: noise is independent of spatial coordinates.**
  - **Partially invalid.**
  - **No correlation between pixel values and noise.**
  - **Exception: spatially periodic noise.**

NOISE PROBABILITY DENSITY FUNCTIONS



We are concerned with the statistical behavior of the gray-level values of the noise component of the model.

These values may be considered random variables characterized by a probability density function (PDF).

Most common PDFs:

1. **Gaussian (normal) noise**
2. **Rayleigh noise**
3. **Erlang (Gamma) noise**
4. **Exponential noise**
5. **Uniform noise**
6. **Impulse (salt-and-pepper) noise**

MODELING A BROAD RANGE OF NOISE CORRUPTIONS

- The 6 PDFs are useful tools.
- Gaussian noise arises in an image due to factors such as electronic circuit noise and sensor noise due to poor illumination and/or high temperature.
- Rayleigh density is helpful in characterizing noise phenomena in range imaging.
- Gamma and exponential densities find application in laser imaging.
- Impulse noise if found in situations where quick transients, such as faulty switching, take place during imaging.
- Uniform density may be the least descriptive of practical situations but it is quite useful as the basis for numerous random number generators.

A TEST PATTERN

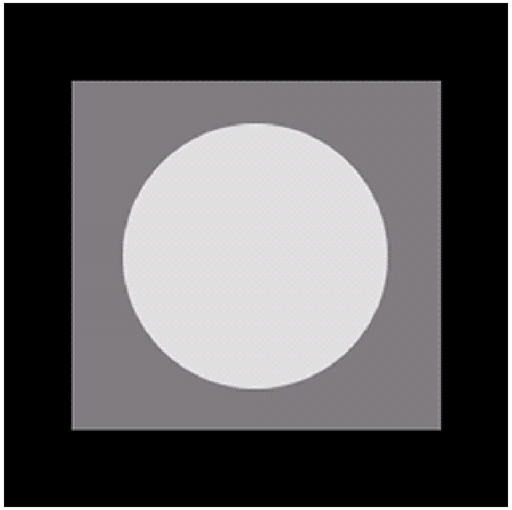


FIGURE 5.3 Test pattern used to illustrate the characteristics of the noise PDFs shown in Fig. 5.2.

composed of simple, constant areas that span the gray scale from black to near white in 3 increments.

NOISY IMAGES AND THEIR HISTOGRAMS – GAUSSIAN, RAYLEIGH, AND GAMMA

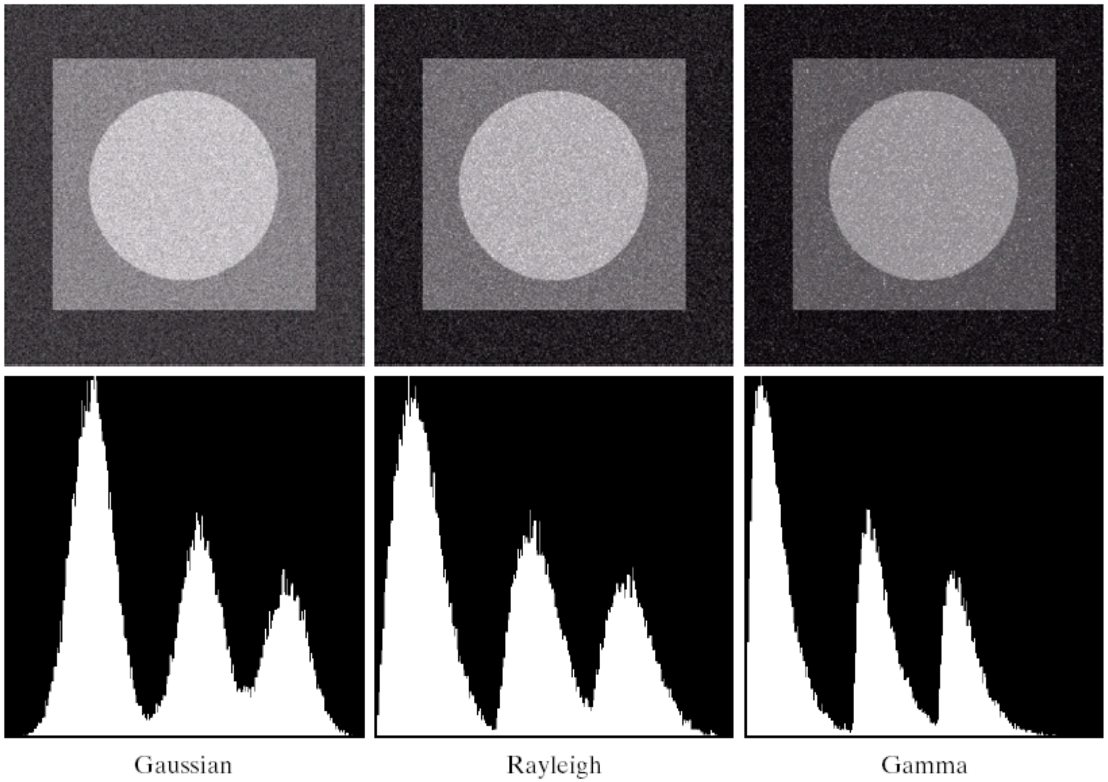


FIGURE 5.4 Images and histograms resulting from adding Gaussian, Rayleigh, and gamma noise to the image in Fig. 5.3.

The parameters of the noise were chosen in each case so that the histogram corresponding to the 3 gray levels in the test pattern would start to merge.

NOISY IMAGES AND THEIR HISTOGRAMS – EXPONENTIAL, UNIFORM, AND IMPULSE

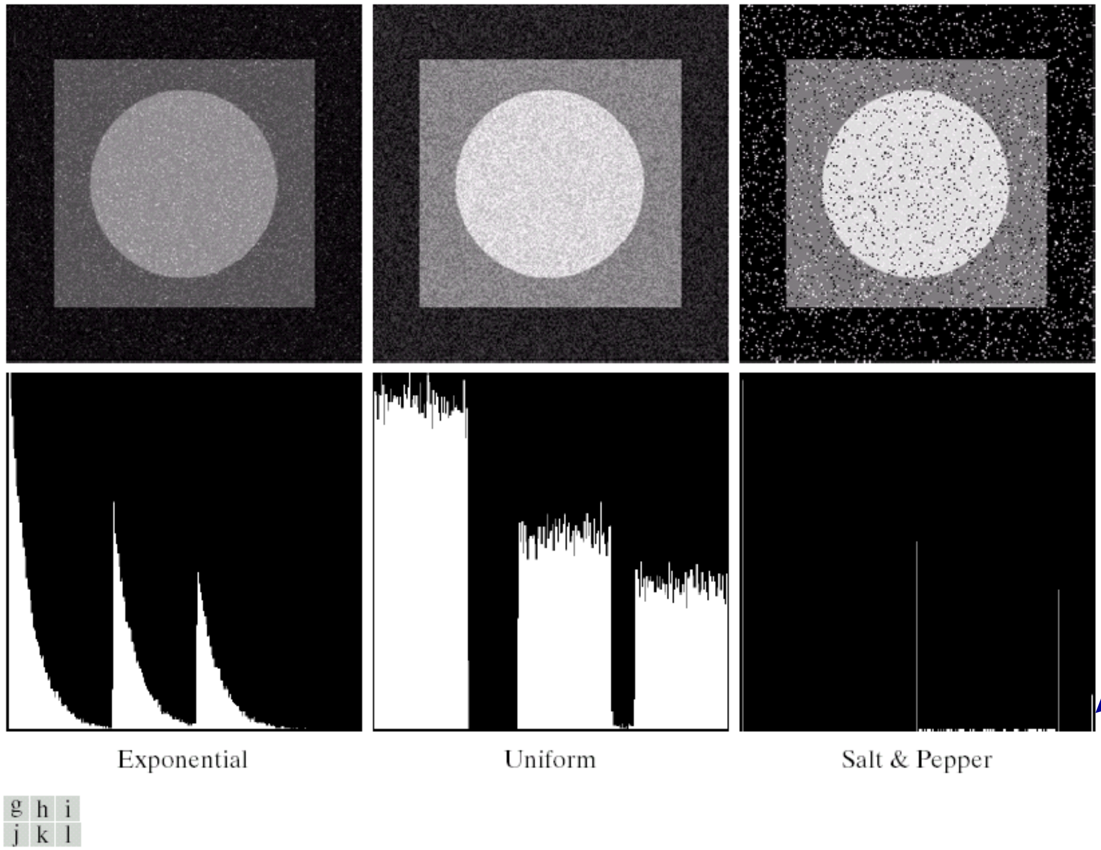
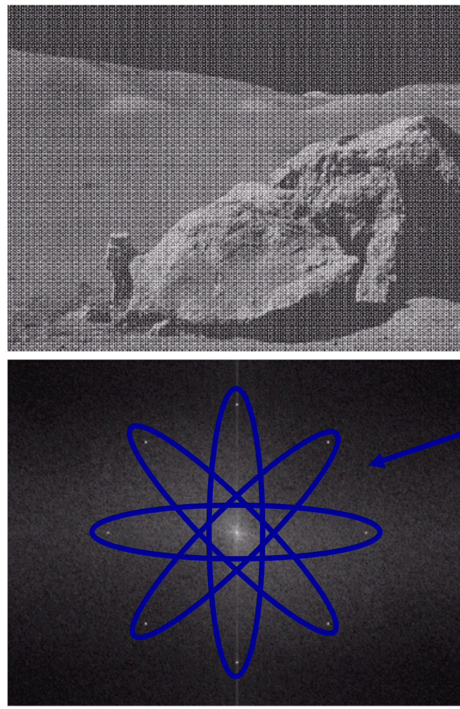


FIGURE 5.4 *(Continued)* Images and histograms resulting from adding exponential, uniform, and impulse noise to the image in Fig. 5.3.

The parameters of the noise were chosen in each case so that the histogram corresponding to the 3 gray levels in the test pattern would start to merge.

Extra peak: the noise components were pure black and white, and the lightest component of the test pattern is light gray.

PERIODIC NOISE



a
b

FIGURE 5.5 (a) Image corrupted by sinusoidal noise. (b) Spectrum (each pair of conjugate impulses corresponds to one sine wave). (Original image courtesy of NASA.)

Image is severely corrupted by spatial sinusoidal noise of various frequencies.

The Fourier Transform of a pure sinusoid is a pair of conjugate impulses located at the conjugate frequencies of the sine wave.

If the amplitude of a sine wave in the spatial domain is strong enough, we would see in the spectrum of the image a pair of impulses for each sine wave in the image.

Periodic noise arises typically from electrical or electromechanical interference during acquisition.

This is the only spatially dependent noise considered here.

ESTIMATION OF NOISE PARAMETERS

- **Periodic noise**
  - Inspection of the Fourier spectrum
  - Inspection of the image (possible only in simple cases)
  - Automated analysis
    - Noise spikes are exceptionally pronounced.
    - Some knowledge is available about the general location of the frequency components.
- **Noisy PDFs**
  - Parameters may be partially known from sensor specs
  - Imaging system available
    - Capture a set of images of flat environments.
  - Images are available
    - Crop small patches of reasonably constant gray level.
    - Obtain the histogram.
    - Compute mean and variance.
    - Gaussian PDF: Completely determined by the mean and variance.
    - Impulse noise: actual probability of occurrence of white and black pixels is needed.
    - Others: Use the mean and variance to solve for *a* and *b*.

SPATIAL DOMAIN FILTERING FOR ADDITIVE NOISE

$$g(x,y) = f(x,y) + \eta(x,y)$$ ← **additive noise**

$$G(u,v) = F(u,v) + N(u,v)$$

Mean filters

Arithmetic mean filter
Geometric mean filter
Harmonic mean filter
Contraharmonic mean filter

Order-statistics filters

Median filter
Max & min filters
Midpoint filter
Alpha-trimmed mean filter

Adaptive filters

Adaptive local noise reduction filter
Adaptive median filter

MEAN FILTERS

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

Arithmetic mean

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

Geometric mean
(comparable to arithmetic mean but tends to lose less image detail)

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

Harmonic mean
(works well for salt noise but fails for pepper noise. OK for other types of noise as well)

Q=0 ⇒ arithmetic mean
Q=-1 ⇒ harmonic mean

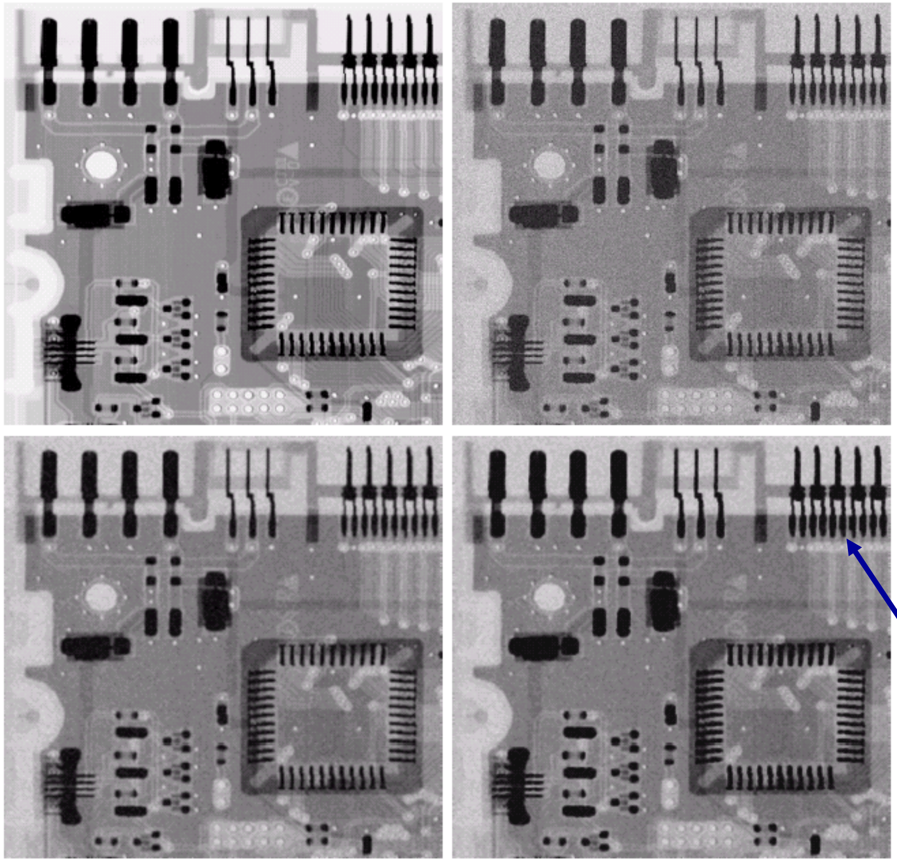
$$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}}$$

Order of the filter

Contraharmonic mean
(+Q: eliminates pepper noise
-Q: eliminates salt noise
not simultaneously!)

ARITHMETIC AND GEOMETRIC MEAN FILTERS

mean=0, variance=400



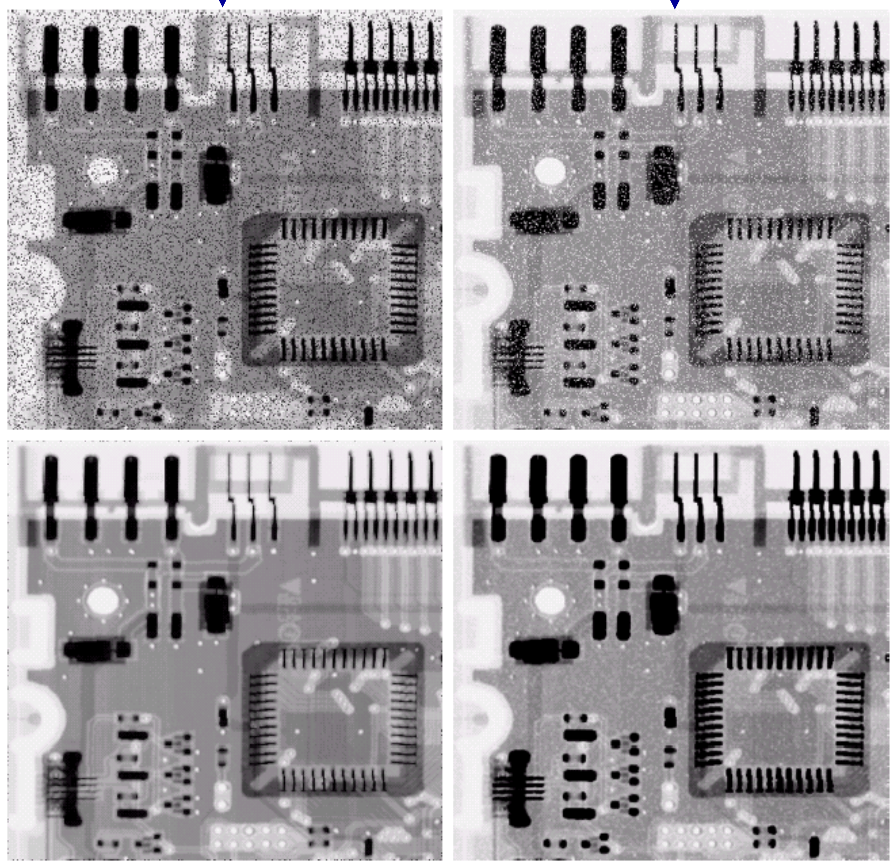
a	b
c	d

FIGURE 5.7 (a) X-ray image. (b) Image corrupted by additive Gaussian noise. (c) Result of filtering with an arithmetic mean filter of size 3 × 3. (d) Result of filtering with a geometric mean filter of the same size. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Geometric mean filter did not blur the image as much as the arithmetic mean filter

SPATIAL FILTERING FOR ADDITIVE NOISE

pepper noise **salt noise**

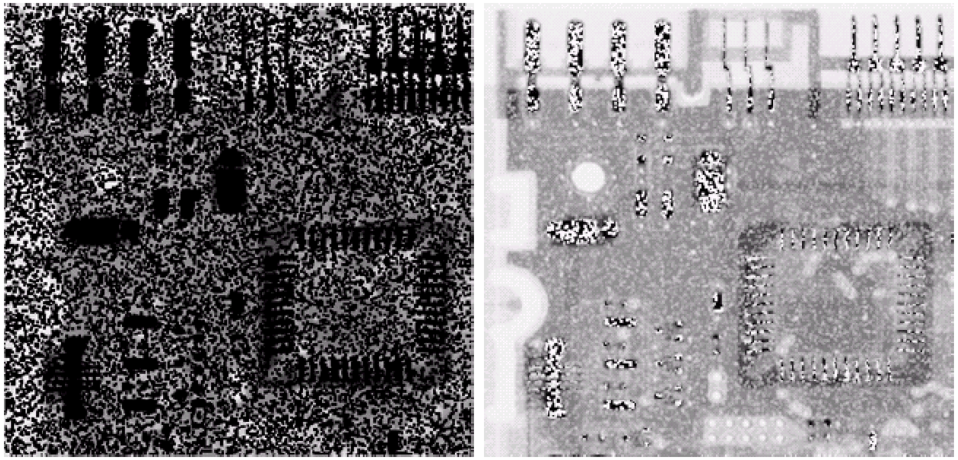


a b
c d

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 of order 1.5. (d) Result of filtering (b) with $Q = -1.5$.

Better job of cleaning the background at the expense of blurring the dark areas!

CONTRAHARMONIC FILTERING WITH THE WRONG SIGN



a b

FIGURE 5.9 Results of selecting the wrong sign in contraharmonic filtering. (a) Result of filtering Fig. 5.8(a) with a contraharmonic filter of size 3 × 3 and $Q = -1.5$. (b) Result of filtering 5.8(b) with $Q = 1.5$.

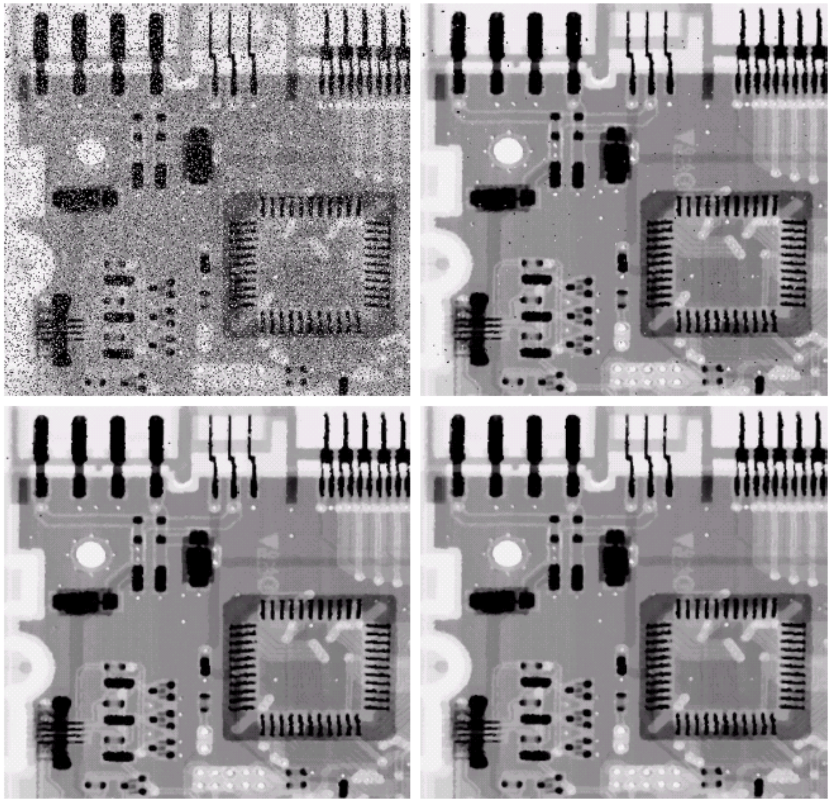
Wrong sign for Q

Wrong sign for Q

ORDER-STATISTICS FILTERS

Effective for bipolar and unipolar impulse noise	$\hat{f}(x,y) = \underset{(s,t)\in S_{xy}}{\text{median}} \{g(s,t)\}$	Median filter
Useful for finding the brightest points in an image	$\hat{f}(x,y) = \max_{(s,t)\in S_{xy}} \{g(s,t)\}$	Max filter
Useful for finding the darkest points in an image	$\hat{f}(x,y) = \min_{(s,t)\in S_{xy}} \{g(s,t)\}$	Min filter
Works best for randomly distributed noise	$\hat{f}(x,y) = \frac{1}{2}\left[\max_{(s,t)\in S_{xy}} \{g(s,t)\} + \min_{(s,t)\in S_{xy}} \{g(s,t)\}\right]$	Mid-point filter
d = 0 → arithmetic *d* = (mn-1)/2 → median Other *d*: Useful for multiple types of noise	$\hat{f}(x,y) = \frac{1}{mn-d} \sum_{(s,t)\in S_{xy}} g_r(s,t)$	Alpha-trimmed filter

3 PASSES OF MEDIAN FILTER FOR IMPULSE NOISE



a b
c d

FIGURE 5.10 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.1$. (b) Result of one pass with a median filter of size 3×3. (c) Result of processing (b) with this filter. (d) Result of processing (c) with the same filter.

1ST pass

Significant Improvement!

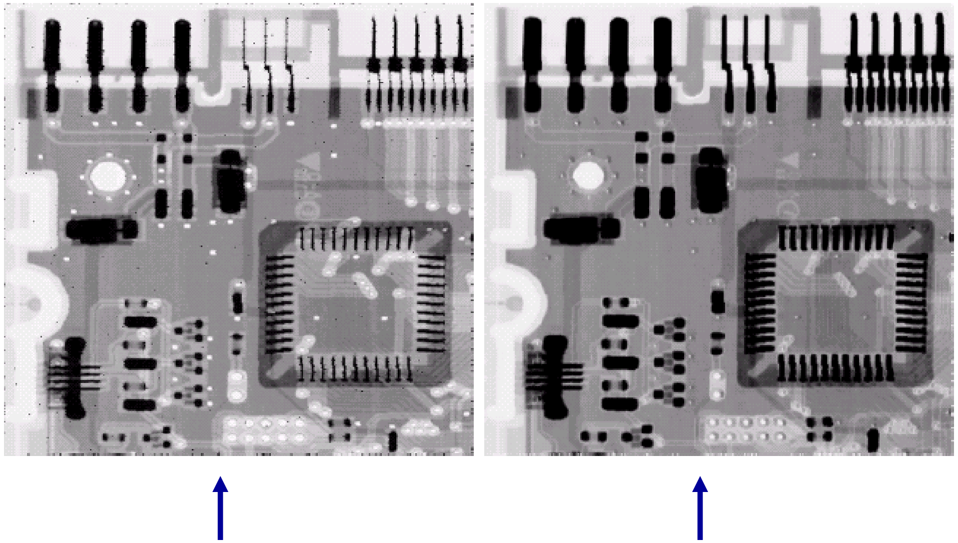
2nd pass

barely visible noise points!

3rd pass

# of passes should be as low as possible!

MAX & MIN FILTERS FOR PEPPER NOISE



a b

FIGURE 5.11 (a) Result of filtering Fig. 5.8(a) with a max filter of size 3 × 3. (b) Result of filtering 5.8(b) with a min filter of the same size.

Pepper noise is reasonably removed.

The filter also removed some dark points from the borders of the dark objects.

Better job!

The filter also removed some white points from the borders of the light objects.

REDUCTION OF NOISE WITH 4 TYPES OF FILTERS

Uniform noise (mean=0, variance=800) →

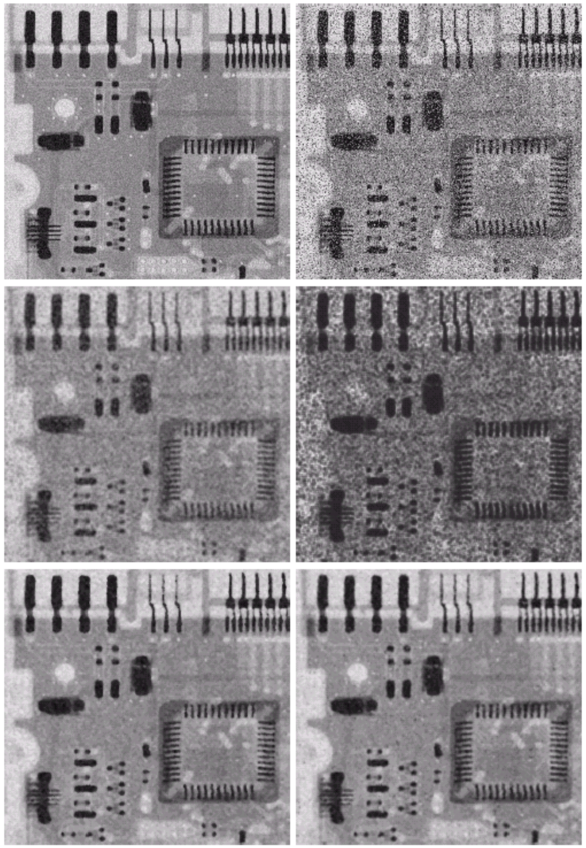
← **Additional salt-and-pepper noise** (P_a=P_b=0.1)

Arithmetic mean (no good) →

← **Geometric mean** (no good)

Median (much better) →

← **Alpha-trimmed mean** (better than median)



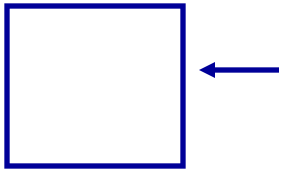
a b
c d
e f

FIGURE 5.12 (a) Image corrupted by additive uniform noise. (b) Image additionally corrupted by additive salt-and-pepper noise. Image in (b) filtered with a 5×5: (c) arithmetic mean filter; (d) geometric mean filter; (e) median filter; and (f) alpha-trimmed mean filter with $d = 5$.

ADAPTIVE FILTERS

- The filters discussed so far are applied to an image independent of how image characteristics vary from one point to another.
- 2 simple adaptive filters
  - Adaptive, local noise reduction filter
  - Adaptive median filter
- Their behavior changes based on statistics of the image inside the filter region.
- Advantage: superior performance
- Disadvantage: increase in filter complexity

ADAPTIVE, LOCAL NOISE REDUCTION FILTER



Local region S_{xy}

$g(x,y)$:	value of noisy image at (x,y)
σ_η^2:	variance of the noise
m_L:	local mean of pixels in S_{xy}
σ_L^2:	local variance of pixels in S_{xy}

Behavior of the filter:

1. $\sigma_\eta^2 = 0$ ⇒ $g(x,y)$ – trivial case with zero noise
2. $\sigma_L^2 >> \sigma_\eta^2$ ⇒ $\sim g(x,y)$ – edges should be preserved
3. $\sigma_L^2 \approx \sigma_\eta^2$ ⇒ m_L – local noise is reduced by averaging

Definition:

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

Analysis of σ_η^2:

Needs to be known or estimated (but we seldom have exact knowledge).
Tacit assumption: $\sigma_L^2 \geq \sigma_\eta^2$ (because $S_{xy} \subset g(x,y)$)
$\sigma_L^2 < \sigma_\eta^2$ ⇒ set ratio=1 (this makes the filter nonlinear but prevents meaningless results)
$\sigma_L^2 < \sigma_\eta^2$ ⇒ allow negative values and rescale (results in loss of dynamic range in the image)

COMPARISON OF ADAPTIVE FILTER WITH ARITHMETIC AND GEOMETRIC MEAN FILTERS

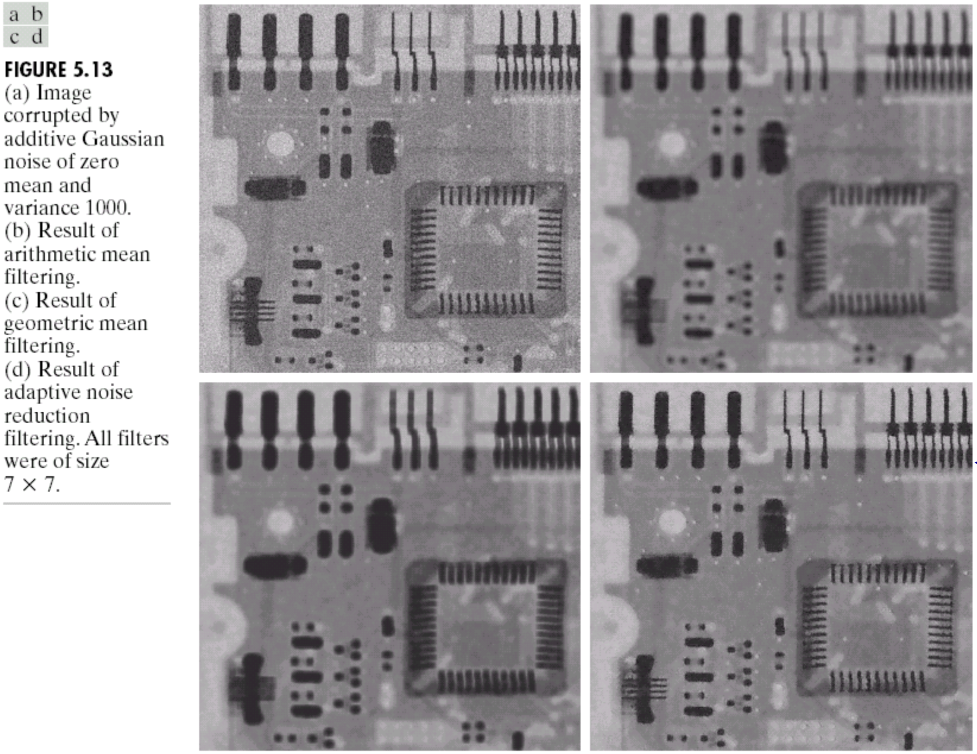


FIGURE 5.13 (a) Image corrupted by additive Gaussian noise of zero mean and variance 1000. (b) Result of arithmetic mean filtering. (c) Result of geometric mean filtering. (d) Result of adaptive noise reduction filtering. All filters were of size 7 × 7.

Best results:

Noise reduction is <u>comparable</u> but the restored image is much sharper!

ADAPTIVE MEDIAN FILTER

Z_{min} = min gray level value in S_{xy}
z_{max} = max gray level value in S_{xy}
z_{med} = median of gray levels in S_{xy}
z_{xy} = gray level at (x,y)
S_{max} = max allowed size of S_{xy}

The filter works in 2 levels:

Level *A*

$A1 = z_{\text{med}} - z_{\text{min}}$
$A2 = z_{\text{med}} - z_{\text{max}}$
If $A1 > 0$ and $A2 < 0$, goto level B
Else increase the window size
If window size ⊙ S_{max}, repeat level A
Else output z_{xy}

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}

3 main purposes:

1. To remove impulsive noise
2. To provide smoothing of other noise
3. To reduce distortion (e.g., excessive thinning or thickening of object boundaries)

A SIMPLE EXAMPLE ADAPTIVE MEDIAN FILTER

center point (15)

10	20	20
20	15	20
20	25	100

$z_{min} = 10$

$z_{max} = 100$

$z_{med} = 20$

$A1 = 20\text{-}10 = 10$
$A2 = 20\text{-}100 = -80$
$A1 > 0$ & $A2 < 0$: $z_{min} < z_{med} < z_{max}$ Hence, z_{med} cannot be an impulse.
Go to level B

$B1 = 15\text{-}10 = 5$
$B2 = 15\text{-}100 = -85$
$B1 > 0$ & $B2 < 0$: $z_{min} < z_{xy} < z_{max}$ Hence, z_{xy} cannot be an impulse.

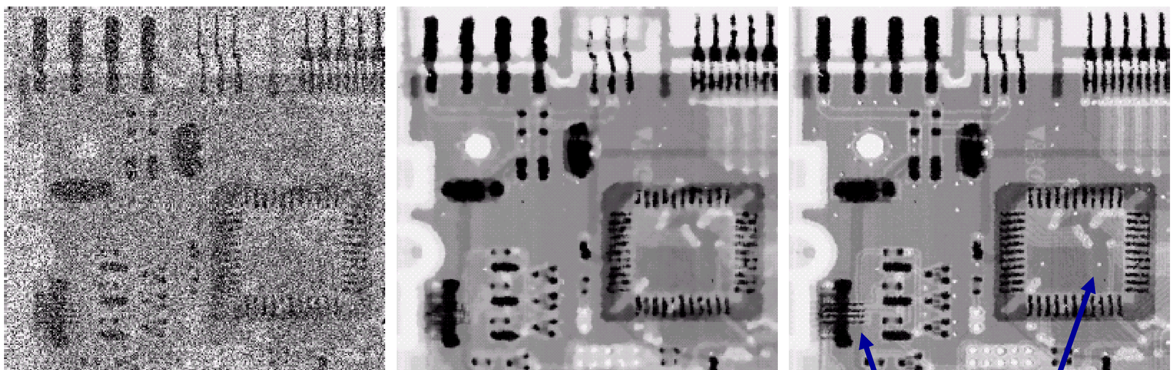
Output $z_{xy} = 15$ ← these intermediate-level points are not changed

($B1 > 0$ & $B2 < 0$) is false: $z_{xy} = z_{min}$ or $z_{xy} = z_{max}$

Output $z_{med} = 20$

COMPARISON OF MEDIAN AND ADAPTIVE MEAN FILTERS

Median filter **Adaptive median filter**



a b c

FIGURE 5.14 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.25$. (b) Result of filtering with a 7×7 median filter. (c) Result of adaptive median filtering with $S_{\max} = 7$.

Noise reduction is <u>comparable</u> but the filter preserved sharpness!

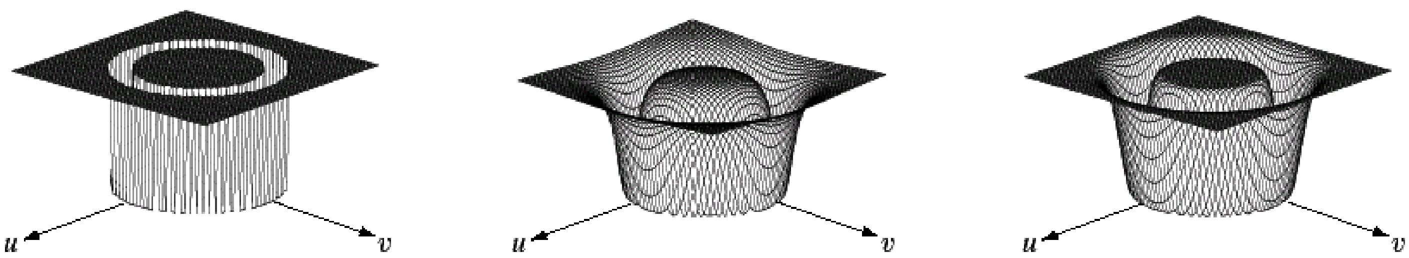
FREQUENCY DOMAIN FILTERING FOR PERIODIC NOISE

- **Bandreject filters: remove or attenuate a band of frequencies about the origin of the Fourier Transform.**

- **Bandpass filters: pass or strengthen a band of frequencies about the origin of the Fourier Transform .**

- **Notch filters: reject or pass frequencies in predefined neighborhoods about a center frequency.**

BANDREJECT FILTERS



a b c

FIGURE 5.15 From left to right, perspective plots of ideal, Butterworth (of order 1), and Gaussian bandreject filters.

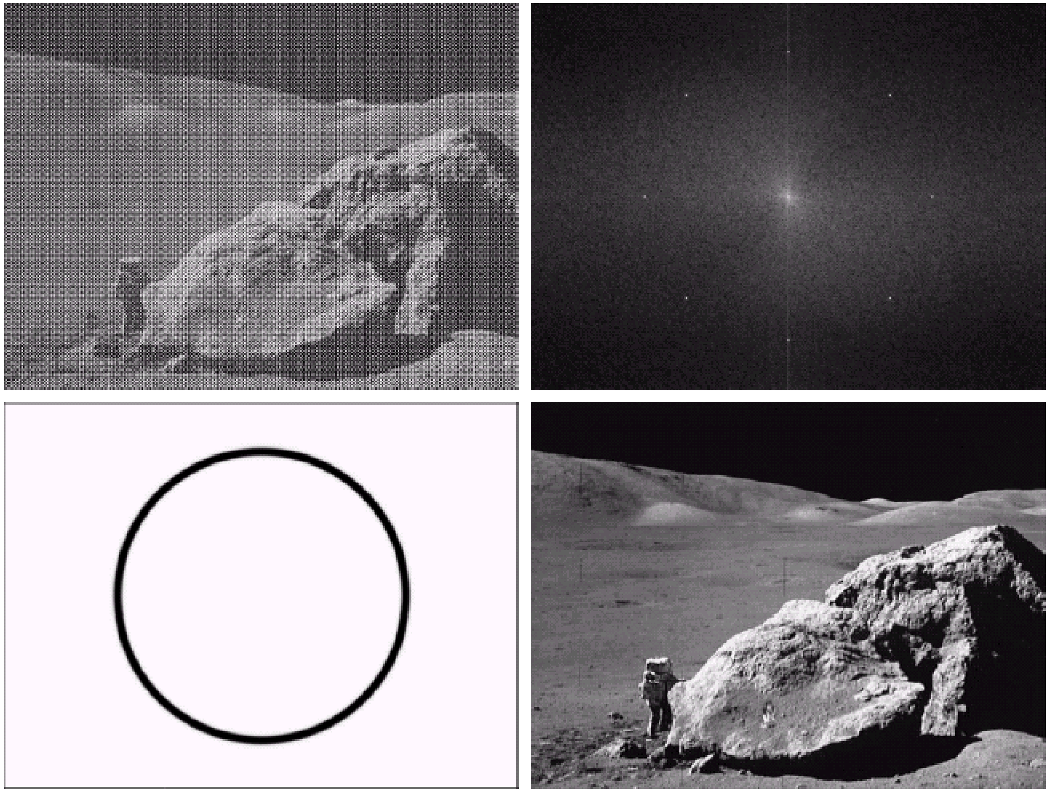
$$H(u,v)=\begin{cases} 1 & \text{if } D(u,v) < D_0 - \frac{W}{2} \\ 0 & \text{if } D_0 - \frac{W}{2} \le D(u,v) \le D_0 + \frac{W}{2} \\ 1 & \text{if } D(u,v) > D_0 - \frac{W}{2} \end{cases}$$

$$H(u,v)=\frac{1}{1+\left[\frac{D(u,v)W}{D^2(u,v)-D_0^2}\right]^{2n}}$$

$$H(u,v)=1-e^{-\frac{1}{2}\left[\frac{D^2(u,v)-D_0^2}{D(u,v)W}\right]^2}$$

A principal application: noise removal in situations where the general location of the noise components in the frequency domain is approximately known.

APPLICATION OF A BANDREJECT FILTER



a b
c d

FIGURE 5.16 (a) Image corrupted by sinusoidal noise. (b) Spectrum of (a). (c) Butterworth bandreject filter (white represents 1). (d) Result of filtering. (Original image courtesy of NASA.)

Restoration is evident!

Note that it would not be possible to get equivalent results by spatial domain filtering using small convolution masks!

BANDPASS FILTERS

$$H_{bp}(u,v) = 1 - H_{br}(u,v)$$

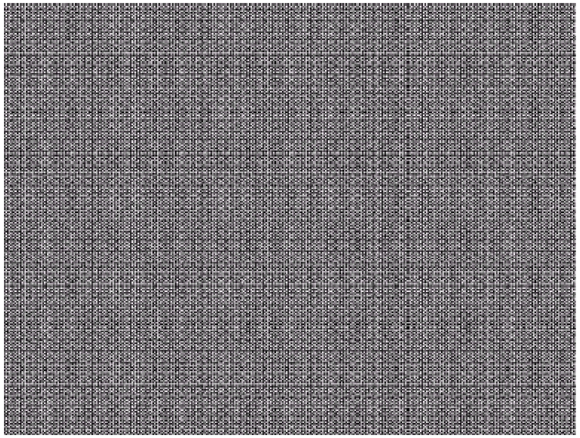
Performs the opposite operation of a bandreject filter.

Performing straight bandpass filtering on an image is not a common procedure because it generally removes too much image detail.

Bandpass filtering is quite useful in isolating the effect on an image of selected frequency bands.

APPLICATION OF A BANDPASS FILTER

FIGURE 5.17 Noise pattern of the image in Fig. 5.16(a) obtained by bandpass filtering.



Most image detail is lost but the remaining information is very useful as it shows a noise pattern that is close to that of the noise that corrupted the image!

Generated by:

- **using the bandpass filter corresponding to the bandreject filter in the previous example**
- **taking the inverse transform**

NOTCH FILTERS

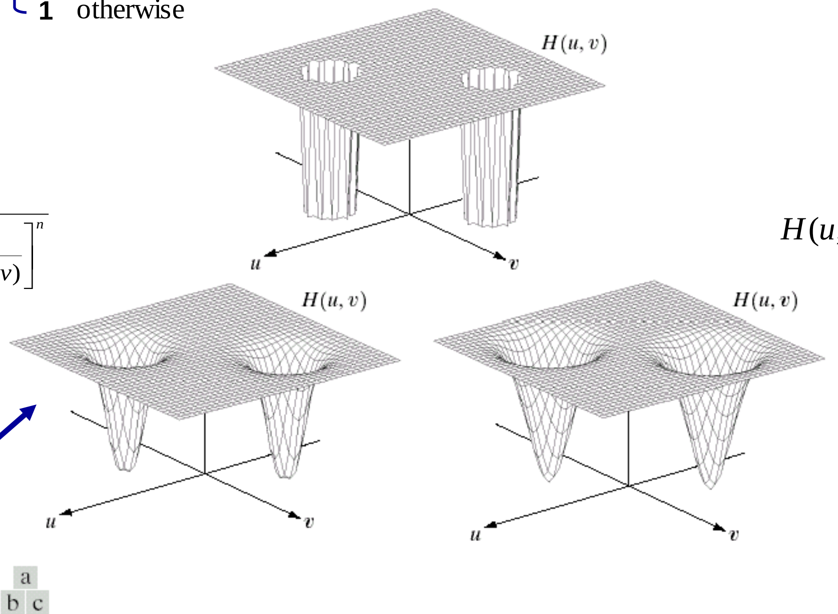
- Notch filters must appear in symmetric pairs about the origin in order to obtain meaningful results.
- The notch filter located at the origin is an exception.
- The # of pairs of notch filters is arbitrary.
- The shape of the notch areas is also arbitrary.
- Two classes
  - Notch reject filters
  - Notch pass filters
- Notch reject filters
  - Ideal notch reject filters
  - Butterworth notch reject filters
  - Gaussian notch reject filters
- Notch pass filters
  - Ideal notch pass filters
  - Butterworth notch pass filters
  - Gaussian notch pass filters

NOTCH REJECT FILTERS

$$H(u,v)=\begin{cases}\mathbf{0} & D_1(u,v)\le D_0 \text{ or } D_2(u,v)\le D_0\\ \mathbf{1} & \text{otherwise}\end{cases}$$

$$H(u,v)=\frac{1}{1+\left[\dfrac{D_0^2}{D_1(u,v)D_2(u,v)}\right]^n}$$

$$H(u,v)=1-e^{-\frac{1}{2}\left[\frac{D_1(u,v)D_2(u,v)}{D_0^2}\right]}$$



Note that these 3 filters become highpass filters if u_0=v_0=0.

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

$$D_1(u,v)=[(u-M/2-u_0)^2+(v-N/2-v_0)^2]^{1/2}$$

$$D_2(u,v)=[(u-M/2+u_0)^2+(v-N/2+v_0)^2]^{1/2}$$

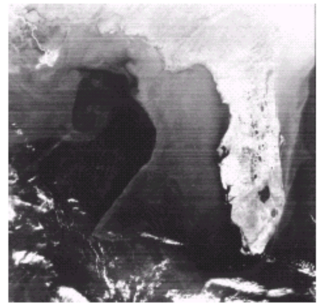
NOTCH PASS FILTERS

$$H_{np}(u,v) = 1 - H_{nr}(u,v)$$

Performs the opposite operation of a notch reject filter.

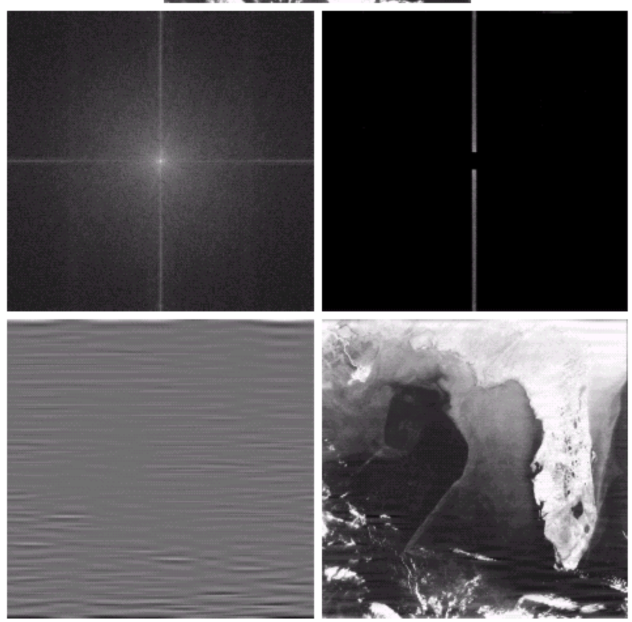
Note that the 3 filters become lowpass filters if $u_0 = v_0 = 0$.

APPLICATION OF A NOTCH PASS FILTER



Notch filtering can do a better job to reduce the scan lines without introducing undesirable blurriness.

Spectrum of the input image



We start with a simple ideal notch pass filter (superimposed on the spectrum) to get an idea about the noise contribution.

Inverse transform of notch-pass filtered result

Now that we have a suitable notch pass filter, it can be used to obtain the corresponding notch reject filter. Hence, the filtered image!

a
b c
d e

FIGURE 5.19 (a) Satellite image of Florida and the Gulf of Mexico (note horizontal sensor scan lines). (b) Spectrum of (a). (c) Notch pass filter shown superimposed on (b). (d) Inverse Fourier transform of filtered image, showing noise pattern in the spatial domain. (e) Result of notch reject filtering. (Original image courtesy of NOAA.)

LINEAR, POSITION-INVARIANT DEGRADATIONS

- **Linear, position-invariant techniques**
  - Many types of degradations can be approximated by linear, position-invariant processes.
  - Tools of linear system theory become available.
- **Nonlinear, position-dependent techniques**
  - More general and usually more accurate
  - Very difficult to solve or no known solutions
- **Before restoration:** $g(x,y) = H[f(x,y)] +$ ♒(x,y)
- **Position-invariant operator**
  - $H[f(x-$♋$, y-$♌$)] = g(x-$♋$, y-$♌$)$, for any $f(x,y)$, and any ♋ & ♌.
  - The response at any point in the image depends only on the value of the input at that point, not on its position.
- **Impulse response of** H: $H[$♎$(x-$♋$, y-$♌$)] = h(x,$♋$,y,$♌$)$
- **If the impulse response of a linear system is known, we can compute the response to any input** f**!**
  - $g(x,y) = h(x,y)*f(x,y)+$ ♒(x,y) or $G(u,v) = H(u,v)F(u,v)+ N(u,v)$
  - ♒(x,y): random values, independent of position.

DEGRADATION FUNCTION ESTIMATION

- Degradations are modeled as being the result of convolution.
- Restoration seeks to find filters that apply the process in reverse.
- Linear image restoration: image deconvolution
- Restoration filters: deconvolution filters
- 3 principal ways to estimate the degradation function
  - Observation
  - Experimentation
  - Mathematical modeling
- Image restoration using a degradation function
  - Blind deconvolution
  - The true degradation function is seldom known completely.

ESTIMATION BY IMAGE OBSERVATION

- Suppose we are given a degraded image without any knowledge about the degradation function *H*.
- One way to estimate the function is to gather information from the image itself.
- Estimation of the degradation function in a blurred image
  - Look for areas of strong signal content (to reduce the effect of noise).
  - Choose a small section of the image with simple structures (e.g., part of an object and the background).
  - Construct an unblurred image of the same size and characteristics of the observed subimage.

    $g_s(x,y)$: observed subimage

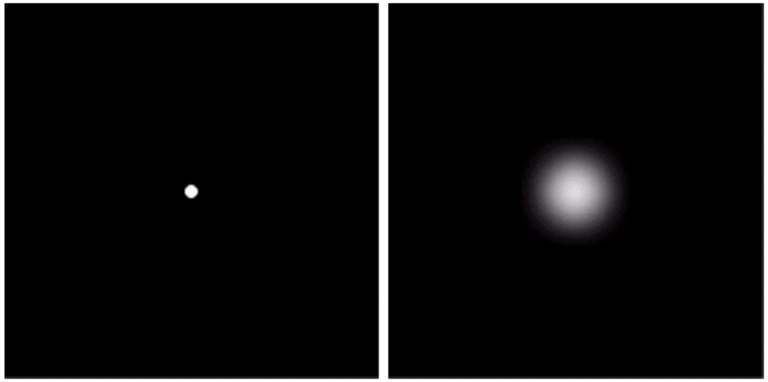
    $\hat{f}_s(x,y)$: constructed subimage

    $$H_s(u,v) = \frac{G_s(u,v)}{\hat{F}_s(u,v)}$$

  - Suppose a plot of $H_s(u,v)$ looks like a Butterworth filter.
  - Construct a function $H(u,v)$ on a larger scale but with the same shape.

ESTIMATION BY EXPERIMENTATION

- If equipment similar to the equipment used to acquire the degraded image, it is possible to obtain an accurate estimation of the degradation.
- Estimation of the degradation function
  - Use various system settings until the image is degraded as closely as possible to the image we want to restore.
  - Obtain the impulse response of the degradation by imaging an impulse.
  - $H(u,v) = G(u,v)/A$



a b

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

ESTIMATION BY MODELING

- Provides insight into the restoration problem.
- In some cases, models take into account environmental conditions.

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.)

***k* = 0.0025**

***k* = 0.001**

***k* = 0.00025**

$$H(u,v) = e^{-k(u^2+v^2)^{5/6}}$$

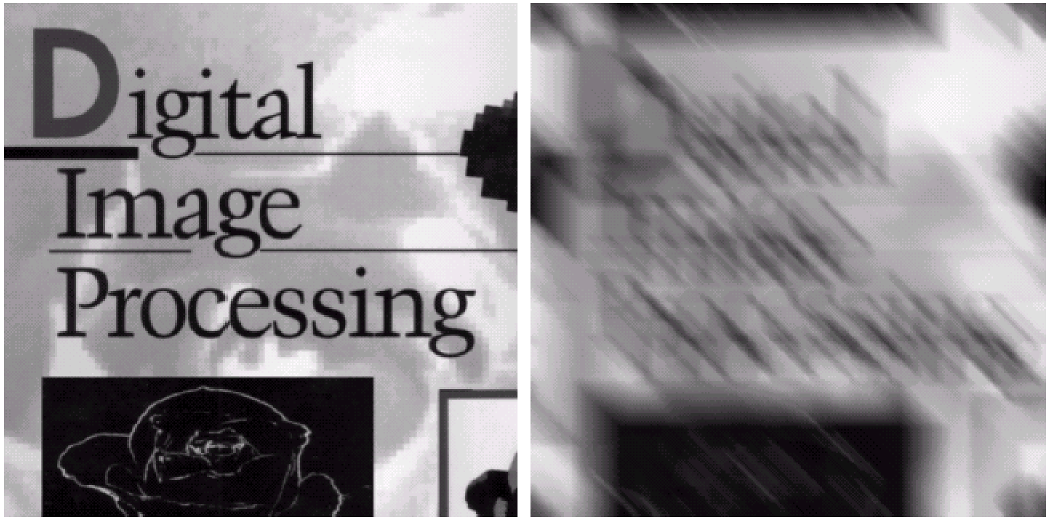
A degradation model based on the physical characteristics of atmospheric turbulence.

Sometimes, Gaussian LPF is used to model mild, uniform blurring.

AN EXAMPLE OF MODELING

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

Model derived from an image that has been blurred by <u>uniform linear motion</u> between the image and the sensor.



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

- **Direct inverse filtering**: **The simplest approach to restoration.**
- **The following derivation shows that we cannot recover the undegraded image exactly because *N(u,v)* is a random function whose Fourier Transform is not known!**

$$\hat{F}(u,v) = \frac{G(u,v)}{H(u,v)}$$

$$G(u,v) = H(u,v)F(u,v) + N(u,v)$$

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

- **More bad news**: **If the degradation has zero or very small values, the ratio *N*(*u*,*v*)/*H*(*u*,*v*) may dominate the estimate.**
  - **To circumvent the problem, we can limit the filter frequencies to values near the origin.**
  - ***H*(0,0) = Mean of *h*(*x*,*y*): usually the highest value of *H*(*u*,*v*).**

AN EXAMPLE OF 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.

Degradation function:

$$H(u,v) = e^{-k[(u-M/2)^2+(v-N/2)^2]^{5/6}}$$

with $k = 0.0025$

The cutoff was implemented by applying a Butterworth lowpass function of order 10.

MINIMUM MEAN SQUARE ERROR (WIENER) FILTERING

- **Inverse filtering makes no explicit provision for handling noise.**
- **Now we consider both images and noise as <u>random processes</u>.**
- **The objective: Find an estimate $\hat{f}$ of the uncorrupted image f such that the mean squared error between them is minimized.**
- **Assumptions**
  - **The image and the noise are <u>uncorrelated</u>.**
  - **The image or the noise has <u>zero mean</u>.**
  - **The gray levels in the estimate are a <u>linear function</u> of the levels in the degraded image.**
- **The restored image is given by the inverse Fourier transform of $\hat{F}(u,v)$.**
- **Noise = zero ⇒ Wiener filter = inverse filter!**

DERIVATION OF THE WIENER FILTER

Minimize $e^2 = E\{(f - \hat{f})^2\}$

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

$H(u,v) =$ transform of the degradation function

$G(u,v) =$ transform of the degraded image

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

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

If the 2 power spectrums are not known, use $S_\eta(u,v)/S_f(u,v) = K$

AN EXAMPLE OF WIENER FILTERING

Degraded input image

a b c

FIGURE 5.28 Comparison of inverse- and Wiener filtering. (a) Result of full inverse filtering of Fig. 5.25(b). (b) Radially limited inverse filter result. (c) Wiener filter result.

K **was chosen interactively to yield the best possible visual results.**

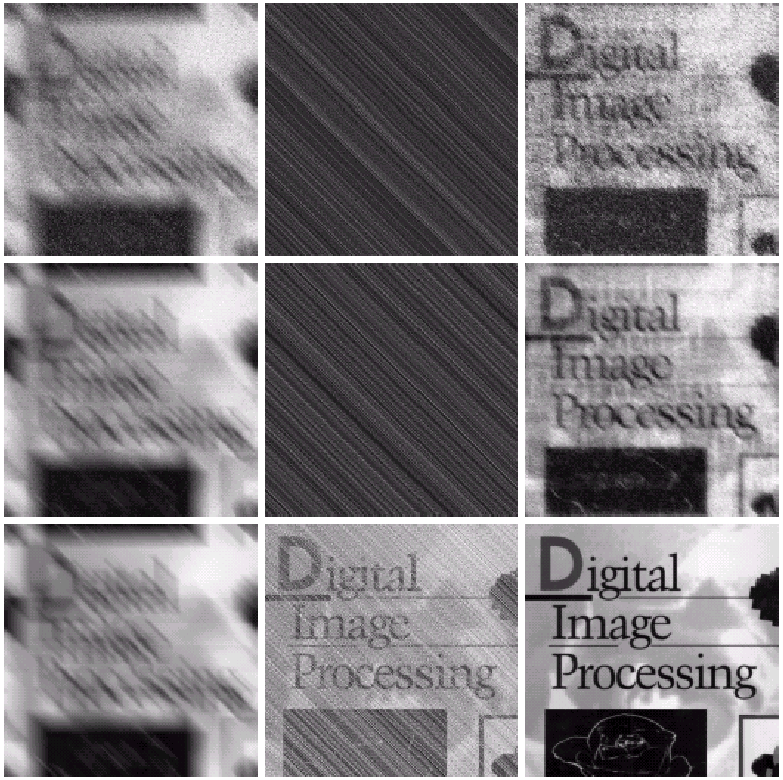
Previous results for comparison

AN EXAMPLE OF WIENER FILTERING

Blurred image with additive Gaussian noise (mean=0, variance=650) →

Noise variance is reduced by one order of magnitude →

Noise variance is reduced by five orders of magnitude →



a b c
d e f
g h i

FIGURE 5.29 (a) 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.

Wiener filter with

$$H(u,v) = e^{-k[(u-M/2)^2+(v-N/2)^2]^{5/6}}$$

where *k* = 0.0025.

***K* was chosen interactively.**