Image Restoration
Image degradation model

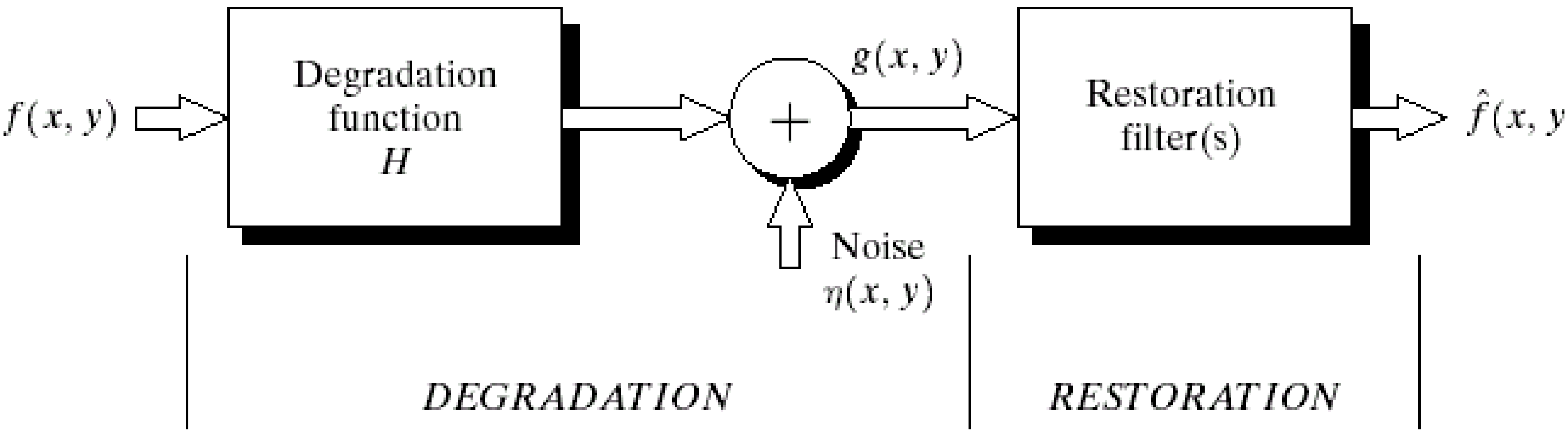


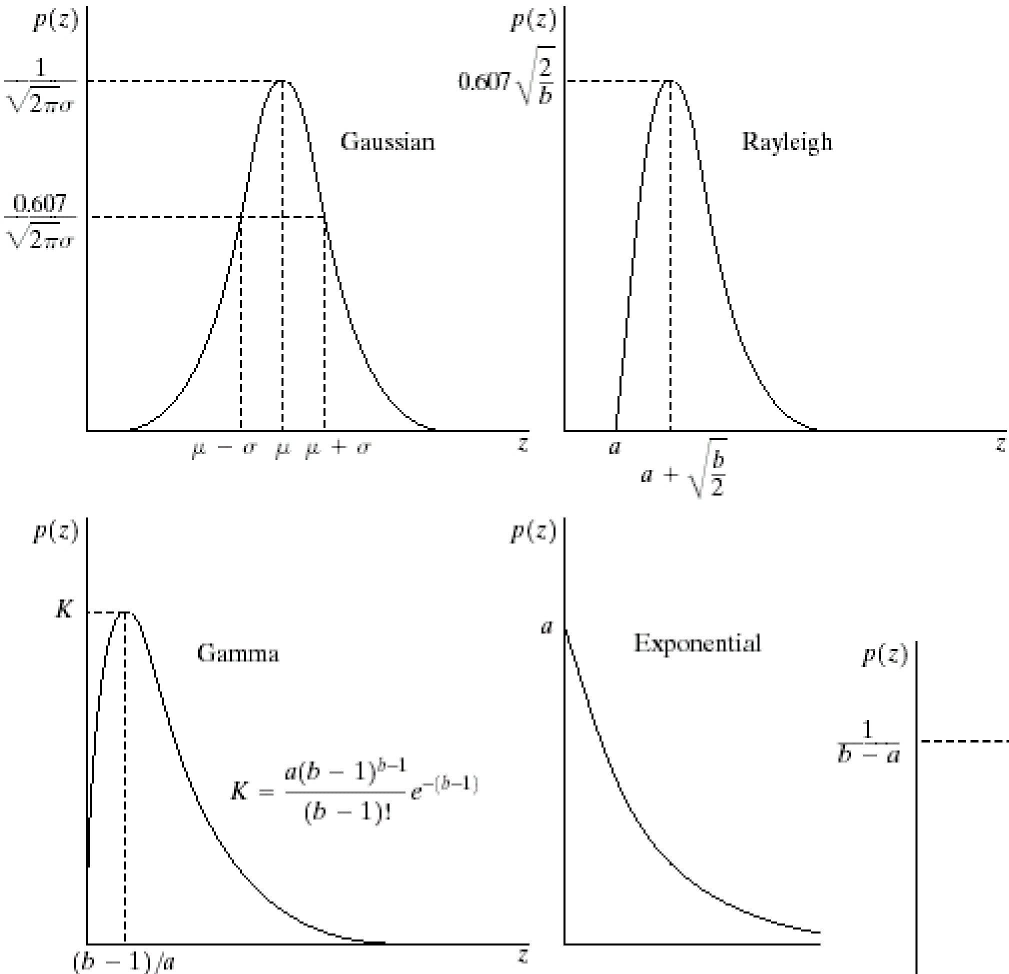
FIGURE 5.1 A model of the image degradation/restoration process.

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

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

Note: even small amounts of noise can lead to large errors in restoration → dip08_0.m

Image Restoration
Noise models



Typical pdf's of i.i.d. noise

Note 1: In the additive model, *zero-mean* Gaussian and Uniform noises are typically used

Note 2: In real cases, noise is often *signal-dependent*, e.g. multiplicative (with mean=1) → does not comply with our model

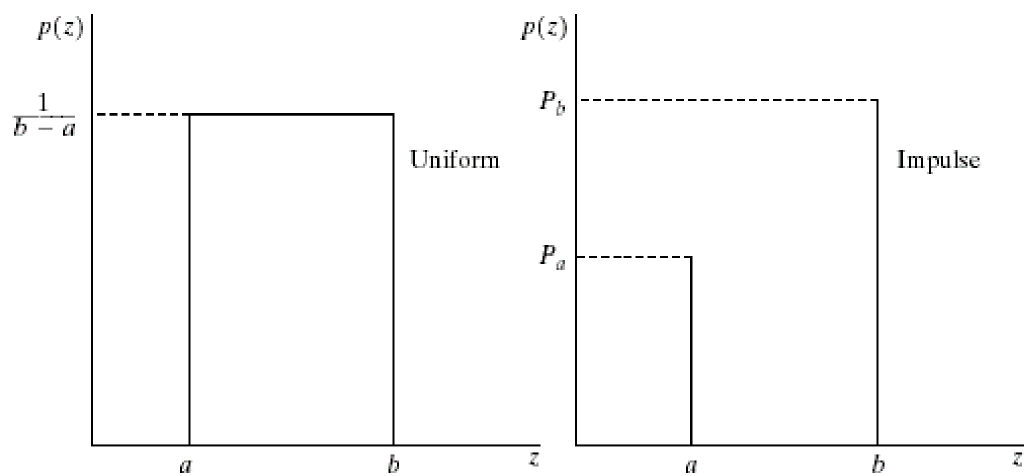


Image Restoration
Noise models

Name	PDF	Mean and Variance	CDF
Uniform	$p_z(z) = \begin{cases} \dfrac{1}{b-a} & \text{if } a \le z \le b \\ 0 & \text{otherwise} \end{cases}$	$m = \dfrac{a+b}{2}, \quad \sigma^2 = \dfrac{(b-a)^2}{12}$	$F_z(z) = \begin{cases} 0 & z < a \\ \dfrac{z-a}{b-a} & a \le z \le b \\ 1 & z > b \end{cases}$
Gaussian	$p_z(z) = \dfrac{1}{\sqrt{2\pi}b} e^{-(z-a)^2/2b^2}$ $-\infty < z < \infty$	$m = a, \quad \sigma^2 = b^2$	$F_z(z) = \int_{-\infty}^{z} p_z(v)\, dv$
Salt & Pepper	$p_z(z) = \begin{cases} P_a & \text{for } z = a \\ P_b & \text{for } z = b \\ 0 & \text{otherwise} \end{cases}$ $b > a$	$m = aP_a + bP_b$ $\sigma^2 = (a-m)^2 P_a + (b-m)^2 P_b$	$F_z(z) = \begin{cases} 0 & \text{for } z < a \\ P_a & \text{for } a \le z < b \\ P_a + P_b & \text{for } b \le z \end{cases}$
Lognormal	$p_z(z) = \dfrac{1}{\sqrt{2\pi}bz} e^{-[\ln(z)-a]^2/2b^2}$ $z > 0$	$m = e^{a+(b^2/2)}, \sigma^2 = [e^{b^2} - 1]e^{2a+b^2}$	$F_z(z) = \int_0^z p_z(v)\, dv$
Rayleigh	$p_z(z) = \begin{cases} \dfrac{2}{b}(z-a)e^{-(z-a)^2/b} & z \ge a \\ 0 & z < a \end{cases}$	$m = a + \sqrt{\pi b/4}, \sigma^2 = \dfrac{b(4-\pi)}{4}$	$F_z(z) = \begin{cases} 1 - e^{-(z-a)^2/b} & z \ge a \\ 0 & z < a \end{cases}$
Exponential	$p_z(z) = \begin{cases} ae^{-az} & z \ge 0 \\ 0 & z < 0 \end{cases}$	$m = \dfrac{1}{a}, \quad \sigma^2 = \dfrac{1}{a^2}$	$F_z(z) = \begin{cases} 1 - e^{-az} & z \ge 0 \\ 0 & z < 0 \end{cases}$

Image Restoration
Noise models

Gaussian noise is typical in sensors, especially in low light conditions

Impulse (salt-and-pepper) noise comes from disturbed switching devices

Silver halide grains in photographic films yield **lognormal** distributions

Rayleigh noise (multipl.) arises in range images (*n*-look SAR images)

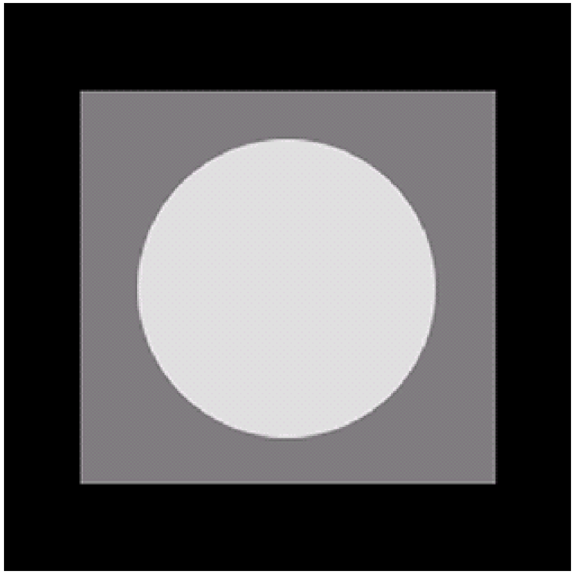
Exponential noise (multipl.) is present in laser imaging (1-look SAR)

For simulation purposes, a noise having a specified cdf *Fz* can be generated starting from a **uniform** random field *w* by the inverse function mapping: $z = F_z^{-1}(w)$

(like in the case of histogram specification)

(Matlab: use *rand* and *randn,* or *imnoise* → [NoiseGeneration.m])

Image Restoration
Noise models

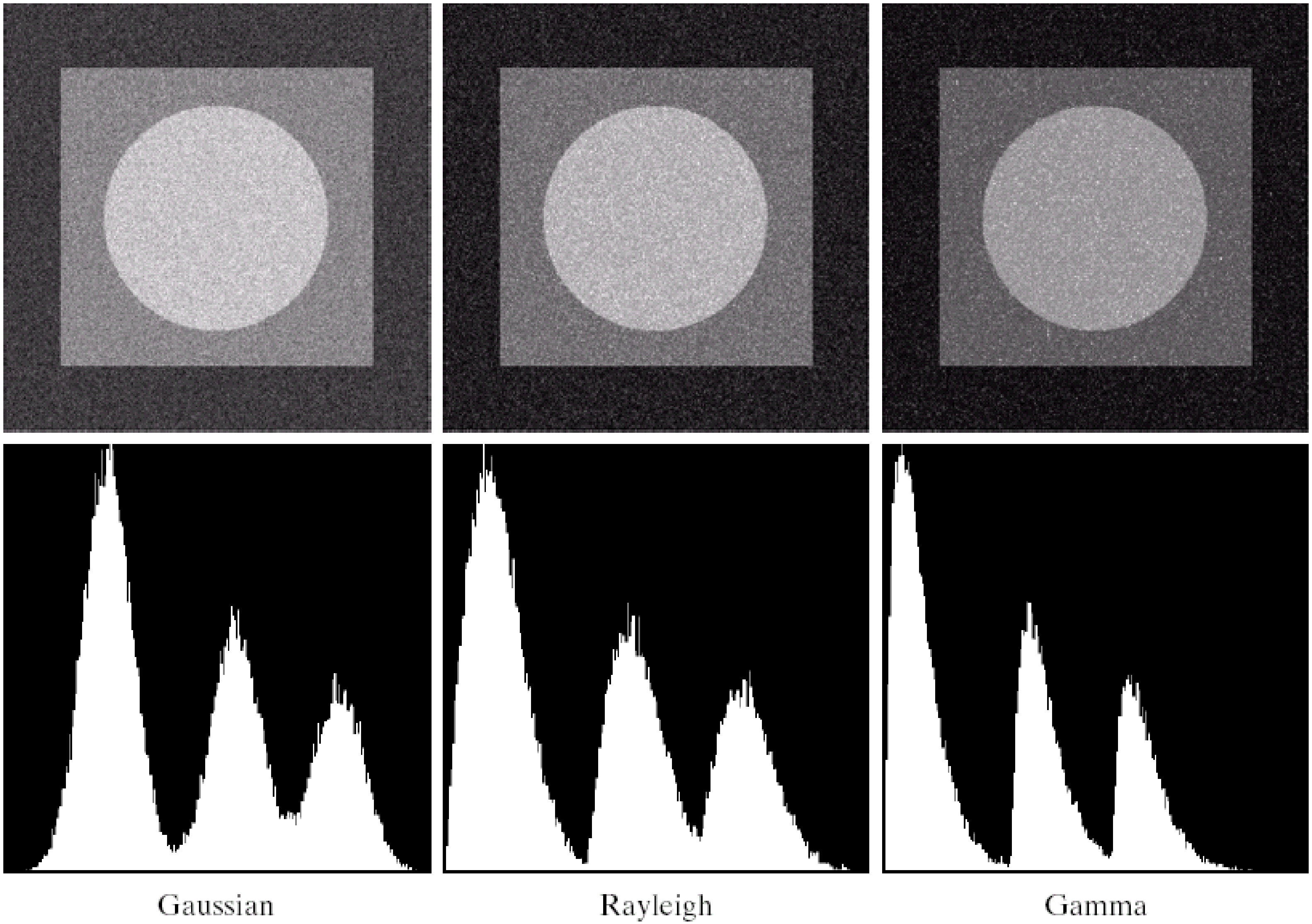


black, gray1, gray2

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

In general, the noise distribution is not easily recognizable by image inspection.

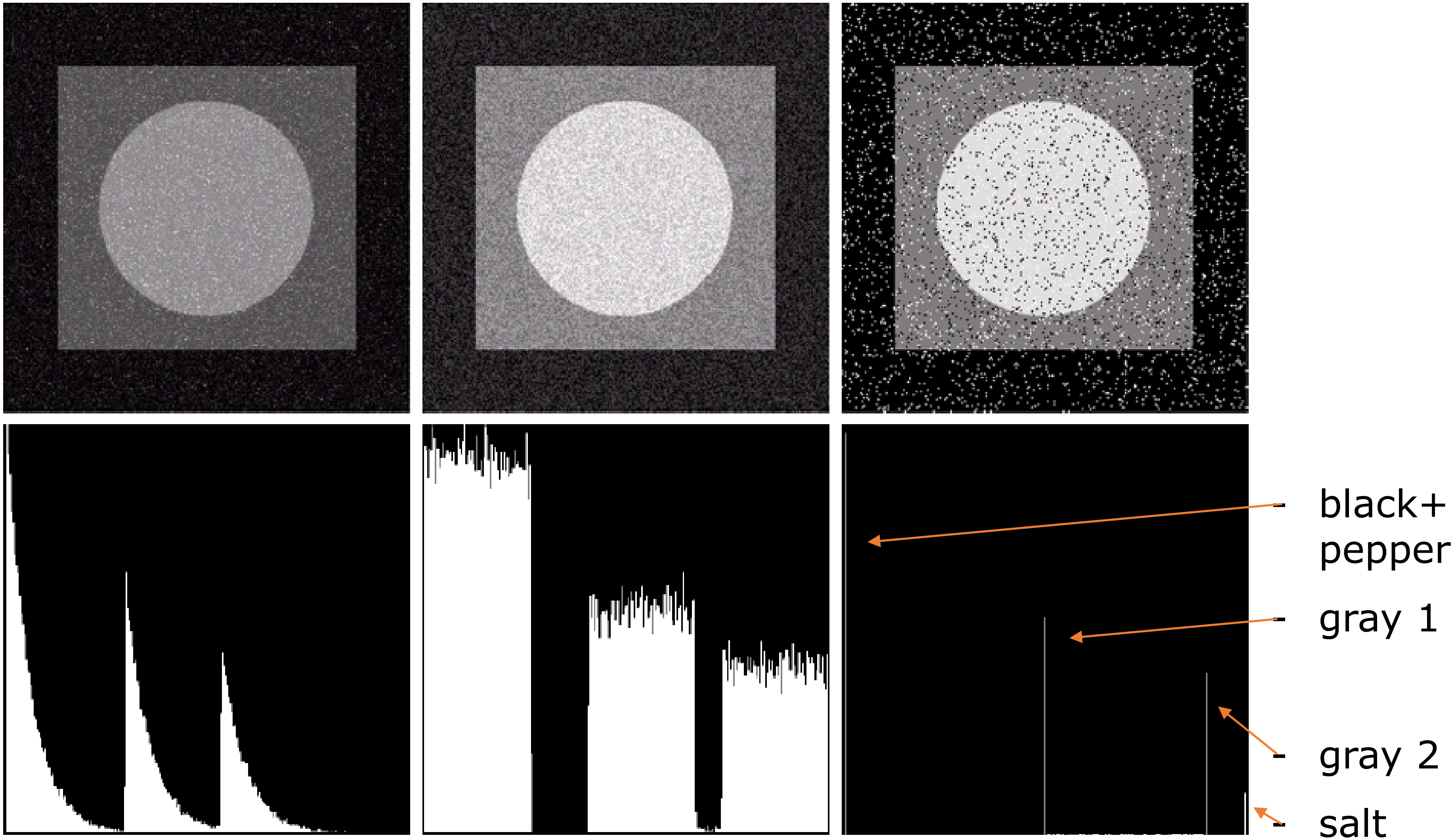
Image Restoration
Noise models



Note:

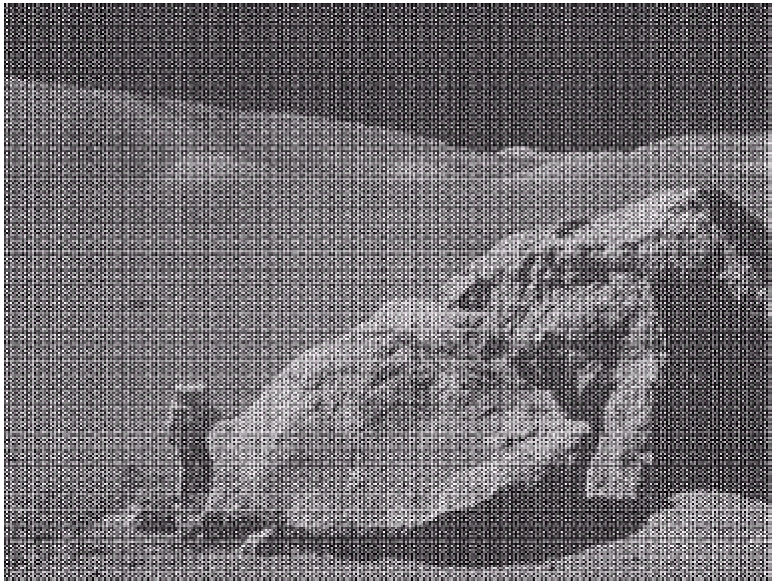
noise with mean>0 was used in these images

Image Restoration
Noise models



Exponential | Uniform | Salt & Pepper

Image Restoration
Noise models



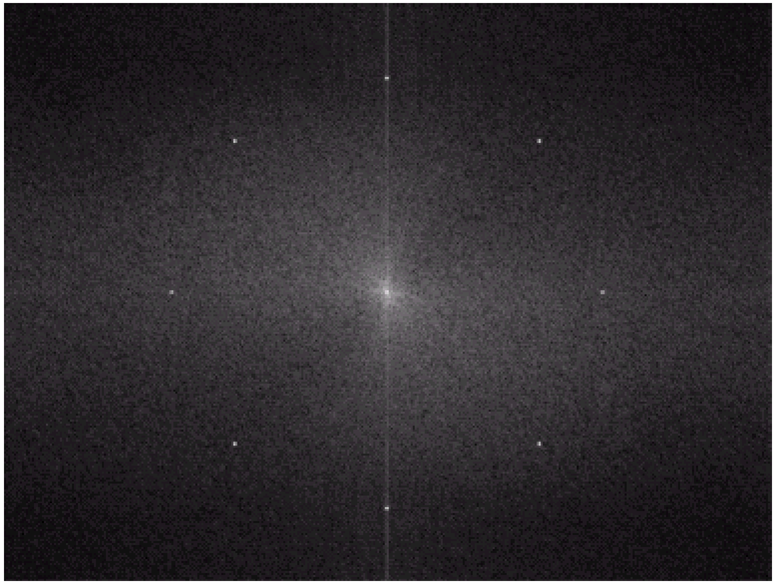
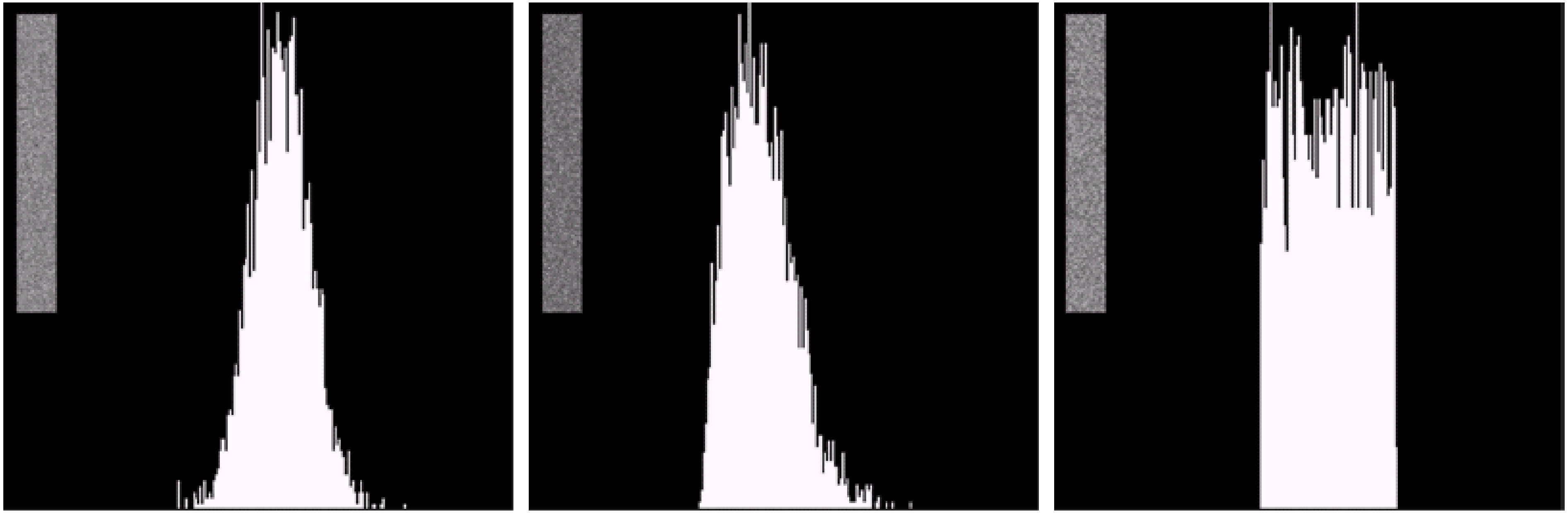


FIGURE 5.5
(a) Image corrupted by sinusoidal noise. (b) Spectrum (each pair of conjugate impulses corresponds to one sine wave). (Original image courtesy of NASA.)

Image Restoration
Noise models

Noise estimation from a locally uniform area of the image



a b c

FIGURE 5.6 Histograms computed using small strips (shown as inserts) from (a) the Gaussian, (b) the Rayleigh, and (c) the uniform noisy images in Fig. 5.4.

UNIVERSITÀ DEGLI STUDI DI TRIESTE

Nonlinear mean filters

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

We are looking for the best compromise between noise attenuation and detail preservation.

Let S_{xy} be an *mxn* neighborhood of (*x*,*y*); define various **mean filters**:

Arithmetic mean

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

Geometric mean
always smaller than the arithm. mean

$$\hat{f}(x,y) = \left(\prod_{(s,t)\in Sxy} g(s,t) \right)^{1/mn}$$

Contraharmonic means

$$\hat{f}(x,y) = \sum_{(s,t)\in Sxy} g(s,t)^{Q+1} \Big/ \sum_{(s,t)\in Sxy} g(s,t)^{Q}$$

Q=0 → arithmetic mean;

Q>0 → larger than the arithm. mean (good for dark impulse noise)

Q<0 → smaller than the arithm. mean (good for bright impulse noise)

Order statistics filters

Median filter

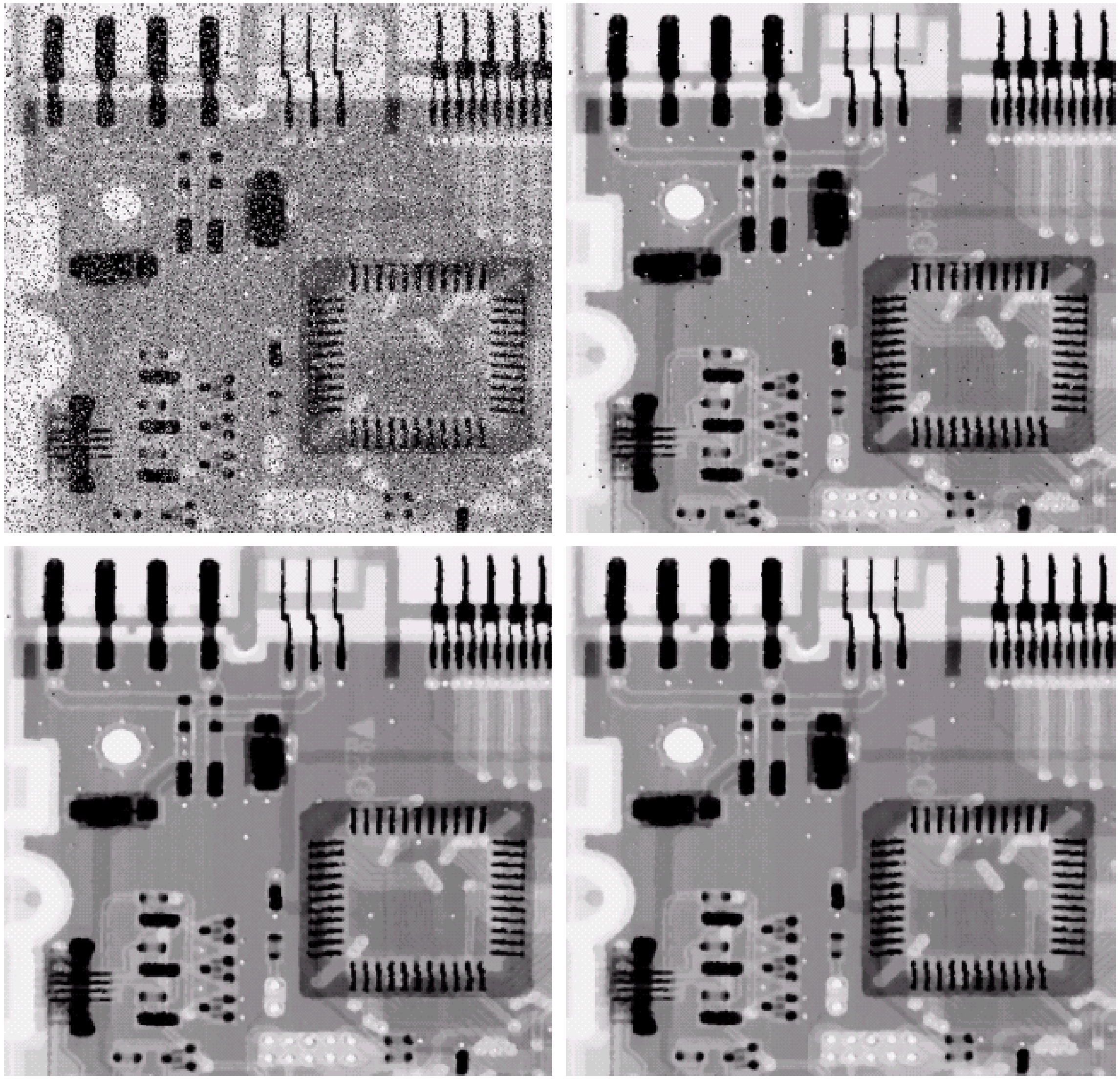
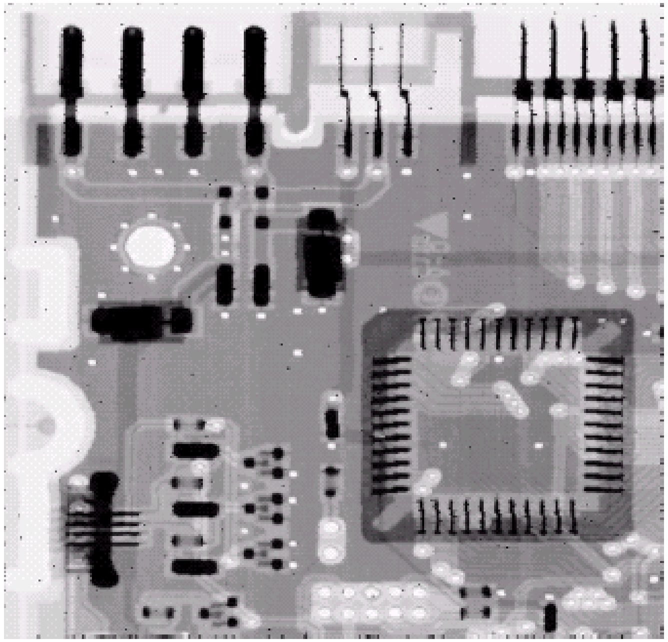
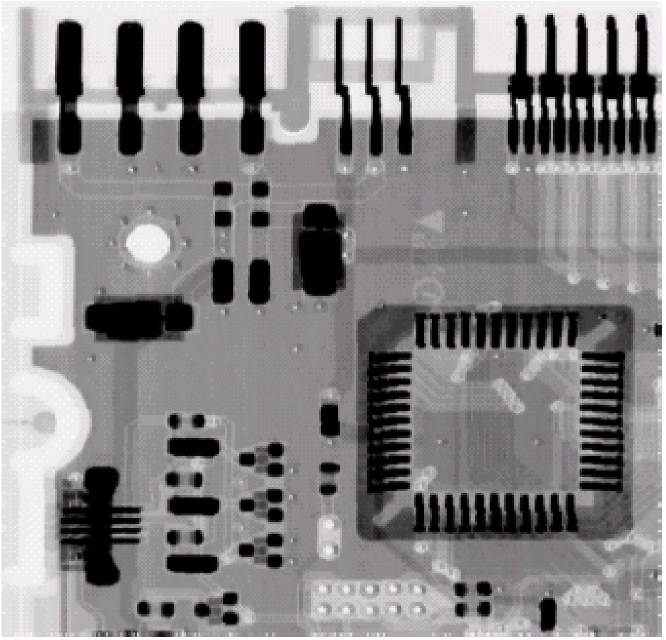


FIGURE 5.10 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.1$. (b) Result of one pass with a median filter of size 3×3. (c) Result of processing (b) with this filter. (d) Result of processing (c) with the same filter.

Order statistics filters





Max filter **Min filter**

Order statistics filters

Weighted Median (WM) filters: output depends also on the *position* of the gray levels within the window

WM filter of span N associated with integer weights $w = [w_1, w_2, \ldots, w_N]$:

$$\hat{f}(x) = \underset{s \in Sx}{\mathrm{median}}\{g_W(s)\}$$

where g_W is obtained from g by *replication* of its elements:

$$g_W(s) = [w_1 \Diamond g(1), w_2 \Diamond g(2), \ldots, w_N \Diamond g(N)], \quad \text{and} \quad k \Diamond g = \overbrace{g, g, \ldots, g}^{k}$$

I.e.: replicate each sample $g(i)$ for w_i times, and choose the median value from the new sequence.

Example: length-5 WM filter with weights [1, 2, 3, 2, 1]. Input sequence g= […, -1, 5, 8, 11, -2, …], window centered at sample value 8. After sorting and duplication: g_W=[11, 11, 8, 8, 8, 5, 5, -1, -2]. Output is 8, whereas the 5-point median would output 5.

*Note: Central WM (**CWM**) filters are particularly important in this class*

Order statistics filters

Alpha-trimmed mean filter:

Sort the pixels in S_{xy} and delete the first *d/2* and the last *d/2*; let S'_{xy} be the set of the remaining pixels. The output is:

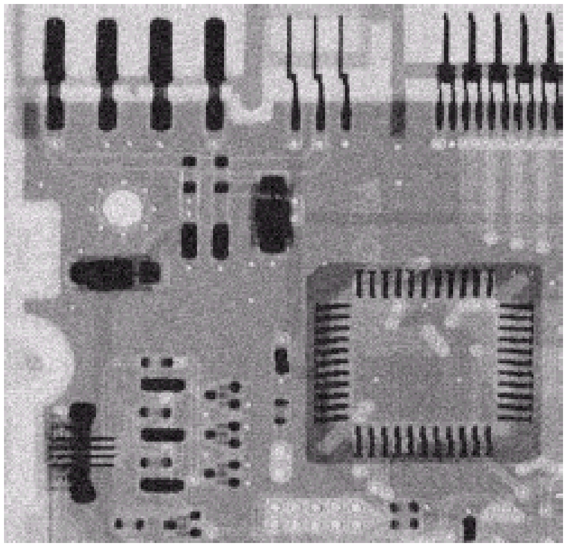
$$\hat{f}(x,y) = \frac{1}{mn-d} \sum_{(s,t)\in S'xy} g(s,t)$$

Defaults to the arithmetic mean filter when $d=0$ and to the median filter when $d=mn$-1.

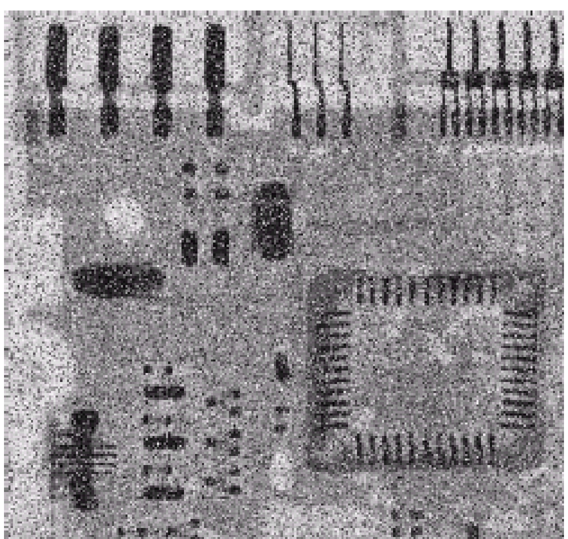
Good for *mixed short- and long-tailed noise*

Order statistics filters

a



b



c d

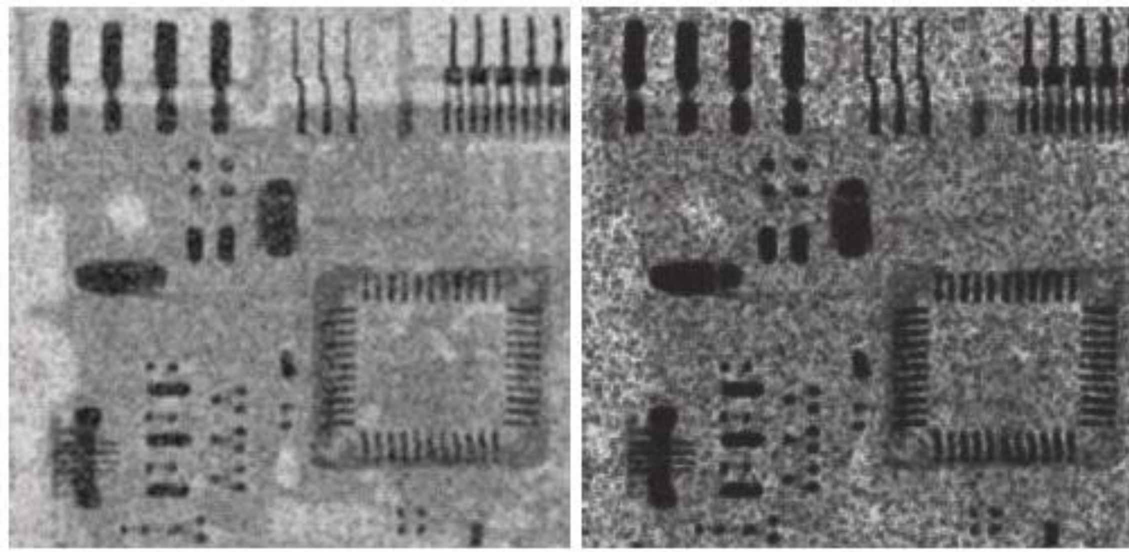
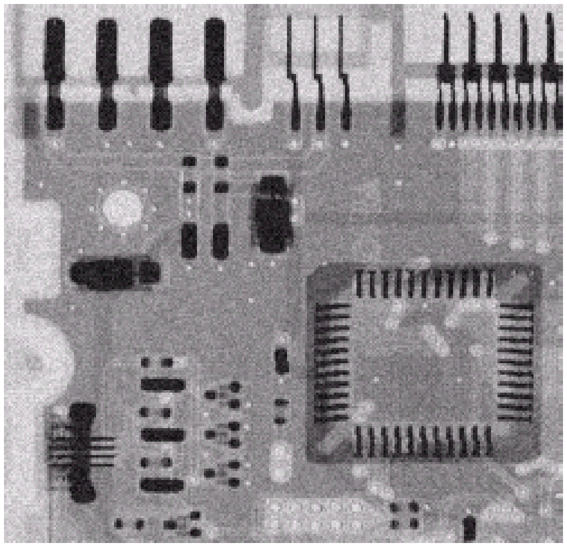


FIGURE 5.12 (a) Image corrupted by additive uniform noise. (b) Image additionally corrupted by additive salt-and-pepper noise. Image in (b) filtered with a 5×5: (c) arithmetic mean filter; (d) geometric mean filter; (e) median filter; and (f) alpha-trimmed mean filter with $d = 5$.

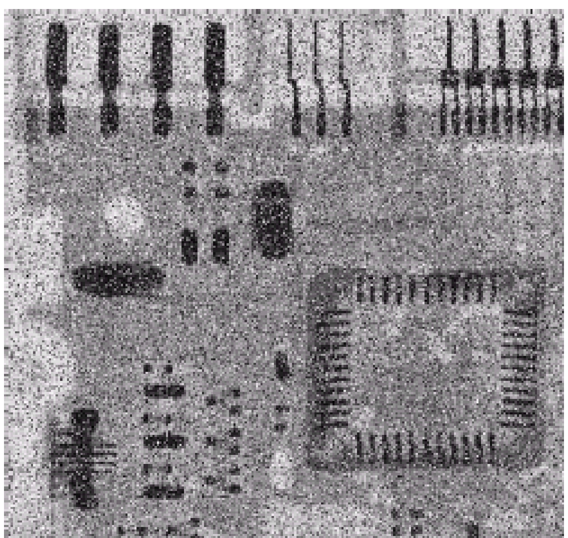
Order statistics filters

→ NonlinearFilters.m

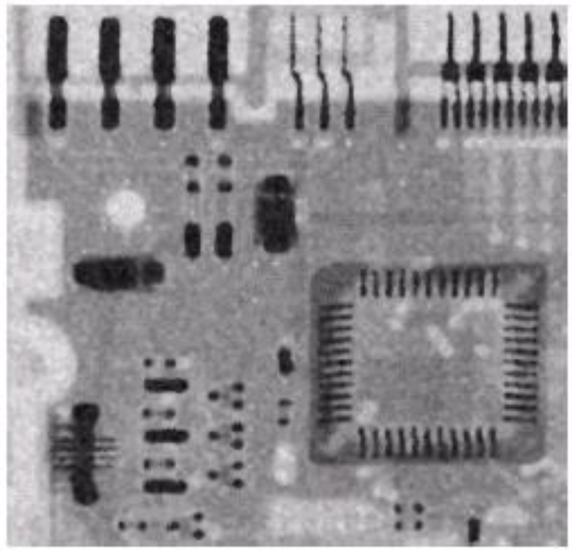
a



b



e



f

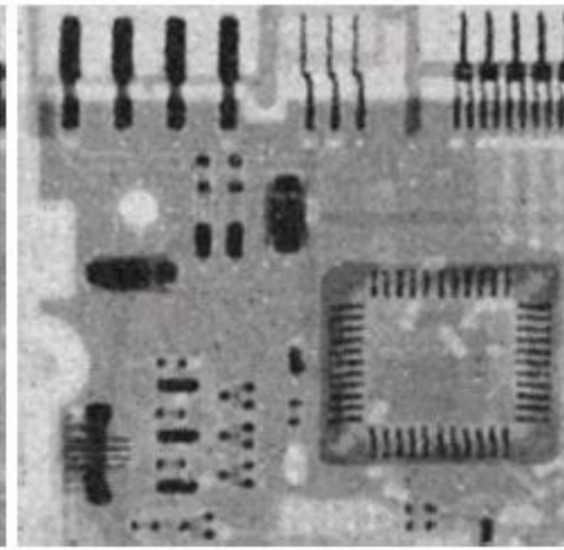


FIGURE 5.12 (a) Image corrupted by additive uniform noise. (b) Image additionally corrupted by additive salt-and-pepper noise. Image in (b) filtered with a 5×5: (c) arithmetic mean filter; (d) geometric mean filter; (e) median filter; and (f) alpha-trimmed mean filter with $d = 5$.

Adaptive linear filters

The response of the filter changes according to the local properties of the image. Used features:

m_L, σ_L^2 local average, local variance of the pixels in *Sxy*

σ_n^2 variance of the noise corrupting *f*(*x*,*y*) (to be estimated!)

The output is (Wiener filter-like approach, see [Lim p.538]):

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

Note1: it is reasonable to suppose that $\sigma_L^2 \geq \sigma_n^2$ everywhere

but for safety truncate $\sigma_n^2 / \sigma_L^2 > 1 \quad \rightarrow \quad = 1$

Note2: good noise estimate: $\sigma_n^2 \cong mean(\sigma_L^2)$ (Matlab's `wiener2`)

Adaptive linear filters

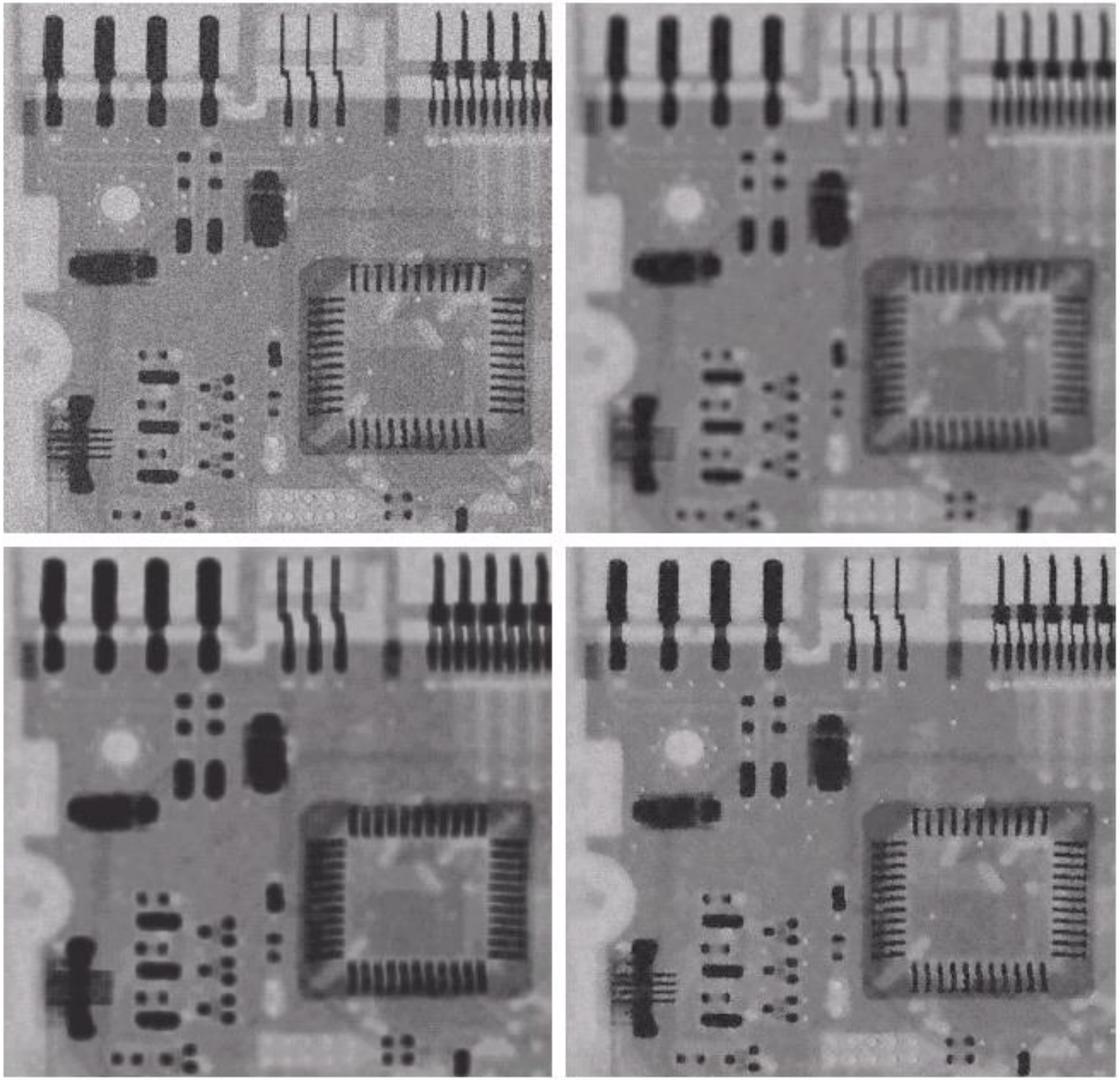


FIGURE 5.13 (a) Image corrupted by additive Gaussian noise of zero mean and variance 1000. (b) Result of arithmetic mean filtering. (c) Result of geometric mean filtering. (d) Result of adaptive noise reduction filtering. All filters were of size 7×7.

Adaptive median filters

Define:

$$mi = \min_{(s,t)\in Sxy}\{g(s,t)\}; \quad ma = \max_{(s,t)\in Sxy}\{g(s,t)\}; \quad md = \operatorname*{median}_{(s,t)\in Sxy}\{g(s,t)\};$$

Algorithm:

window size = minsize:

while (window size < maxsize) do

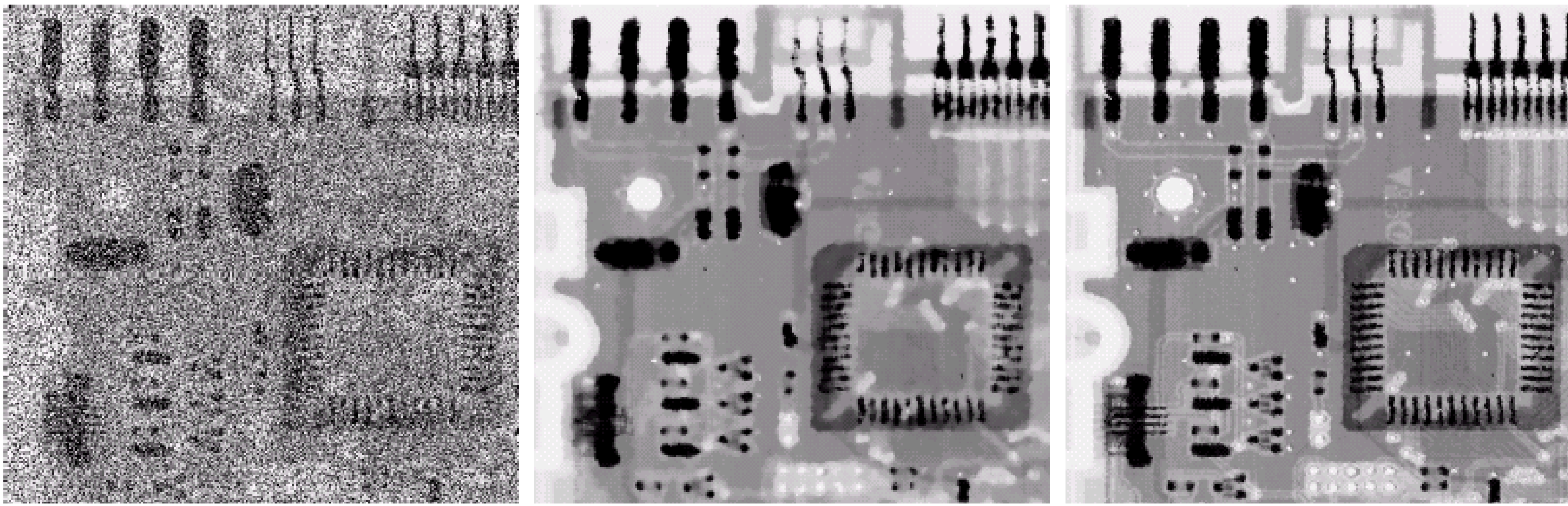
if (*md=mi* or *md=ma*) increase window size;

end do;

if (*mi<g(x,y)<ma*) output *g(x,y)*, else output *md*;

Useful for images corrupted by heavy impulse noise.

Adaptive median filters



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 $S_{\max} = 7$.

Rational filter

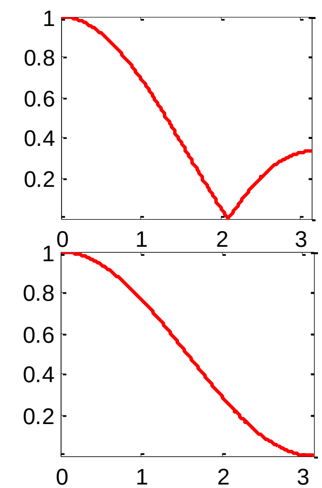
A linear lowpass filter can be combined with an edge sensor to achieve **edge-preserving noise smoothing**

	d	
a	*b*	*c*
	e	

$$b' = b + \frac{a+c-2b}{k(a-c)^2+5+W} + \frac{d+e-2b}{k(d-e)^2+5+W}$$

k: edge protection factor
W: lowpass filter control factor

- $k = 0,\ W = 0$: $b' = (a+b+c+d+e)\ /\ 5$
- $k = 0,\ W > 0$: $b' = [(a+c+d+e-4b)+(5+W)b]\ /\ (5+W)$ $= [(a+c+d+e)+(1+W)b]\ /\ (5+W)$
- $k > 0$: smoothing only **along** edges



Note 1: filter is meant to operate iteratively
Note 2: this is a sort of *Anisotropic Diffusion*
Note 3: asymptotic output is a uniform image

Rational filter for speckle noise

On a 3x3 support:

1	**2**	**3**
8	**0**	**4**
7	**6**	**5**

$$x'_0 = x_0 + \sum_{i=1}^{8} \frac{x_i - x_0}{k(x_i - x_0)^2/(x_i + x_0 + 1)^2 + A} \,. \qquad (2)$$

This filter performs a smoothing action that is more delicate in low luminance areas (where $(x_i + x_0 + 1)^2$ is small), and becomes stronger in bright image zones. In this way, it complies with the nature of speckle noise, which is multiplicative and, hence, has amplitude proportional to the local mean value of the signal.

k: edge protection factor, range [700, 1100]
A: lowpass filter control factor, range [4.0, 5.0]
no. of passes: [6, 10]

- note k=0; A=9 → 3x3 average filter
- if k>0 the denominator is "locally larger" than A

Bilateral filter

Combined **domain** and **range** filtering:

$$\mathbf{h}(\mathbf{x}) = k^{-1}(\mathbf{x}) \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \mathbf{f}(\xi) c(\xi, \mathbf{x}) s(\mathbf{f}(\xi), \mathbf{f}(\mathbf{x}))\, d\xi$$

with the normalization

$$k(\mathbf{x}) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} c(\xi, \mathbf{x}) s(\mathbf{f}(\xi), \mathbf{f}(\mathbf{x}))\, d\xi \ .$$

The *closeness function* $c(\xi, \mathbf{x}) = e^{-\frac{1}{2}\left(\frac{d(\xi,\mathbf{x})}{\sigma_d}\right)^2}$ and

the *similarity function* $s(\xi, \mathbf{x}) = e^{-\frac{1}{2}\left(\frac{\delta(\mathbf{f}(\xi),\mathbf{f}(\mathbf{x}))}{\sigma_r}\right)^2}$

are Gaussian functions of

- the Euclidean distance *d*
- the absolute difference δ between their arguments.

- σ_d, σ_r are the *geometric (domain) spread* and the *photometric (range) spread* parameters

- Fast realization techniques exist zz08_FastBilateral

BM3D filter – Noise modelling

Foi07 and later – https://webpages.tuni.fi/foi/GCF-BM3D/
Matlab sw updated continuously as of 2021

Image Denoising by Sparse 3-D Transform-Domain Collaborative Filtering

Kostadin Dabov, *Student Member, IEEE*, Alessandro Foi, Vladimir Katkovnik, and Karen Egiazarian, *Senior Member, IEEE*

Rakhshanfar16 - https://users.encs.concordia.ca/~amer/

Estimation of Gaussian, Poissonian–Gaussian, and Processed Visual Noise and Its Level Function

Meisam Rakhshanfar, *Student Member, IEEE*, and Maria A. Amer, *Senior Member, IEEE*

Noise smoothing

ISO/ASA is a sensitivity measure for photographic film, i.e. for signal amplification in a digital camera
High ASA setting → small signal → large amplification → large noise

- Noise smoothing in a consumer camera:

100 ASA, 1.3 s

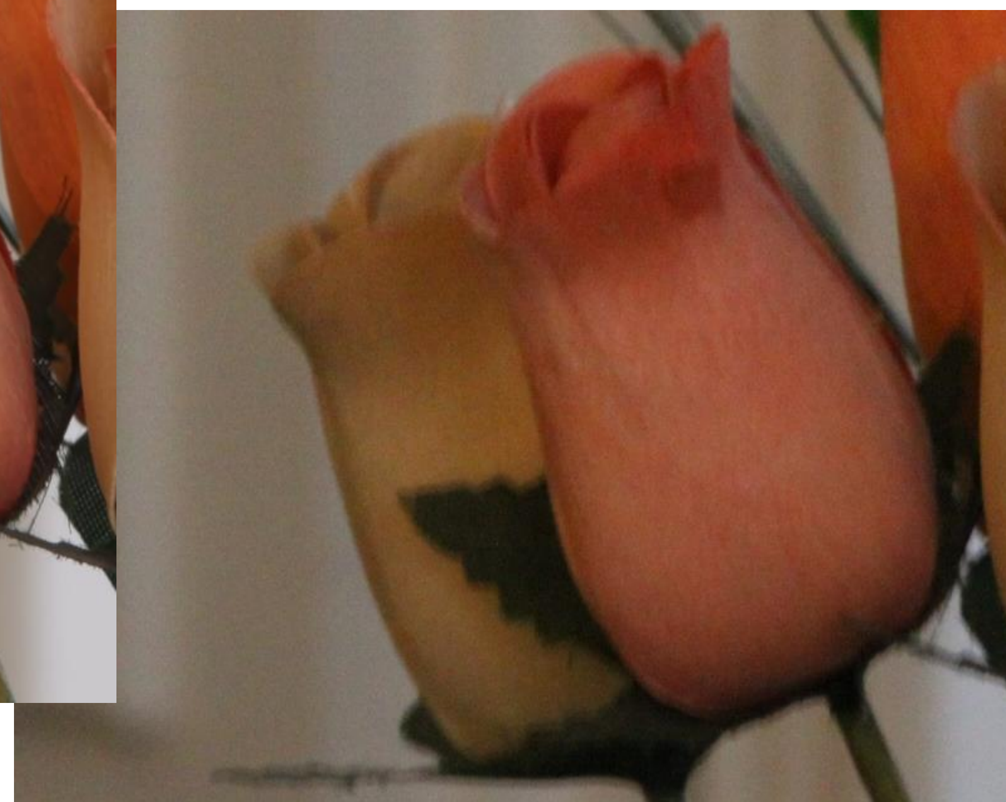
1600 ASA, 0.1 s

Noise smoothing

Noise smoothing in a professional camera (JPEG output)

100 ASA, 6 s

3200 ASA, 1/8 s



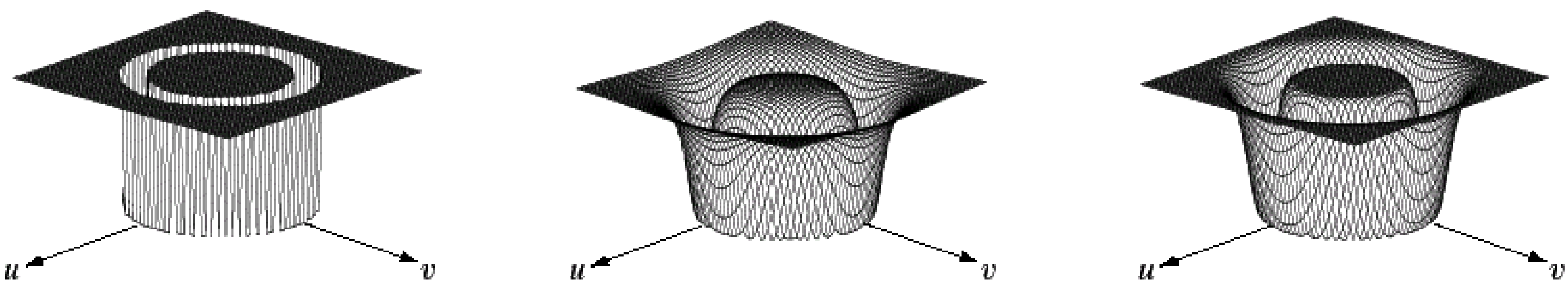
Frequency-selective filters

Let $D(u,v)$ be the distance from the origin of a given frequency.

Band-reject filters (reject center frequency D_0, reject bandwidth W):

Ideal

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



a b c

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

Frequency-selective filters

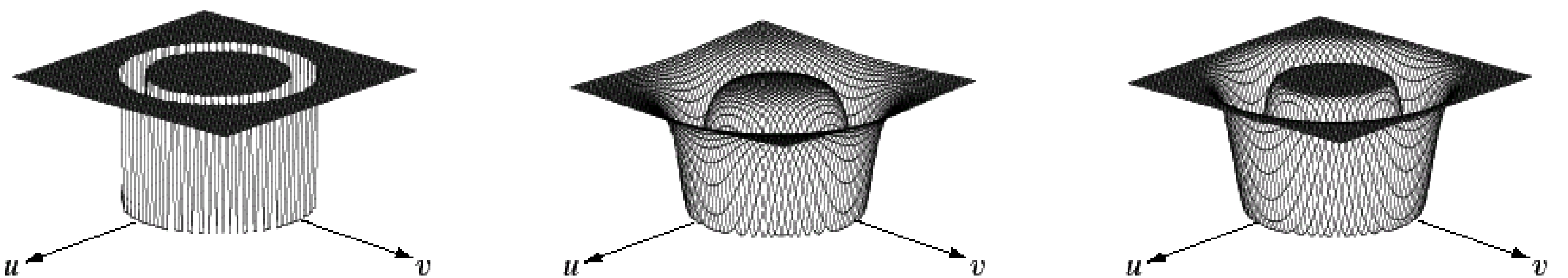
Butterworth

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

and Gaussian

band-reject filters:

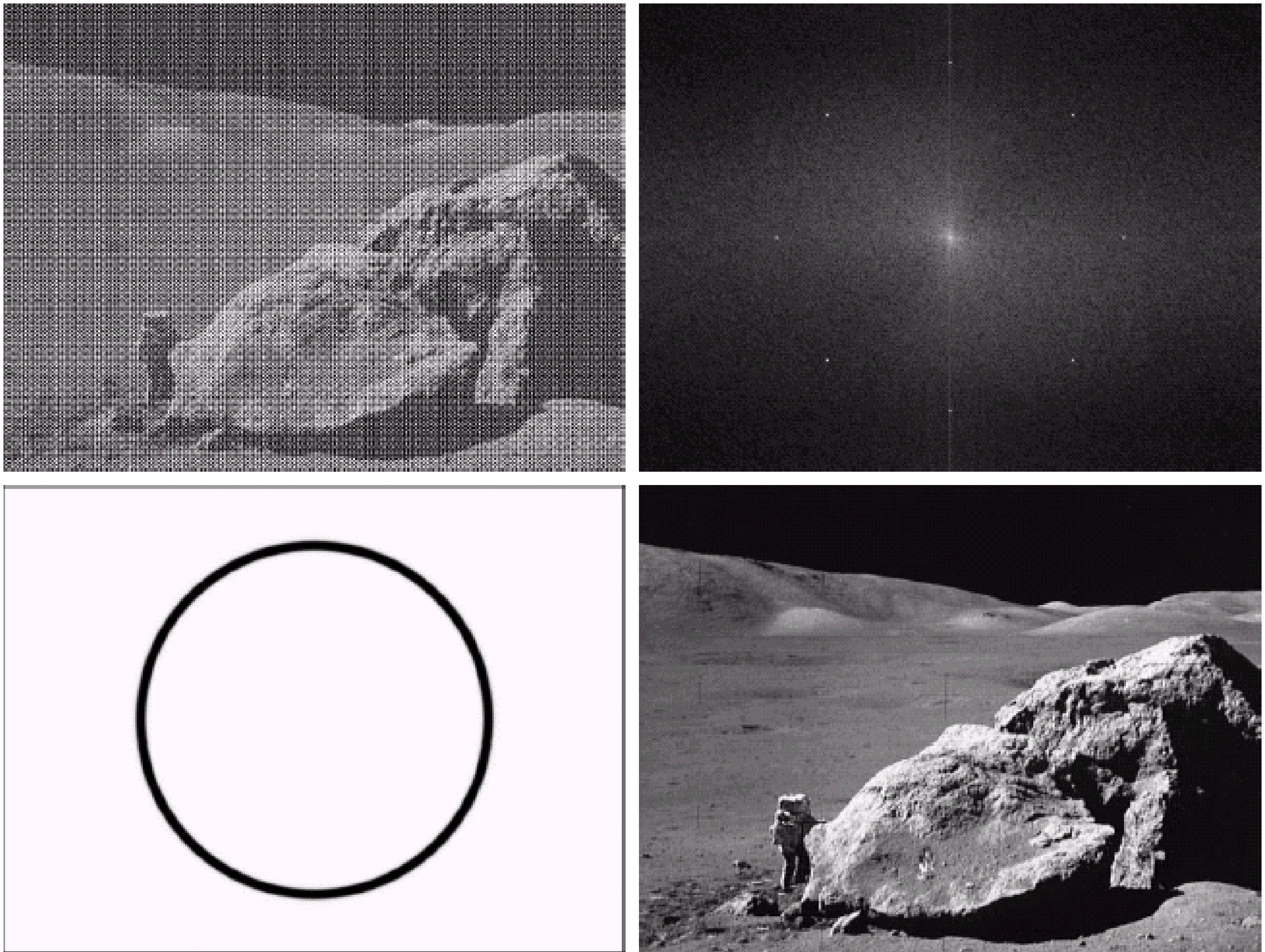
$$H(u,v)=1-\exp\left[-\frac{1}{2}\left(\frac{D^2(u,v)-D_0^2}{D(u,v)W}\right)^2\right]$$



a b c

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

Frequency-selective filters



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.)

Frequency-selective filters

Band-pass filter: $H_{BP}(u,v) = 1 - H_{BR}(u,v)$

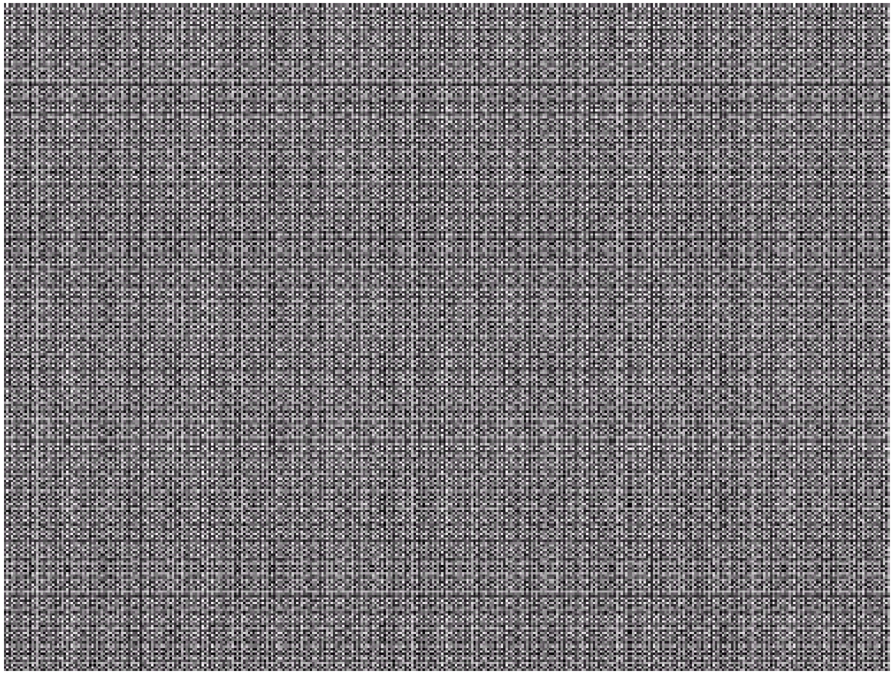


FIGURE 5.17 Noise pattern of the image in Fig. 5.16(a) obtained by bandpass filtering.

Frequency-selective filters

Notch filters

Let

$$D_1(u,v)=\sqrt{(u-M/2-u_0)^2+(v-N/2-v_0)^2}$$

$$D_2(u,v)=\sqrt{(u-M/2+u_0)^2+(v-N/2+v_0)^2}$$

distances of center-shifted freq. (u,v) from freqs. (u_0, v_0) and $(-u_0, -v_0)$

Ideal

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

Butterworth

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

Gaussian

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

Note: for the filter coeff. to be real, notch areas must always be defined in symmetric pairs

Frequency-selective filters

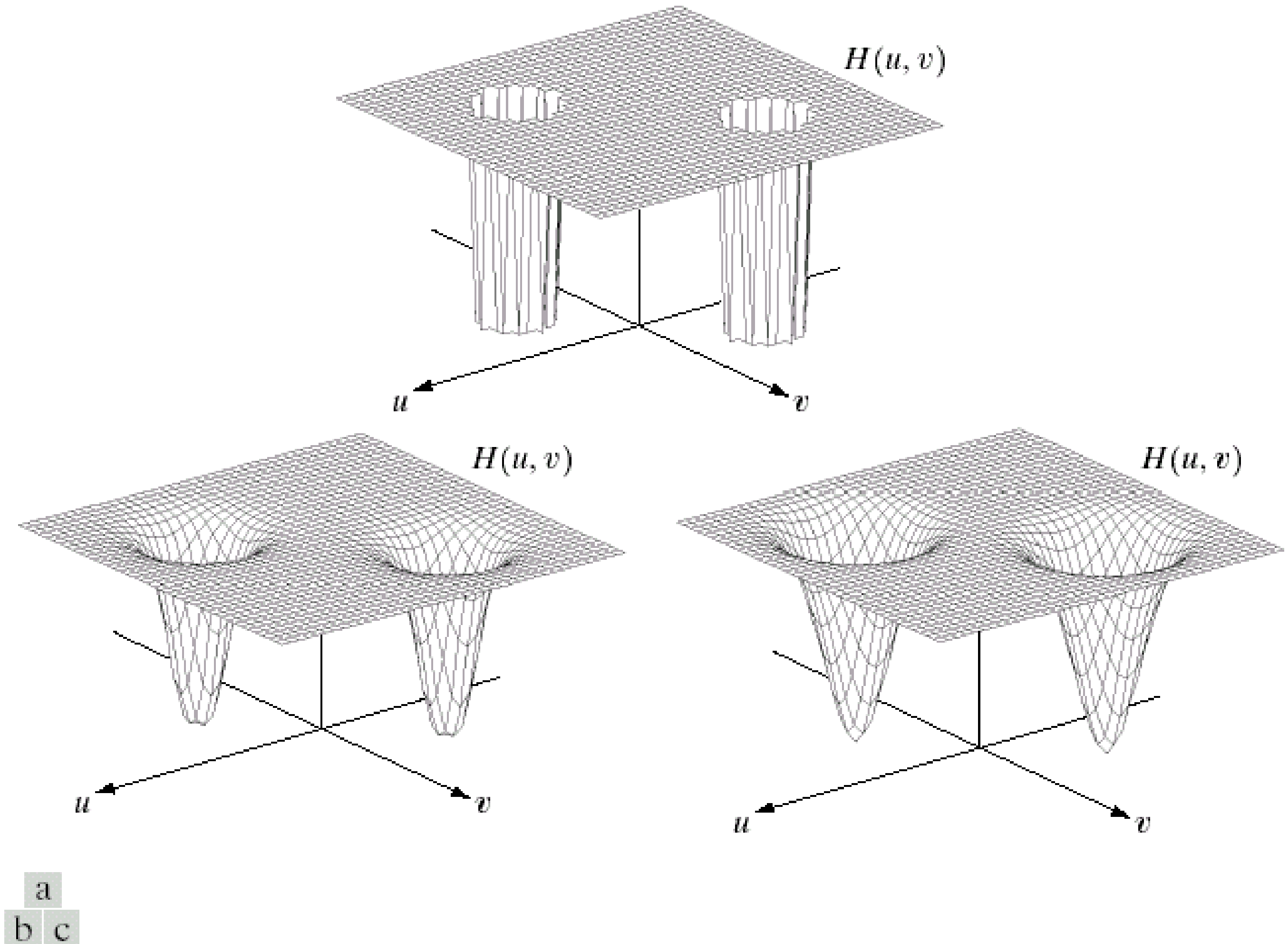
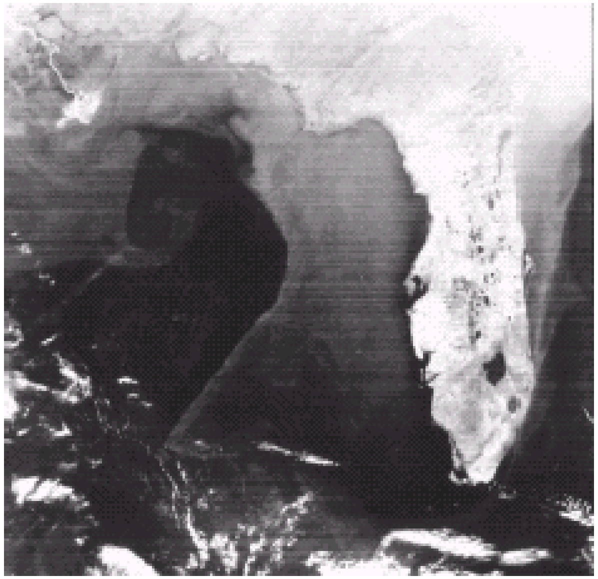
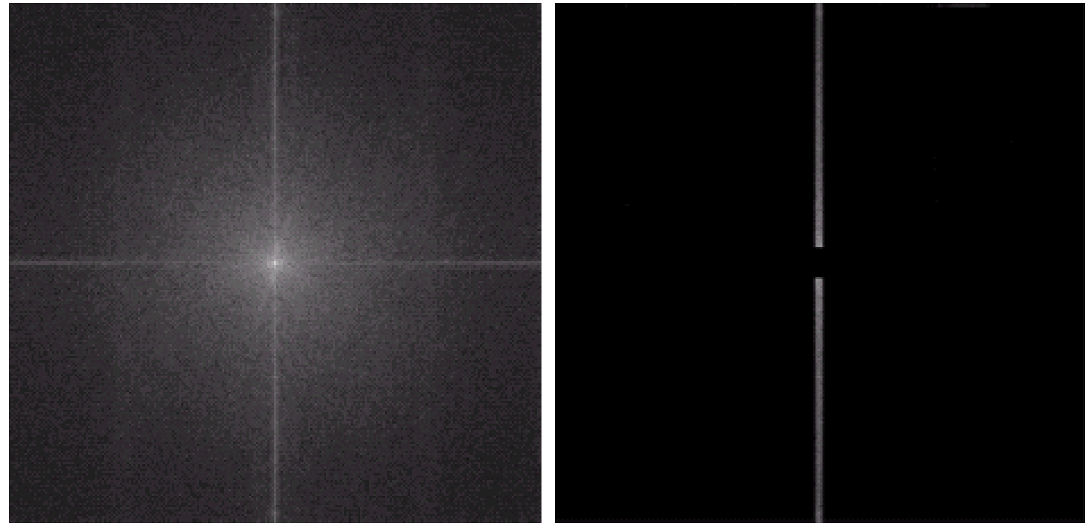


FIGURE 5.18 Perspective plots of (a) ideal, (b) Butterworth (of order 2), and (c) Gaussian notch (reject) filters.

Frequency-selective filters





"Notch-pass" filter:

Hnp(*u*,*v*)=1-*Hnr*(*u*,*v*)

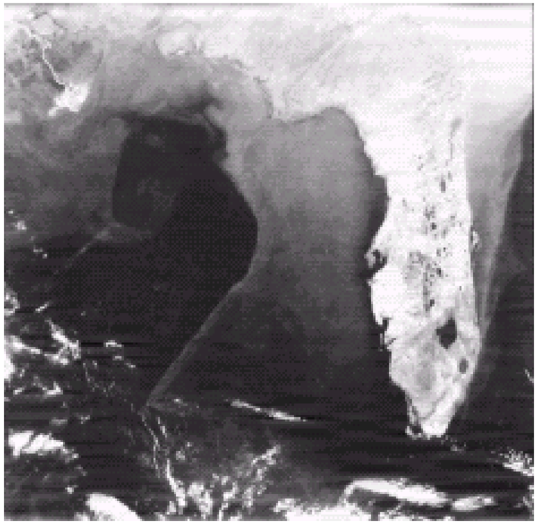


FIGURE 5.19 (a) Satellite image of Florida and the Gulf of Mexico (note horizontal sensor scan lines). (b) Spectrum of (a). (c) Notch pass filter shown superimposed on (b). (d) Inverse Fourier transform of filtered image, showing noise pattern in the spatial domain. (e) Result of notch reject filtering.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

Note: The *notch-pass* filter here is simply a vertical highpass filter

Image Restoration
Image degradation model

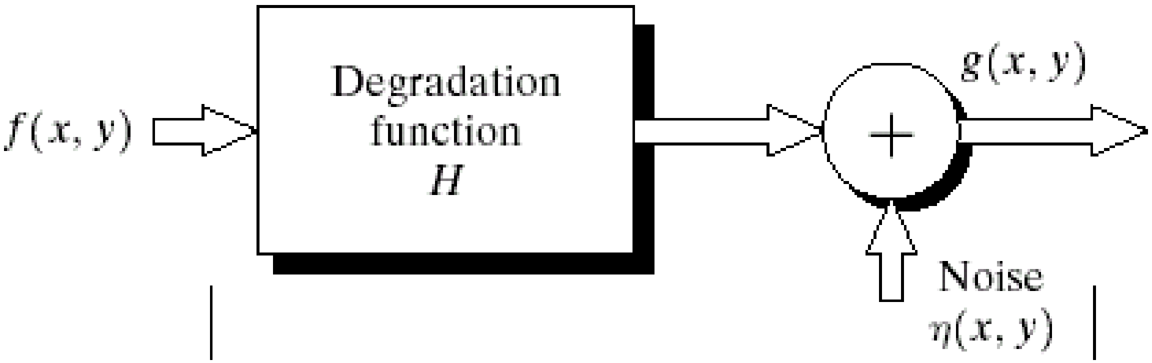


FIGURE 5.1 A model of the image degradation/

Linear, **space-invariant** degradation model with impulse response *h* (point-spread function, *psf*) (note acronyms «*psf*» or «*PSF*» are often used interchangeably in the space and the Fourier domains):

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

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

Note that space invariance is an often unrealistic constraint (think of haze in outdoor scenes, motion of different objects, distance from moving objects...)

Estimating the *PSF*

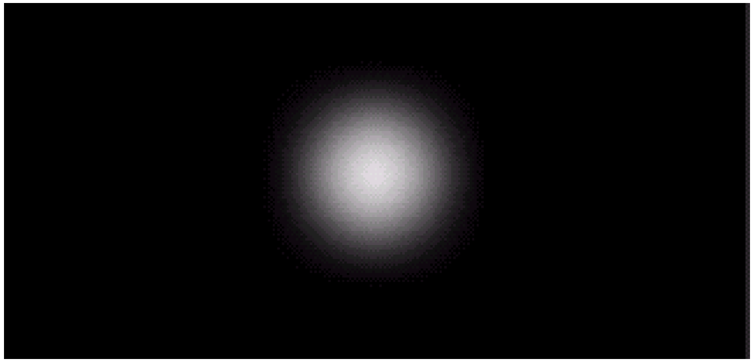
By image observation:

Look at a detail of the degraded image, in a high SNR area, and construct its estimated version f_{est} .Then, $H(u,v) = G(u,v)/F_{est}(u,v)$

By experimentation:

If equipment similar to the one used for acquisition is available, an accurate estimation of the degradation is obtained by imaging an impulse of amplitude *A* using the same system settings. Then,

$$H(u,v) = G(u,v)/A$$



impulse characterization. (a) An impulse of light (shown magnified). (b) Imaged (degraded) impulse.

Estimating the *PSF*

By mathematical modelling:

1. lens defocus

The distance D to an imaged point is related to the parameters of the lens system and the amount of defocus by

$$D = \frac{F v_0}{v_0 - F - \sigma f} \qquad (1)$$

where v_0 is the distance between the lens and the image plane (e.g., the film location in a camera), f the f-number of the lens system, F the focal length of the lens system, and σ the spatial constant of the point spread function (i.e., the radius of the imaged point's "blur circle") which describes how an image point is blurred by the imaging optics. The point spread function may be usefully approximated by a two-dimensional Gaussian $G(r, \sigma)$ with a spatial constant σ and radial distance r. The validity of

NOTE:
Defocus normally is shift-variant

Thin-lens eq. for defocused object

f/# = F / A
f-number = focal length / aperture diameter

Useful also for *depth-from-focus* algorithms

Estimating the *PSF*

2. atmospheric turbulence:

(similar to a Gaussian) $$H(u,v) = \exp[-k(u^2+v^2)^{5/6}]$$

Fig.5.25 (a) Negligible turbulence. (b) Severe turbulence, $k = 0.0025$. (c) Mild turbulence, $k = 0.001$. (d) Low turbulence, $k = 0.00025$.

Note: *H* should be *space-variant* to take into account the different distances of the objects

Estimating the *PSF*

3. linear planar motion, described by the components $x(t)$ and $y(t)$, in the acquisition interval $[0,T]$:

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

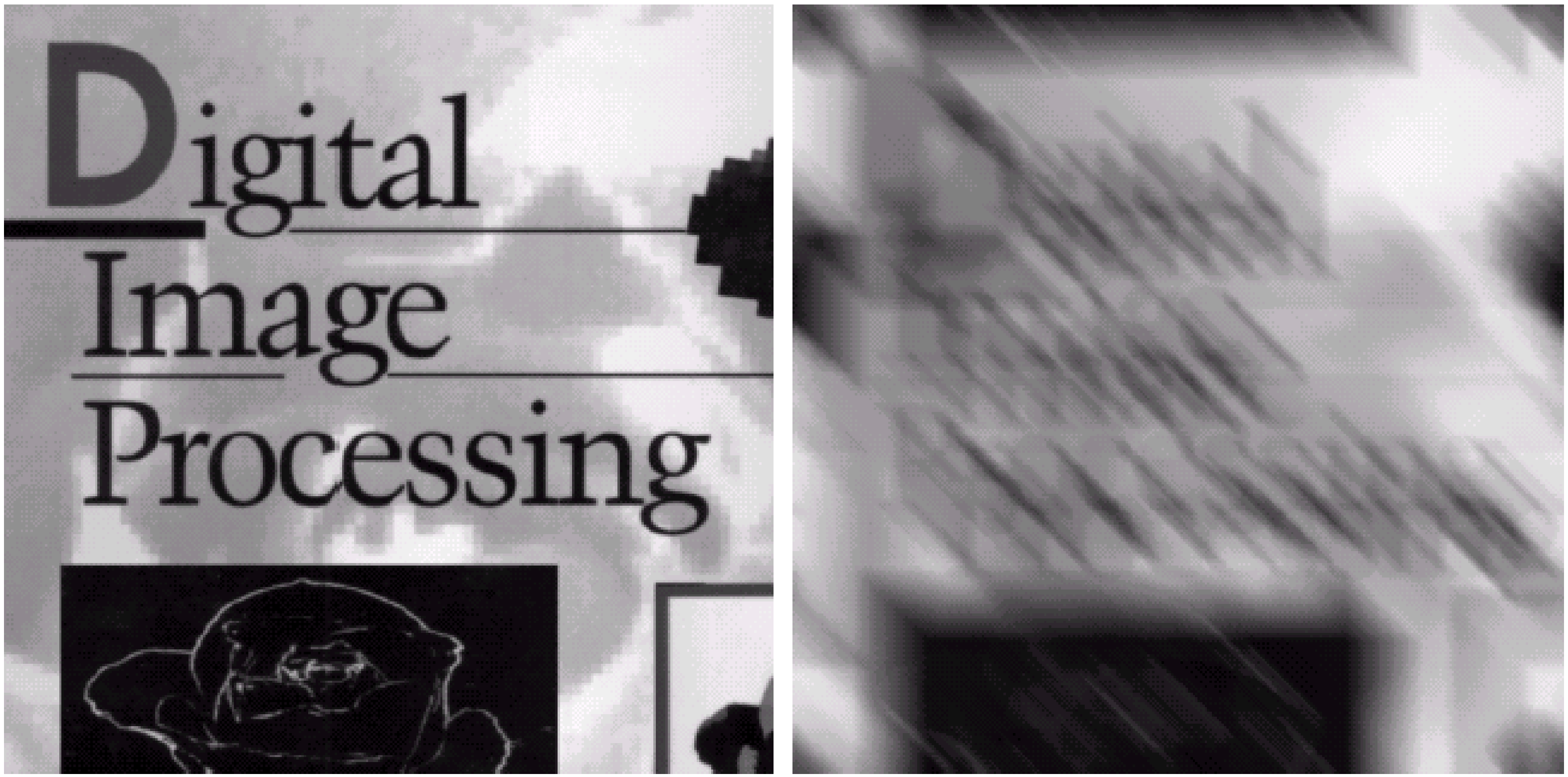
$$G(u,v) = F(u,v)\int_0^T \exp[-j2\pi[ux(t) + vy(t)]]\, dt = F(u,v)H(u,v)$$

e.g. uniform horizontal motion:

$$x(t) = at/T\,; \quad y(t) = 0$$

$$H(u,v) = \int_0^T \exp[-j2\pi\, u\, at/T]\, dt = \frac{T}{\pi\, ua}\sin(\pi\, ua)\exp[-j\pi\, ua]$$

Estimating the *PSF*



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 filter

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

Suppose we know *H*. The most obvious restoration operator is

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

Note: if *H* has zeros on the unit circle or is nonminimum-phase, one can try to *cap* the amplitude of *1/H*

Note: where the degradation is small, noise can dominate. If *H* has zeros close to the unitary circle, 1/*H* has large peaks. This often happens at high frequencies → set 1/*H* = 1 above a given distance from the origin

E.g., starting from Fig.5.25b above (image size 480x480, Butterworth response used to cap 1/*H* at various frequencies):

Inverse filter

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.

Wiener filter

Wiener filter: 1-D derivation [Jain]

Given two zero-mean, stationary random sequences *f(x)* and *g(x)*, having nonzero cross-correlation, it is desired to obtain from *g(x)* a *linear* estimate of *f(x)* that minimizes the MSE:

$$\hat{f}(x) = \sum_{s=-\infty}^{\infty} h_W(x-s)g(s), \qquad e^2 = E\{[f(x)-\hat{f}(x)]^2\} = \min$$

It can be proved (*orthogonality principle*) that the best estimate is obtained from a filter $h_W(k)$ such that:

$$\forall\, x, y \qquad 0 = E\{[f(x)-\hat{f}(x)]g(y)\} = E\{[f(x)-\sum_{s=-\infty}^{\infty} h_W(x-s)g(s)]g(y)\}$$

Using the definition of *cross-correlation* between two real-valued stationary sequences *a(x), b(y)*: $r_{ab}(x-y) = E\{a(x)b(y)\}$

and substituting from above, the orthogonality condition becomes the **Wiener filter equation**:

$$\sum_{s=-\infty}^{\infty} h_W(x-s)\, r_{gg}(s-y) = r_{fg}(x-y)$$

Wiener filter

Taking the Fourier transform we get the *[cross] power spectral densities* and we can solve for H_W:

$$H_W(u)S_{gg}(u) = S_{fg}(u) \quad \rightarrow \quad H_W(u) = S_{fg}(u)S_{gg}^{-1}(u)$$

Now, if the usual model is taken for the corruption mechanism:

$$g(x) = h(x) * f(x) + n(x) \quad \Leftrightarrow \quad G(u) = H(u)F(u) + N(u)$$

the needed power spectra are obtained from:

$$S_{gg}(u) = |H(u)|^2 S_{ff}(u) + S_{nn}(u) \quad ; \quad S_{fg}(u) = H^*(u) S_{ff}(u)$$

And the **Wiener filter** is expressed as: $H_W(u) = \dfrac{H^*(u) S_{ff}(u)}{|H(u)|^2 S_{ff}(u) + S_{nn}(u)}$

Wiener filter

For *nonzero-mean* sequences and zero-mean noise: subtract the average from the available data, perform the filtering, and then add the average back.

2-D Wiener filter

$$H_W(u,v) = \frac{H^*(u,v)\,S_{ff}(u,v)}{|H(u,v)|^2 S_{ff}(u,v) + S_{nn}(u,v)} = \frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + S_{nn}(u,v) / S_{ff}(u,v)}$$

Note: if noise is neglectable at all frequencies, this becomes an inverse filter

Otherwise, since $S_{nn}(u,v)/S_{ff}(u,v) = 1 / \mathrm{SNR}(u,v)$, The Wiener filter

- behaves like an inverse filter at frequencies in which the SNR is high
- tendentially blocks all frequency components at which the SNR is low (whatever the inverse filter)

2-D Wiener filter

E.g.: if white noise and blur are present, the Wiener filter seeks a frequency-by-frequency compromise between

- deblurring → highpass filter and
- attenuation of frequencies at which the SNR is low (typically, high frequencies) → lowpass filter

The result in this case is a bandpass filter.

Note: if the spectrum of the original image cannot be effectively estimated, we can use K = 1/SNR:

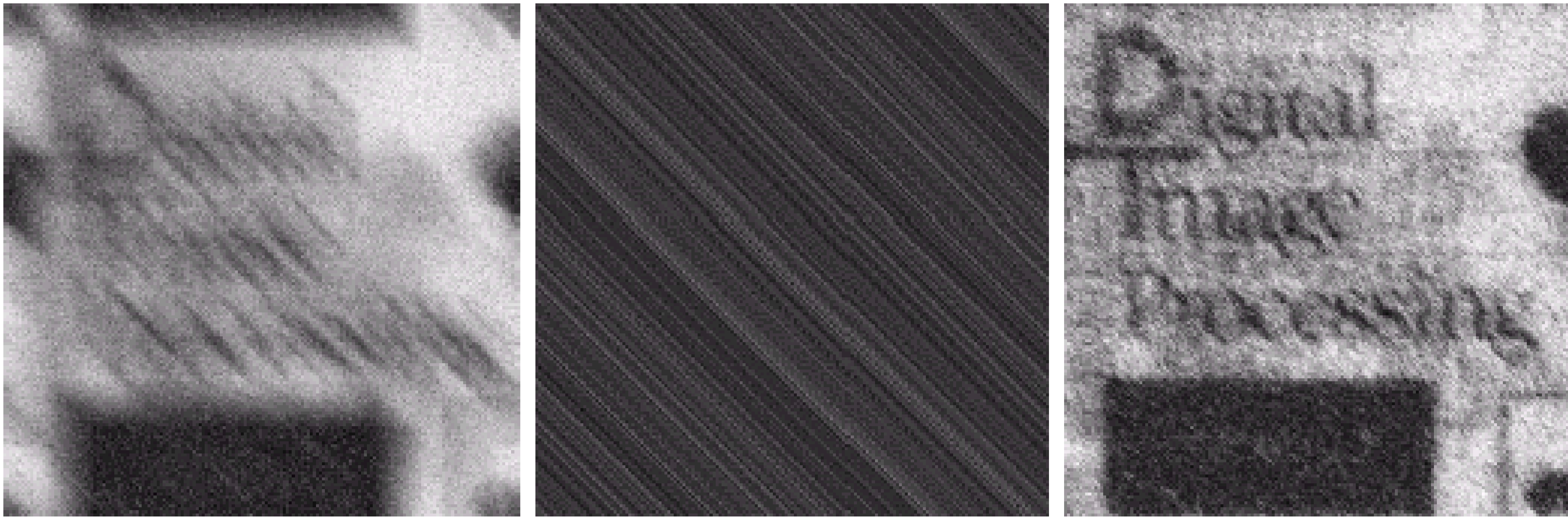
$$H_W(u,v) = \frac{1}{H(u,v)} \frac{|H(u,v)|^2}{|H(u,v)|^2 + K}$$

2-D Wiener filter

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.

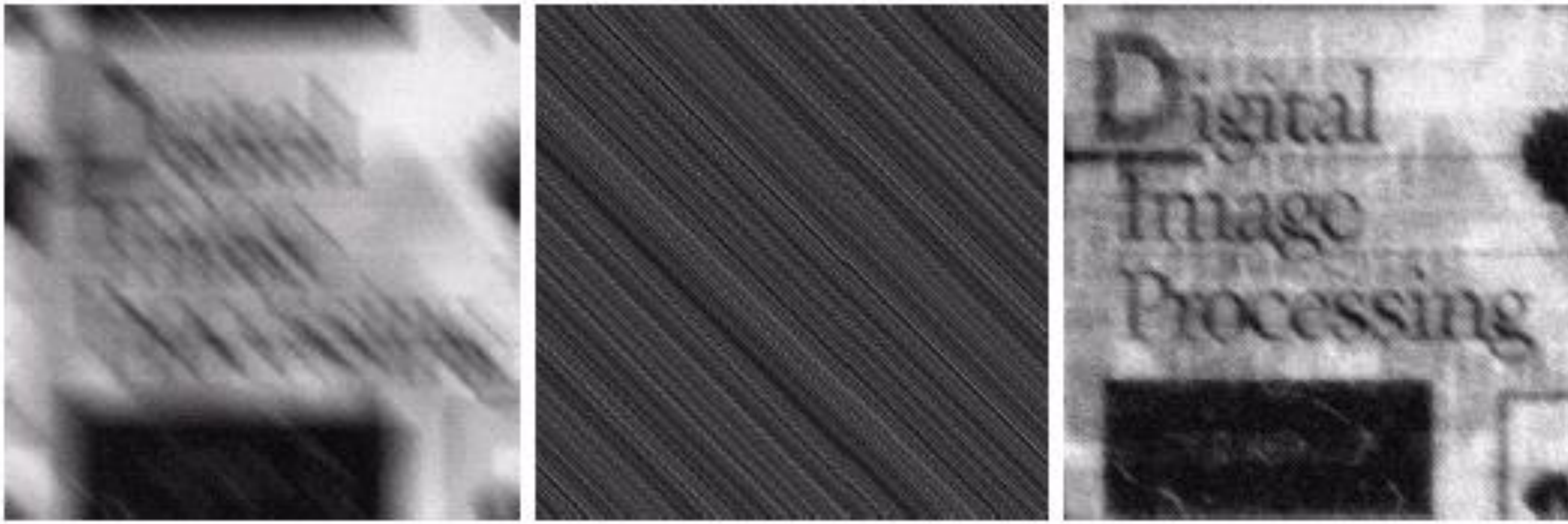
2-D Wiener filter



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.

2-D Wiener filter



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.

2-D Wiener filter

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.

Constrained Least Squares filter

An extension to the simplified Wiener filter seen last. Suppose we know the **noise mean** and **variance**.

Approach: (~Jain) minimize a cost function $J = \| p(x,y) * \hat{f}(x,y) \|^2$

subject to the constraint $\| g(x,y) - h(x,y) * \hat{f}(x,y) \|^2 = \| n(x,y) \|^2$

Normally, *p* is an operator that measures the “roughness” of the estimated image, e.g. its Laplacian:

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

I.e.: we want the output image to be as smooth as possible, but not too different from an estimate of the original

Constrained Least Squares filter

In the Fourier domain: minimize $J = \| P(u,v)\hat{F}(u,v) \|^2$

subject to $\| G(u,v) - H(u,v)\hat{F}(u,v) \|^2 = \| N(u,v) \|^2$

A solution can be obtained via the Lagrange multiplier method: $H_{LS}(u,v) = \dfrac{1}{H(u,v)} \dfrac{|H(u,v)|^2}{|H(u,v)|^2 + \gamma |P(u,v)|^2}$

Gamma can be set iteratively: compute $D = \| G(u,v) - H(u,v)\hat{F}(u,v) \|^2$ then increase gamma if $D < \| N(u,v) \|^2$, decrease it if $D > \| N(u,v) \|^2$

Notes:

- $\| N(u,v) \|^2$ is the noise power (the variance if the noise is zero-mean)
- The LS filter is equivalent to the Wiener filter if we specify:

$$S_{nn}(u,v) = \gamma \quad \text{and} \quad S_{ff}(u,v) = 1 / | P(u,v) |^2$$

Constrained Least Squares filter

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.

- γ values set manually
- LS is better than Wiener when the noise is high or medium

Constrained Least Squares filter

FIGURE 5.31
(a) Iteratively determined constrained least squares restoration of Fig 5.25-b using correct noise parameters.
(b) Result obtained with wrong noise parameters.

Similar to Fig.5.28(c)

Measuring perceived image quality

Mean Squared Error: Love It or Leave It?

SSIM index: Structural SIMilarity

$$S(\mathrm{x},\mathrm{y}) = l(\mathrm{x},\mathrm{y})\cdot c(\mathrm{x},\mathrm{y})\cdot s(\mathrm{x},\mathrm{y}) = \left(\frac{2\mu_x\mu_y + C_1}{\mu_x^2+\mu_y^2+C_1}\right) \cdot \left(\frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2+\sigma_y^2+C_2}\right)\cdot\left(\frac{\sigma_{xy}+C_3}{\sigma_x\sigma_y+C_3}\right), \quad (2)$$

where μ_x and μ_y are (respectively) the local sample means of x and y, σ_x and σ_y are (respectively) the local sample standard deviations of x and y, and σ_{xy} is the sample cross correlation of x and y after removing their means. The items C_1, C_2, and C_3 are small positive constants that stabilize each term, so that near-

Or, using some measurable HVS properties:
From Just Noticeable Differences to Image Quality

Elements of visual perception

The normalized contrast threshold ***nct*** is then

nct(L,S) = 1 / *ncr* = (*L*+*S*)^2 / (*L***S*)

Taking as a reference the perfect adaptation case:

nct(L=S) = (2*S*)^2 / *S*^2 = 4

we get the denormalized contrast threshold ***ct***:

ct(L,S,cto) = (*cto*/4) * *nct* =
= (*cto*/4) * (*L*+*S*)^2 / (*L***S*)

where ***cto*** is the optimal contrast threshold
Barten: *cto* = 0.01
(Ferwerda: *cto* = 0.1)