Image Restoration

FYS-2010-1 25V Image Analysis
by Elisabeth Wetzer

Overview

- **Recap Image Restoration**
- **Recap Denoising**
- **Reconstruction for degradation in absence of noise**
- **Reconstruction for degradation in presence of noise**

Image Restoration

- **Image Enhancement vs Restoration**
  - Enhancement: subjective
  - Restoration: objective w.r.t. some measure, uses a-priori knowledge about degradation
- **Principle**
  - Recover an image that has been degraded and/or is noisy
- **Methods**
  - Often involves optimization to reach optimum w.r.t. the chosen goodness criterion
  - Filtering (spatial and in frequency)

Image Restoration

- **Spatial Domain**
  - $g(x, y) = h(x, y) * f(x, y) + \eta(x, y)$
- **Frequency Domain**
  - $G(u, v) = H(u, v)F(u, v) + N(u, v)$
- **Assumption**
  - Position invariant system/filter
  - Linear system

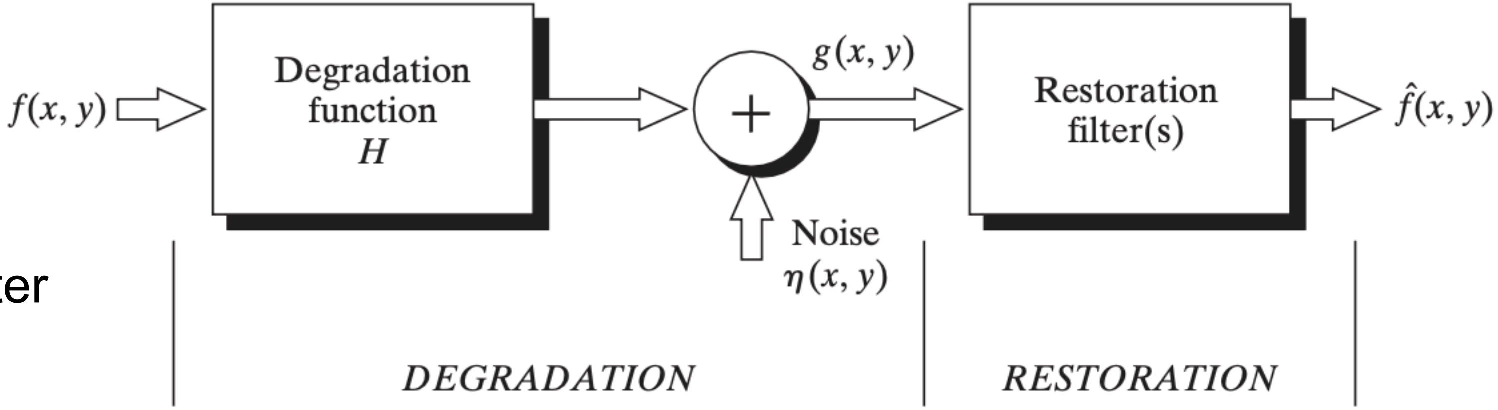


Image Restoration

- **Spatial Domain**
  - $g(x, y) = h(x, y) * f(x, y) + \boxed{\eta(x, y)}$

Additive Noise Terms

- **Frequency Domain**
  - $G(u, v) = H(u, v)F(u, v) + \boxed{N(u, v)}$

- **Assumption**
  - Position invariant system/filter
  - Linear system

Image Restoration

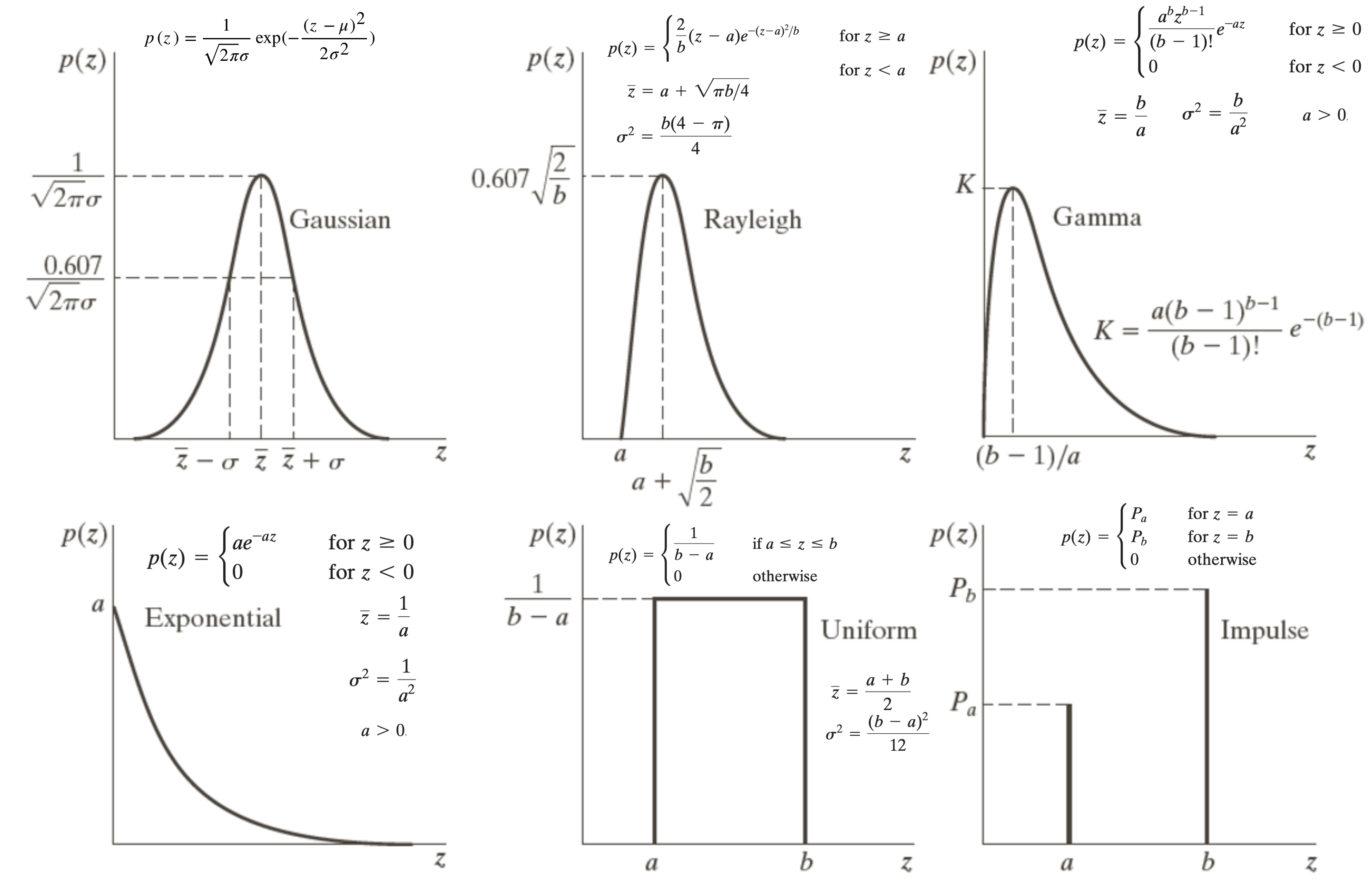
- **Spatial Domain**
  - $g(x, y) = \boxed{h(x, y)} * f(x, y) + \eta(x, y)$

Degradation Function

- **Frequency Domain**
  - $G(u, v) = \boxed{H(u, v)} F(u, v) + N(u, v)$

- **Assumption**
  - Position invariant system/filter
  - Linear system

Common Random Noise Models



Common Random Noise Models

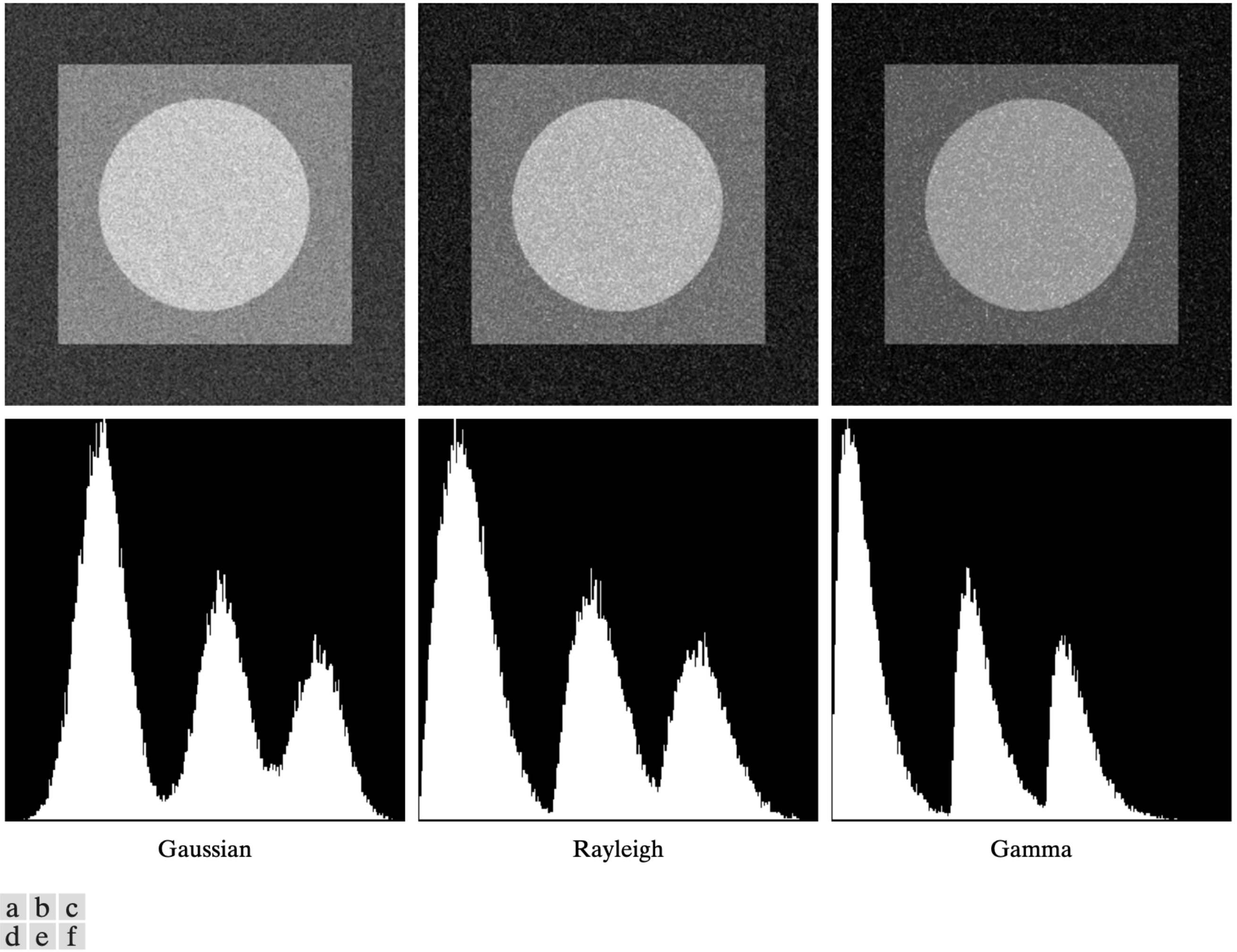


FIGURE 5.4 Images and histograms resulting from adding Gaussian, Rayleigh, and gamma noise to the image in Fig. 5.3.

How to find a suitable noise model?

- **Find a homogeneous area in an image**
- **Assess histogram**
- **Estimate the mean and variance**

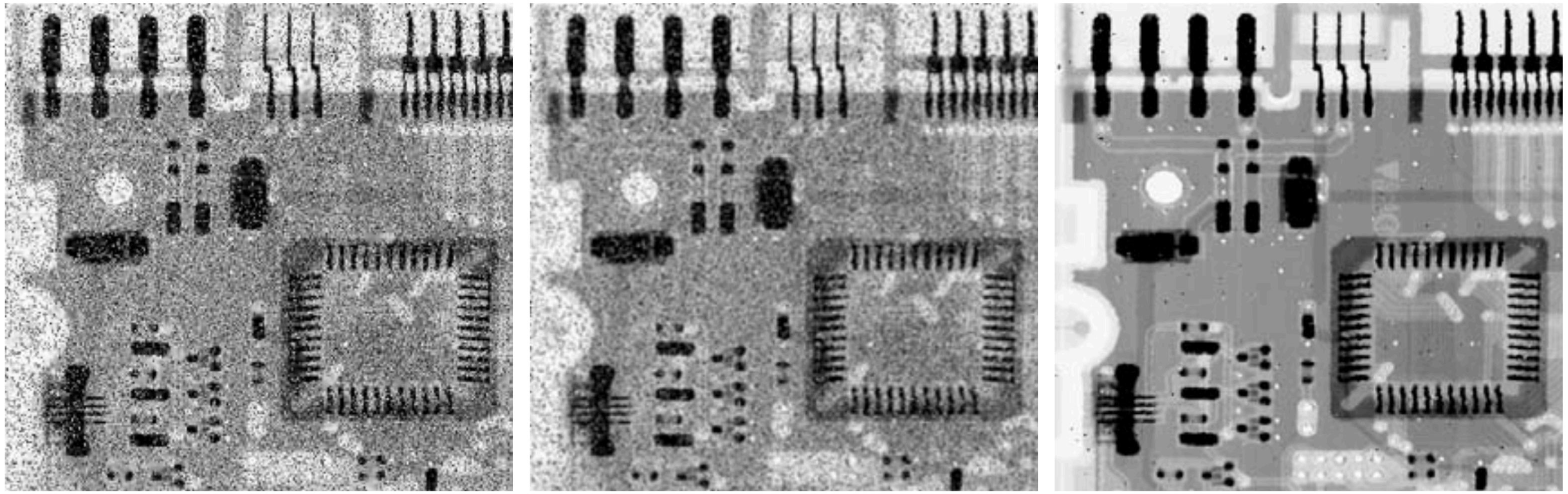
Spatial Filters

- **Mean vs Median Filter**

1	2	3	5
2	1	3	4
7	9	255	1
8	1	2	1
0	3	3	2

	31	31	
	32	31	
	32	31	

	3	3	
	3	2	
	3	2	



a b c

FIGURE 3.35 (a) X-ray image of circuit board corrupted by salt-and-pepper noise. (b) Noise reduction with a 3 × 3 averaging mask. (c) Noise reduction with a 3 × 3 median filter. (Original image courtesy of Mr. Joseph E. Pascente, Lixi, Inc.)

Spatial Filters: Mean Filters

Type of Mean Filter	Mathmatical Expression	Usecases
Arithmetic Mean	$\hat{f}(x,y) = \frac{1}{mn} \sum_{(s,t)\in S_{xy}} g(s,t)$	Noise reduction by blurring **Good** for: Gaussian & uniform noise **Bad** for: Salt & Pepper
Geometric Mean	$\hat{f}(x,y) = \left(\prod_{(s,t)\in S_{xy} g(s,t)} \right)^{\frac{1}{mn}}$	Similar as arithmetic mean, but preserves more detail
Harmonic Mean	$\hat{f}(x,y) = \frac{mn}{\sum_{(s,t)\in S_{xy}} \frac{1}{g(s,t)}}$	Works for salt noise, not for pepper noise. Works for Gaussian noise.
Contraharmonic Mean of Order Q	$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}}$	Good for Salt & Pepper noise if Q>0 for pepper noise if Q<0 for salt noise

Spatial Filters: Order-Statistic Filters

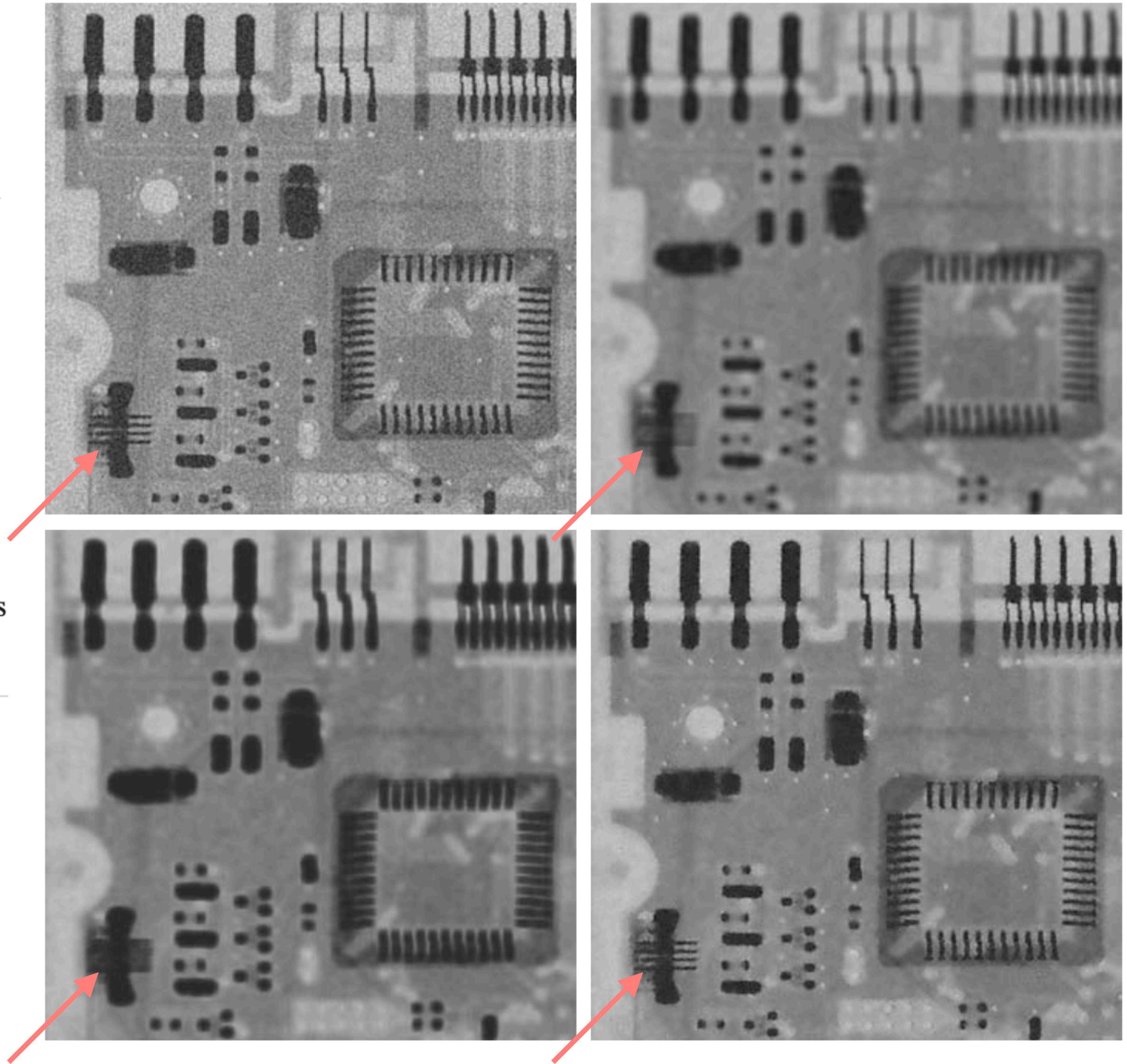
Type of Mean Filter	Mathmatical Expression	Usecases
Median	$\hat{f}(x,y) = \text{median}_{(s,t)\in S_{xy}}\{g(s,t)\}$	Less blur than mean filtering Good for bipolar and unipolar impulse noise
Max	$\hat{f}(x,y) = \max_{(s,t)\in S_{xy}}\{g(s,t)\}$	Good for pepper noise
Min	$\hat{f}(x,y) = \min_{(s,t)\in S_{xy}}\{g(s,t)\}$	Good for salt noise
Midpoint	$\hat{f}(x,y) = \frac{1}{2}\left(\max_{(s,t)\in S_{xy}}\{g(s,t)\} + \min_{(s,t)\in S_{xy}}\{g(s,t)\}\right)$	Good for Gaussian or uniform noise
Alpha-trimmed mean filter	$\hat{f}(x,y) = \frac{1}{mn-\alpha}\sum_{(s,t)\in S_{xy}} g(s,t)$	Good for combination of salt-and-pepper and Gaussian noise

Spatial Filtering: Adaptive Local Noise Filtering

- **Adaptive Filters**
  - Denoising by mean filtering may be useful in homogeneous areas, but less so close to edges
  - Adjust filter behavior based on statistical characteristics of the image inside the filter region
  - Often better performance, but computationally more expensive
  - Reasonable to consider mean and variance of a neighborhood for filter adaptation because they are important parameters also for filter design

a b
c d

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.



Spatial Filtering: Adaptive Local Noise Filtering

- **Setting**
  - Local region S_{xy} and for a point (x, y) in S_{xy}, $g(x, y)$ is the value of the noisy image at (x, y),
  - σ_η^2 is the variance of the noise corrupting $f(x, y)$ to $g(x, y)$,
  - μ_L is the local mean and σ_L^2 the local variance of the pixels in S_{xy}
- **Filter**
  - If $\sigma_\eta^2 = 0$, filter should return $g(x, y)$ as then $g(x, y) = f(x, y)$
  - If $\sigma_L^2 \gg \sigma_\eta^2$, it implies an edge and the filter should return a value close to $g(x, y)$
  - If $\sigma_L^2 \approx \sigma_\eta^2$, the filter should act as a mean filter and return the arithmetic mean

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

Spatial Filtering: Adaptive Median Filtering

- **Setting**
  - Local region S_{xy} and for a point (x, y) in S_{xy}, $g(x, y)$ is the value of the noisy image at (x, y),
  - z_{min} minimum intensity, z_{max} maximum intensity, z_{med} median intensity value in S_{xy}
  - z_{xy} intensity value at (x, y)
  - S_{max} maximum allowed size of S_{xy}
- **Filter in 2 stages:**

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

Stage B:
$B1 = z_{xy} - z_{\text{min}}$
$B2 = z_{xy} - z_{\text{max}}$
If $B1 > 0$ AND $B2 < 0$, output z_{xy}
Else output z_{med}

Spatial Filtering: Adaptive Median Filtering

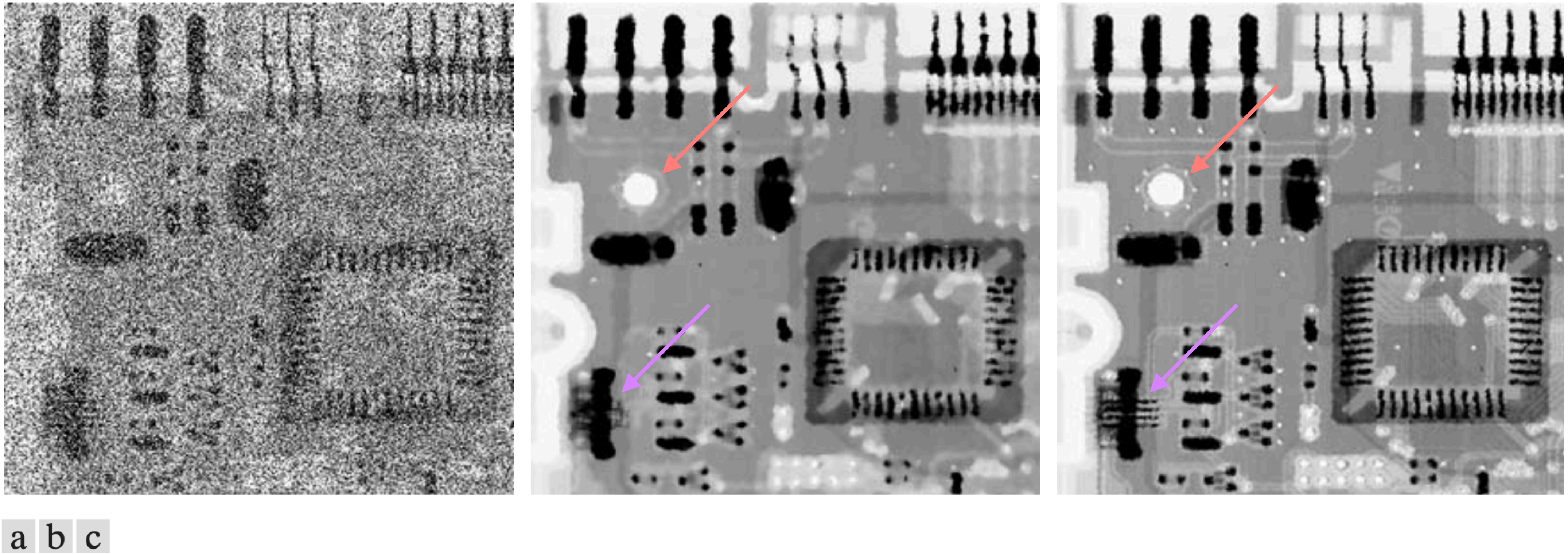
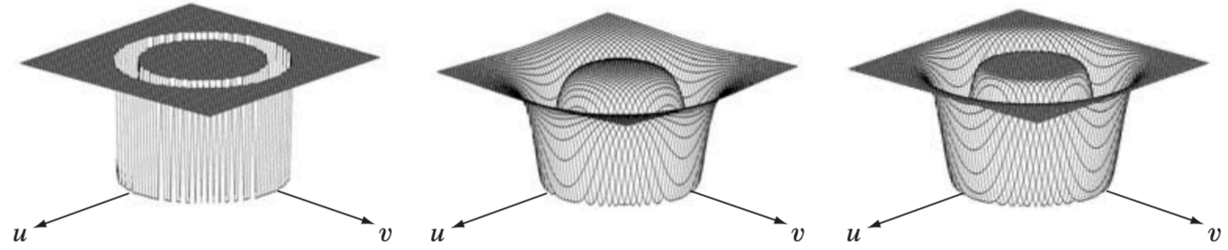


FIGURE 5.14 (a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.25$. (b) Result of filtering with a 7×7 median filter. (c) Result of adaptive median filtering with $S_{max} = 7$.

Frequency Domain Filtering: Band Reject Filter

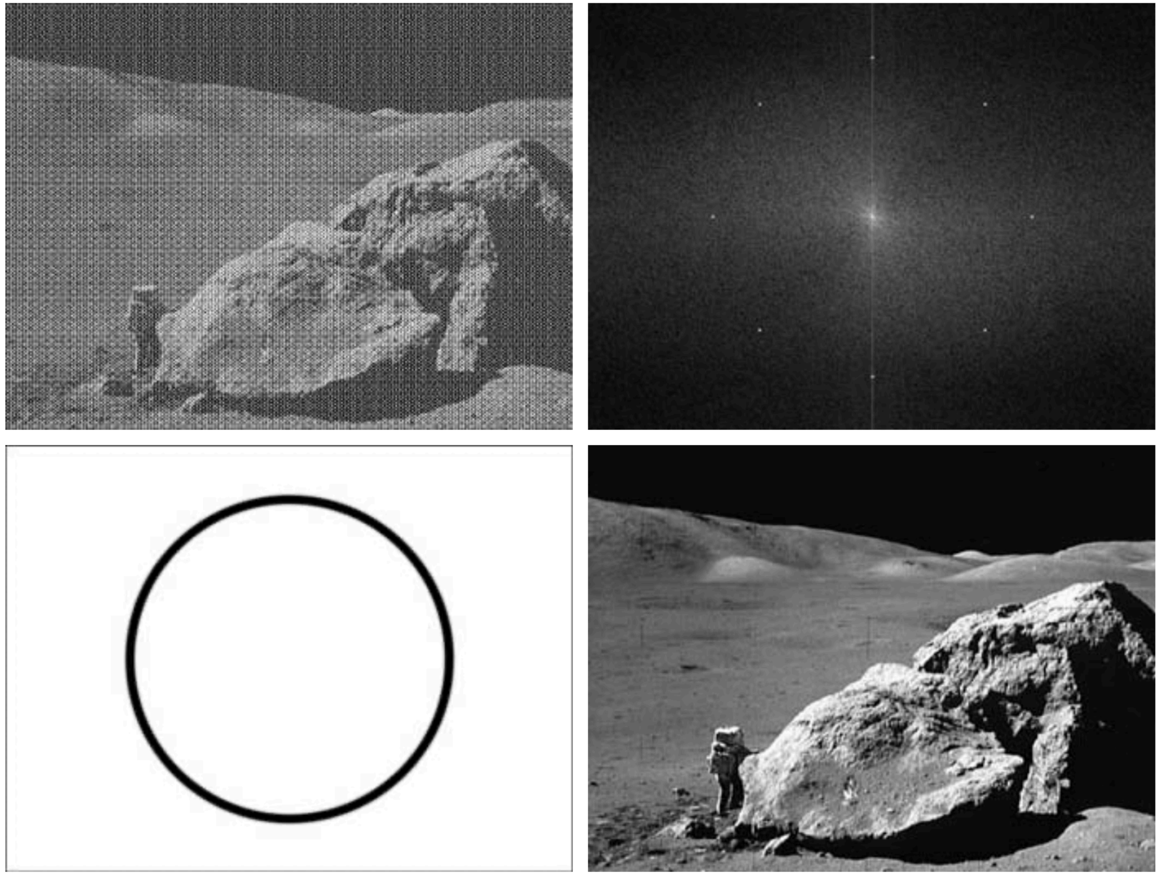


a b c

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

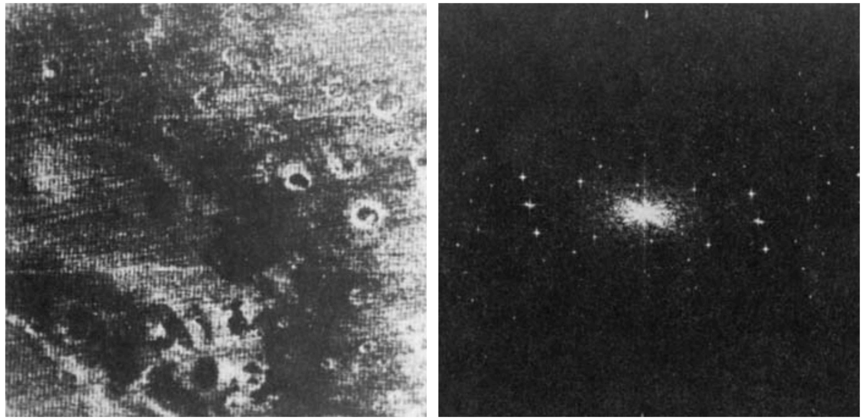
TABLE 4.6
Bandreject filters. W is the width of the band, D is the distance $D(u, v)$ from the center of the filter, D_0 is the cutoff frequency, and n is the order of the Butterworth filter. We show D instead of $D(u, v)$ to simplify the notation in the table.

Ideal	Butterworth	Gaussian
$H(u,v) = \begin{cases} 0 & \text{if } D_0 - \frac{W}{2} \le D \le D_0 + \frac{W}{2} \\ 1 & \text{otherwise} \end{cases}$	$H(u,v) = \dfrac{1}{1 + \left[\dfrac{DW}{D^2 - D_0^2}\right]^{2n}}$	$H(u,v) = 1 - e^{-\left[\frac{D^2 - D_0^2}{DW}\right]^2}$



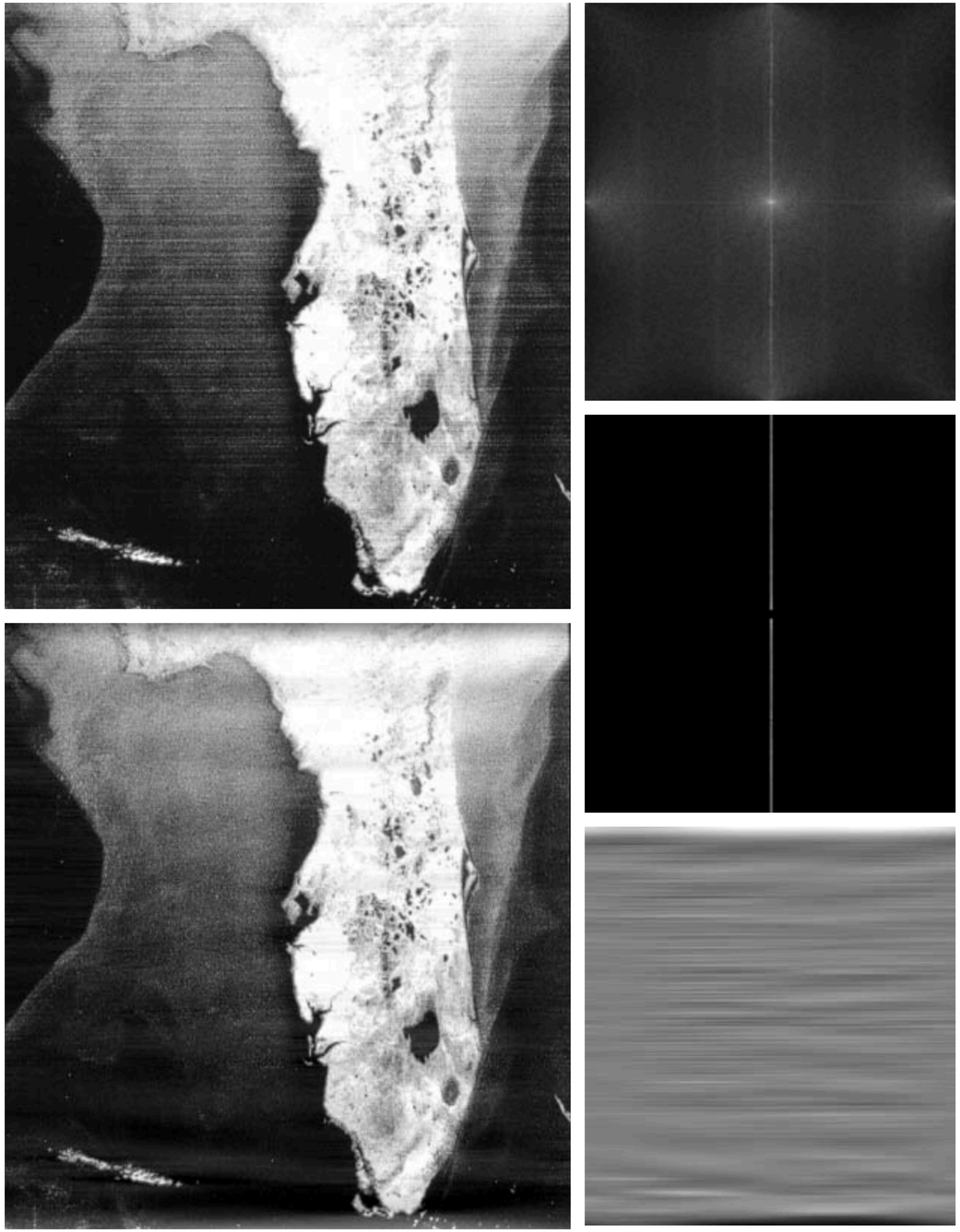
Notch Filter

- Band Pass/Reject would not be possible in this case
- Notch has to be symmetric around origin
- Can also be used to isolate noise pattern



a b

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



a b
c
e d

FIGURE 5.19
(a) Satellite image of Florida and the Gulf of Mexico showing horizontal scan lines. (b) Spectrum. (c) Notch pass filter superimposed on (b). (d) Spatial noise pattern. (e) Result of notch reject filtering. (Original image courtesy of NOAA.)

Frequency Domain Filtering: Optimum-Notch

- Isolate principal contributions of inference pattern by placing a notch filter at the location of each spike
- Subtract a variable, weighted portion of the pattern from the corrupted image, i.e.

$$\hat{f}(x, y) = g(x, y) - w(x, y)\eta(x, y)$$

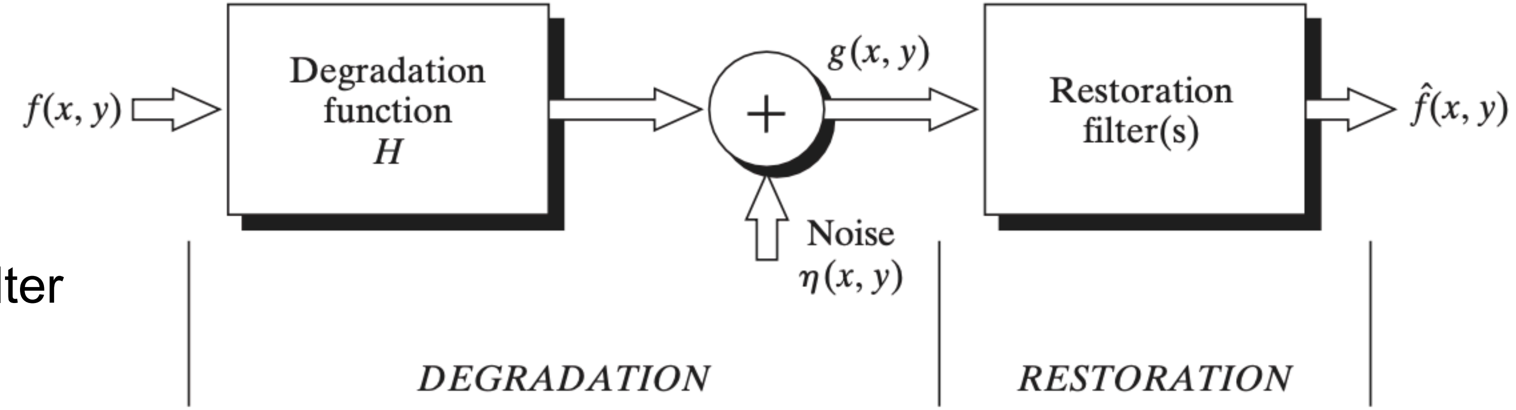
- $w(x, y)$ is *weighting* or *modulation function* and is to be determined by optimization
- One approach for optimization: minimize the local variance of the estimate $\hat{f}(x, y)$

Summary: Denoising

- **Image Restoration**
  - Aim: Recover original signal after degradation, today only noise models were considered
- **Find out something about this noise**
  - If possible, find a homogeneous patch in the image and look at the histogram of intensities there, try to read out the type of noise in the histogram
  - Inspect the frequency domain
- **Apply best suiting filtering technique to the image based on that noise model**
  - Periodic Noise -> Frequency domain
  - Salt & Pepper -> Order-Statistic Filters
  - Adaptive Filters can improve outcome by preserving edges, but come at computational cost

Image Restoration

- **Spatial Domain**
  - $g(x, y) = h(x, y) * f(x, y) + \eta(x, y)$
- **Frequency Domain**
  - $G(u, v) = H(u, v)F(u, v) + N(u, v)$
- **Assumption**
  - Position invariant system/filter
  - Linear system



Linear, Position-Invariant Degradations

- **Assumption**
  - Position invariant system/filter
  - Linear system
- **Advantage**
  - Many types of degradations can be approximated by linear, position-invariant processes
  - Many tools available
- **Degradation Model**
  - Result of convolution
  - Linear image restoration often called image deconvolution

Estimating the Degradation Function

- **Estimation by Image Observation**
  - Select noise-free region find local degradation
- **Estimation by Experimentation**
  - Replicate imaging settings for impulse
- **Estimation by Modeling**
  - Assume mathematical model

- **“Blind Deconvolution”**

Estimating H: Image Observation

- **Assumption**
  - H is a a linear, position-invariant process

- **Gather Information from Image**

Estimating H: Image Observation

- **Assumption**
  - H is a a linear, position-invariant process

- **Gather Information from Image**
  - Inspect subimage $g_s(x, y)$ in which the signal is strong (high contrast) s.t. we can neglect any noise

Estimating H: Image Observation

- **Assumption**
  - H is a a linear, position-invariant process

- **Gather Information from Image**
  - Inspect subimage $g_s(x, y)$ in which the signal is strong (high contrast) s.t. we can neglect any noise
  - Then we know for the estimated subimage that

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

Estimating H: Image Observation

- **Assumption**
  - H is a a linear, position-invariant process

- **Gather Information from Image**
  - Inspect subimage $g_s(x, y)$ in which the signal is strong (high contrast) s.t. we can neglect any noise
  - Then we know for the estimated subimage that

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

  - Due to position-invariance we can derive $H(u, v)$ from $H_s(u, v)$

Estimating H: Experimentation

- **Assumption**
  - H is a a linear, position-invariant process

- **Experiments**

Estimating H: Experimentation

- **Assumption**
  - H is a a linear, position-invariant process

- **Experiments**
  - If imaging equipment used to acquire the degraded image is available, the corresponding image settings have to be found experimentally

Estimating H: Experimentation

- **Assumption**
  - H is a a linear, position-invariant process

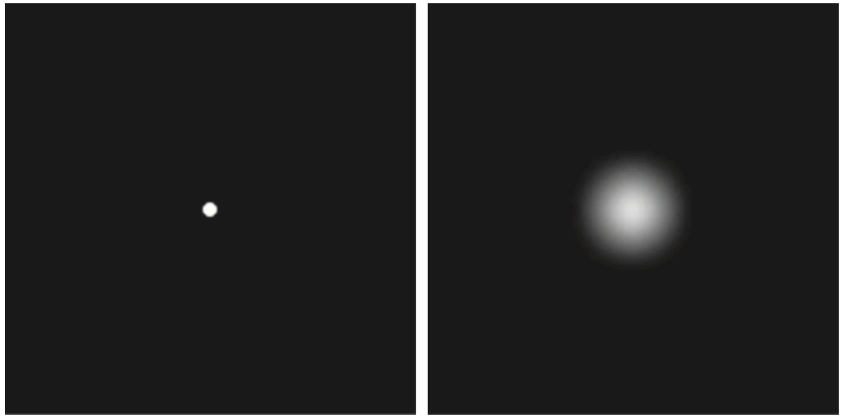
- **Experiments**
  - If imaging equipment used to acquire the degraded image is available, the corresponding image settings have to be found experimentally
  - With those settings, an impulse can be imaged

Estimating H: Experimentation

- **Assumption**
  - H is a a linear, position-invariant process

a b

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



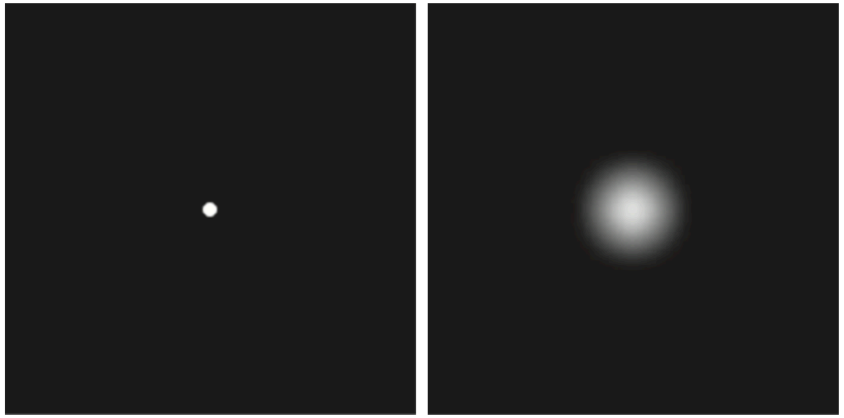
- **Experiments**
  - If imaging equipment used to acquire the degraded image is available, the corresponding image settings have to be found experimentally
  - With those settings, an impulse can be imaged
  - Will result in a point spread function - an impulse response

Estimating H: Experimentation

- **Assumption**
  - H is a a linear, position-invariant process

a b

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



- **Experiments**
  - If imaging equipment used to acquire the degraded image is available, the corresponding image settings have to be found experimentally
  - With those settings, an impulse can be imaged
  - Will result in a point spread function - an impulse response
  - This characterizes the degradation completely because it is a linear & position-invariant

Estimating H: Modeling

- **Mathematical/Physical Model**
  - Blurred by uniform linear motion between the image and the sensor during image acquisition

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

  - Physical characteristics of atmospheric turbulence $H(u,v) = e^{-k(u^2 + v^2)^{5/6}}$

Estimating H: Modeling

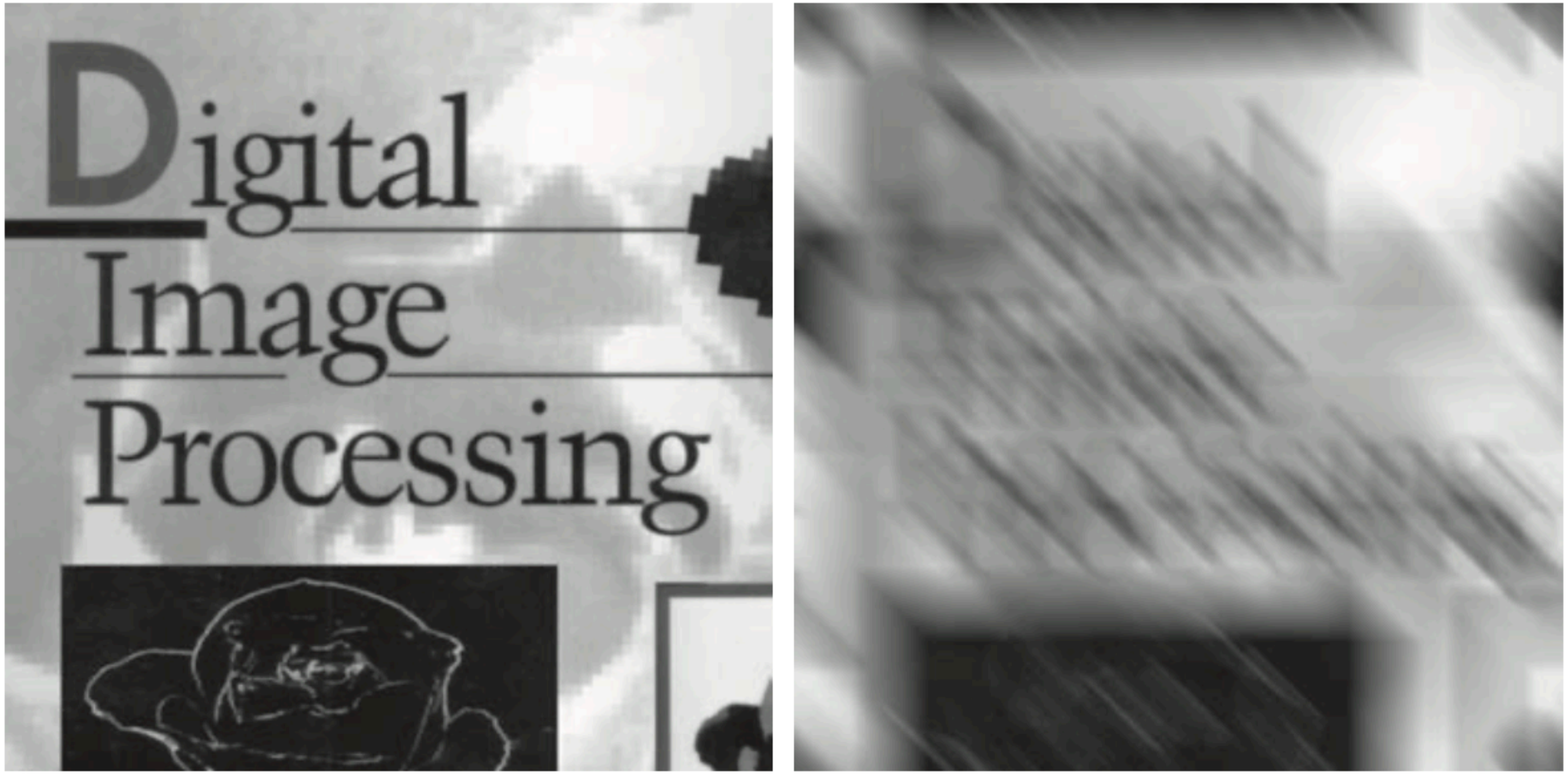
- **Mathematical/Physical Model**
  - Blurred by uniform linear motion between the image and the sensor during image acquisition
  - $f(x, y)$ undergoes planar motion, $x_0(t)$ and $y_0(t)$ time-varying components of motion in the x- and y-directions, respectively
  - Assume perfect imaging process (shutter opens and closes instantaneously)
  - T is exposure time

Estimating H: Modeling

- **Mathematical/Physical Model**
  - Blurred by uniform linear motion between the image and the sensor during image acquisition
  - $f(x, y)$ undergoes planar motion, $x_0(t)$ and $y_0(t)$ time-varying components of motion in the x- and y-directions, respectively
  - Assume perfect imaging process (shutter opens and closes instantaneously)
  - T is exposure time
  - We get blurred image $g(x, y)$

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

Estimating H: Modeling



a b

FIGURE 5.26 (a) Original image. (b) Result of blurring using the function in Eq. (5-77) with $a = b = 0.1$ and $T = 1$.

Estimating H: Modeling

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

The continuous Fourier transform of this expression is

$$G(u, v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x, y)\, e^{-j2\pi(ux + vy)}\, dx\, dy$$

Estimating H: Modeling

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

The continuous Fourier transform of this expression is

$$G(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} g(x,y)e^{-j2\pi(ux + vy)}\,dx\,dy$$

$$G(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left[\int_0^T f[x - x_0(t),\ y - y_0(t)]\,dt\right]e^{-j2\pi(ux + vy)}\,dx\,dy$$

Estimating H: Modeling

$$G(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\left[\int_0^T f[x - x_0(t), y - y_0(t)]dt\right]e^{-j2\pi(ux+vy)}dx\,dy$$

Reversing the order of integration results in the expression

$$G(u,v) = \int_0^T \underbrace{\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f[x - x_0(t), y - y_0(t)]e^{-j2\pi(ux+vy)}dx\,dy\right]}dt$$

Fourier transform of the displaced function $f[x - x_0(t), y - y_0(t)]$

Estimating H: Modeling

$$G(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \left[\int_0^T f[x - x_0(t), y - y_0(t)]\,dt \right] e^{-j2\pi(ux+vy)}\,dx\,dy$$

Reversing the order of integration results in the expression

$$G(u,v) = \int_0^T \underbrace{\left[\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f[x - x_0(t), y - y_0(t)]\, e^{-j2\pi(ux+vy)}\,dx\,dy \right]}\,dt$$

Fourier transform of the displaced function $f[x - x_0(t), y - y_0(t)]$

3)	Translation (general)	$f(x,y)e^{j2\pi(u_0x/M + v_0y/N)} \Leftrightarrow F(u-u_0, v-v_0)$ $f(x-x_0, y-y_0) \Leftrightarrow F(u,v)e^{-j2\pi(ux_0/M + vy_0/N)}$

Estimating H: Modeling

$$G(u,v) = \int_0^T \left[\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f[x - x_0(t), y - y_0(t)] e^{-j2\pi(ux + vy)} dx\, dy \right] dt$$

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

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

Estimating H: Modeling

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

By defining

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

we can express Eq. (5-73) in the familiar form

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

Estimating H: Modeling

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

- Suppose uniform linear motion in the x-direction only, i.e. $y_0(t) = 0$ at a rate of $x_0(t) = at/T$, i.e. when $t = T$ the image has been displaced by a total distance a

$$H(u,v) = \int_0^T \mathrm{e}^{-j2\pi u x_0(t)}\,dt = \int_0^T \mathrm{e}^{-j2\pi uat/T}\,dt$$

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

Estimating H: Modeling

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

- If we allow y-component to vary with a motion given by $y_0(t) = bt/T$, we get

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

Estimating H: Modeling

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

- If we allow y-component to vary with a motion given by $y_0(t) = bt/T$, we get

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

To generate a discrete filter transfer function of size $M \times N$, we sample this equation for $u = 0, 1, 2, \dots, M-1$ and $v = 0, 1, 2, \dots, N-1$.

Restoration

- **Now we have some tools to find out something about H**

- **How do we recover the true, clean image?**

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

Inverse Filtering

- **Assumption**
  - H is now known based on one of the steps from before
  - Simplest restoration approach: devide FFT of degraded image by degradation transfer function:

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

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

Inverse Filtering

- **Assumption**
  - H is now known based on one of the steps from before
  - Simplest restoration approach: devide FFT of degraded image by degradation transfer function:

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

  - putting it into definition $G(u, v) = H(u, v)F(u, v) + N(u, v)$, we get

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

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

Inverse Filtering

- **Assumption**
  - H is now known based on one of the steps from before
  - Simplest restoration approach: devide FFT of degraded image by degradation transfer function:

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

  - putting it into definition $G(u,v) = H(u,v)F(u,v) + N(u,v)$, we get

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

1.) even if H is known, we cannot recover the undegraded image because $N(u,v)$ is unknown
2.) If H has small values, noise term will dominate

Inverse Filtering: Example

- **Degradation**
  - Image blur by air turbulance $H(u,v) = e^{-k\left[(u+M/2)^2+(v-N/2)^2\right]^{5/6}}$
    with $k = 0.0025$, $M/2$ and $N/2$ offset values, $M = N = 480$

Inverse Filtering: Example

- **Degradation**
  - Image blur by air turbulance $H(u,v) = e^{-k\left[(u + M/2)^2 + (v - N/2)^2\right]^{5/6}}$
    with $k = 0.0025$, $M/2$ and $N/2$ offset values, $M = N = 480$

$f(x,y)$

$g(x,y)$

Inverse Filtering: Example

- **Degradation**
  - Image blur by air turbulance $H(u,v) = e^{-k\left[(u + M/2)^2 + (v - N/2)^2\right]^{5/6}}$
    with $k = 0.0025$, $M/2$ and $N/2$ offset values, $M = N = 480$
  - Gaussian has no zeros - yay!

$f(x, y)$

$g(x, y)$

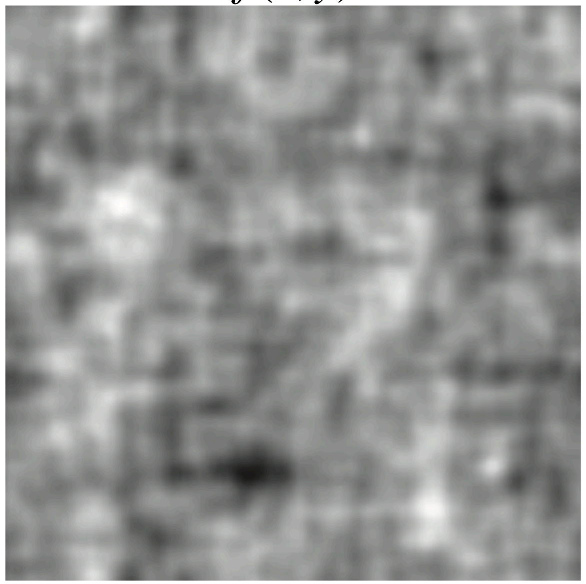
Inverse Filtering: Example

- **Degradation**
  - Image blur by air turbulance $H(u,v) = e^{-k\left[(u + M/2)^2 + (v - N/2)^2\right]^{5/6}}$
    with $k = 0.0025$, $M/2$ and $N/2$ offset values, $M = N = 480$
  - Gaussian has no zeros - yay!

$f(x,y)$

$g(x,y)$

$\hat{f}(x,y)$



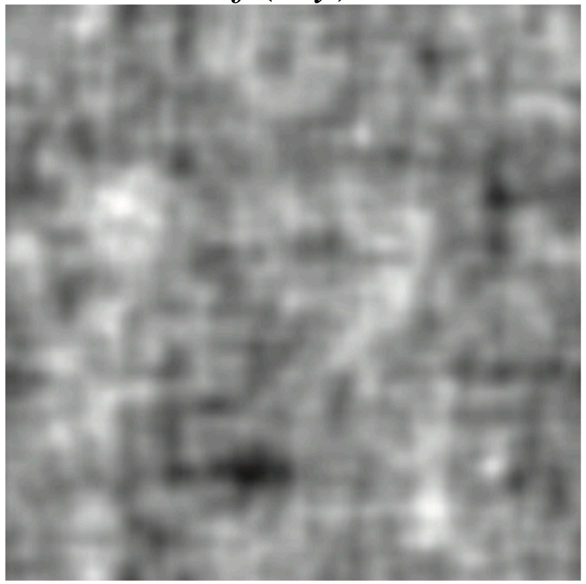
Inverse Filtering: Example

- **Degradation**
  - Image blur by air turbulance $H(u,v) = e^{-k\left[(u\ +\ M/2)^2 + (v\ -\ N/2)^2\right]^{5/6}}$
    with $k = 0.0025$, $M/2$ and $N/2$ offset values, $M = N = 480$
  - Gaussian has no zeros - yay! -> But still many small values - nay!

$f(x,y)$

$g(x,y)$

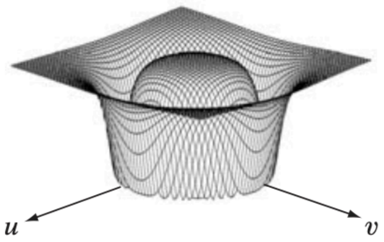
$\hat{f}(x,y)$



Inverse Filtering: Example

- **Cut-off**
  - cutting off values of $\dfrac{G(u,v)}{H(u,v)}$ outside a radius
- **Butterworth Filter**

$$\frac{1}{1+\left[\dfrac{DW}{D^2-D_0^2}\right]^{2n}}$$



a b
c d

FIGURE 5.27 Restoring Fig. 5.25(b) using Eq. (5-78). (a) Result of using the full filter. (b) Result with H cut off outside a radius of 40. (c) Result with H cut off outside a radius of 70. (d) Result with H cut off outside a radius of 85.

Wiener Filter

- **Minimum Mean Square Error Filter, Least Square Error Filter**

- **Considers $H(u, v)$ and $N(u, v)$**
  - first approach we look at that does that

- **Principle**
  - Consider images and noise as random variables
  - Find $\hat{f}$ such that MSE between f and $\hat{f}$ is minimal

Wiener Filter

- **Error Measure**
  - $e = \text{MSE}[(\hat{f} - f)^2]$
- **Assumption**
  - Image and noise uncorrelated
  - One of them has zero mean
  - Intensity levels of $\hat{f}$ are a linear function of intensity levels in g

Wiener Filter

- **Error Measure**
  - $e = \text{MSE}[(\hat{f} - f)^2]$
- **Assumption**
  - Image and noise uncorrelated
  - One of them has zero mean
  - Intensity levels of $\hat{f}$ are a linear function of intensity levels in g
- **Minimize error**

Wiener Filter

- **Minimal Error**

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

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

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

  - Uses that $|H(u,v)|^2 = H(u,v)H^*(u,v)$
  - $S_\eta(u,v) = |N(u,v)|^2$ power spectrum of the noise
  - $S_f(u,v) = |F(u,v)|^2$ power spectrum of the undegraded image

Quality Measures

- **Signal-to-Noise Ratio (SNR)**
  - measure of the level of information-bearing signal power to the level of noise power

$$\mathrm{SNR} = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\left|F(u,v)\right|^2 \Bigg/ \sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\left|N(u,v)\right|^2$$

Quality Measures

- **Signal-to-Noise Ratio (SNR)**
  - measure of the level of information-bearing signal power to the level of noise power

$$\mathrm{SNR} = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\left|F(u,v)\right|^2 \Bigg/ \sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\left|N(u,v)\right|^2$$

- **MSE**
  - Summation of squared difference between the original and restored images

$$\mathrm{MSE} = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[f(x,y) - \hat{f}(x,y)\right]^2$$

Quality Measures

- **Signal-to-Noise Ratio (SNR)**
  - measure of the level of information-bearing signal power to the level of noise power

$$\mathrm{SNR} = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\left|F(u,v)\right|^2 \Bigg/ \sum_{u=0}^{M-1}\sum_{v=0}^{N-1}\left|N(u,v)\right|^2$$

$$\mathrm{SNR} = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\hat{f}(x,y)^2 \Bigg/ \sum_{x=0}^{M-1}\sum_{y=0}^{M-1}\left[f(x,y)-\hat{f}(x,y)\right]^2$$

- **MSE**
  - Summation of squared difference between the original and restored images

$$\mathrm{MSE} = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1}\left[f(x,y)-\hat{f}(x,y)\right]^2$$

Wiener Filter

- **Minimal Error**

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

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

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

- Uses that $|H(u,v)|^2 = H(u,v)H^*(u,v)$
- $S_\eta(u,v) = |N(u,v)|^2$ power spectrum of the noise
- $S_f(u,v) = |F(u,v)|^2$ power spectrum of the undegraded image

often unknown
-> approximate

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

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.

Wiener Filter

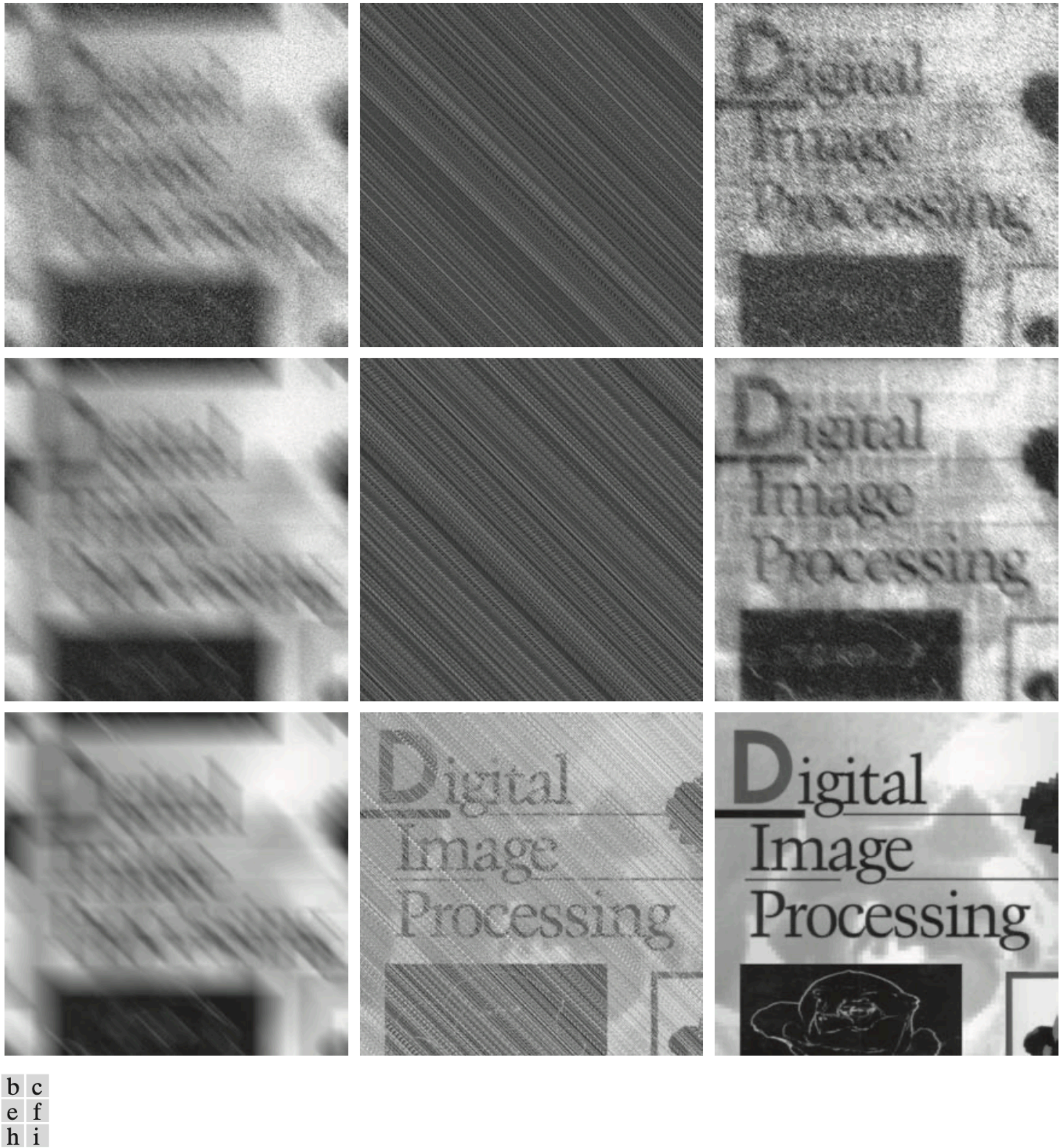


FIGURE 5.29 (a) 8-bit image corrupted by motion blur and additive noise. (b) Result of inverse filtering. (c) Result of Wiener filtering. (d)–(f) Same sequence, but with noise variance one order of magnitude less. (g)–(i) Same sequence, but noise variance reduced by five orders of magnitude from (a).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

Geometric Mean Filter

- **Generalization of the Wiener Filter**

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

  - $\alpha, \beta \in \mathbb{R}^+$
  - For $\alpha = 1$ it is inverse filter
  - For $\alpha = 0$ it is the parametric Wiener filter (for $\beta = 1$ the standard Wiener Filter)

Wiener Filter

- **Wiener Filter**
  - Powerfull
  - But H must be known
  - And even power spectra of η and undegraded image must be known

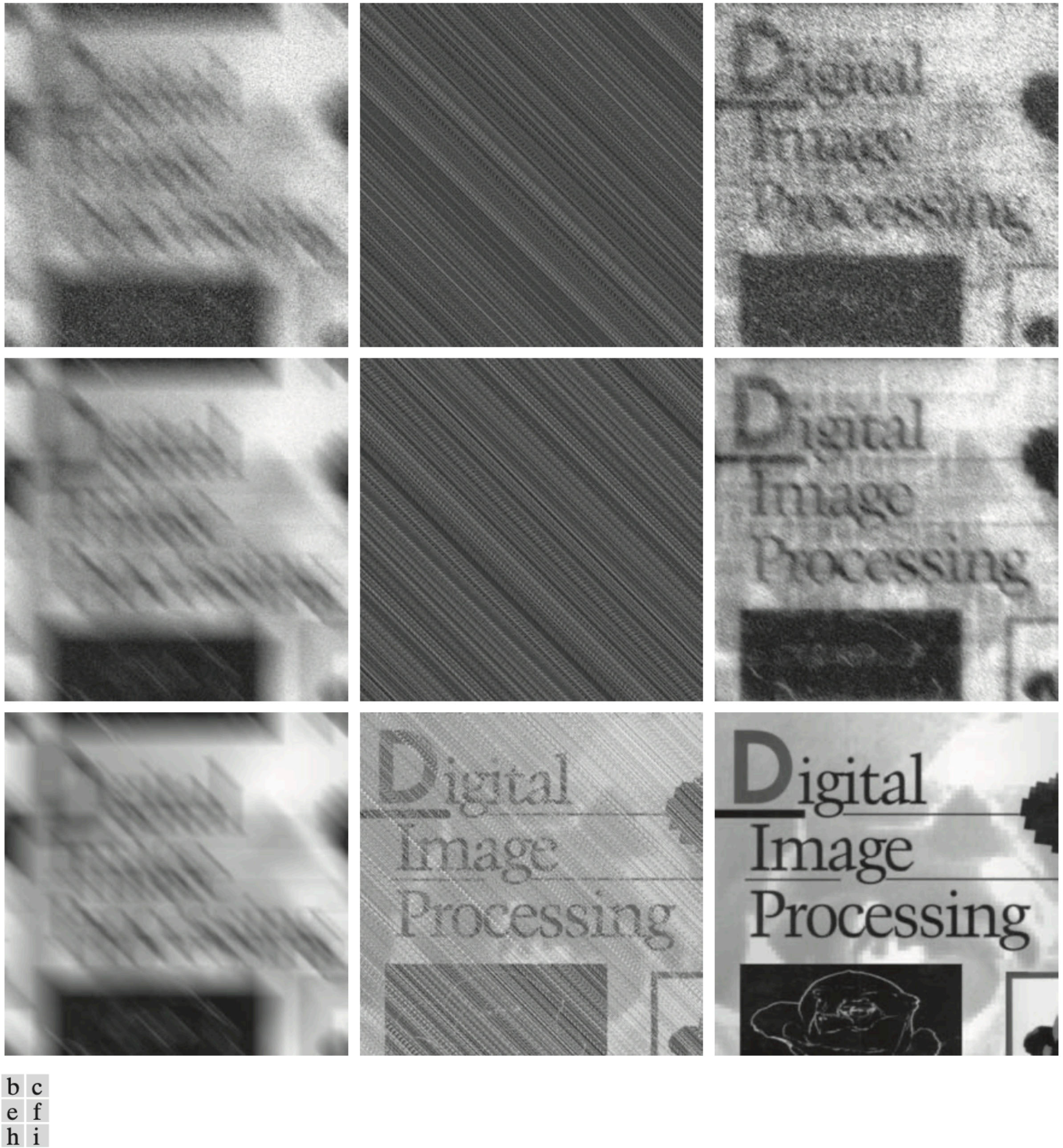


FIGURE 5.29 (a) 8-bit image corrupted by motion blur and additive noise. (b) Result of inverse filtering. (c) Result of Wiener filtering. (d)–(f) Same sequence, but with noise variance one order of magnitude less. (g)–(i) Same sequence, but noise variance reduced by five orders of magnitude from (a).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

Constrained Least Squares Filtering

- **Assumption**
  - Only mean and variance of the noise are known

Constrained Least Squares Filtering

- **Assumption**
  - Only mean and variance of the noise are known

- **Vector-Matrix form**
  - Flatten the image row-by-row, then, $\mathbf{g}, \eta$ are of size $MN \times 1$, and $\mathbf{H}$ of $MN \times MN$

$$\mathbf{g} = \mathbf{H}\mathbf{f} + \boldsymbol{\eta}$$

Constrained Least Squares Filtering

- **Optimization**
  - Minimize

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

subject to the constraint

$$\left\| \mathbf{g} - \mathbf{H}\hat{\mathbf{f}} \right\|^2 = \left\| \boldsymbol{\eta} \right\|^2$$

  - Frequency Domain Solution

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

Constrained Least Squares Filtering

- **Optimization**
  - Minimize

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

subject to the constraint

$$\left\| \mathbf{g} - \mathbf{H}\hat{\mathbf{f}} \right\|^2 = \left\| \boldsymbol{\eta} \right\|^2$$

  - Frequency Domain Solution

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

find through optimization!

Summary

- **Noise η only**
  - Spatial Filters (Mean, Oder-Statistic, Adaptive Filters)
  - Frequency Filters (Bandpass, Notch, Optimum-Notch)
  - Find noise model by inspecting a homogeneous patch in the image
- **Degradation H only**
  - Estimate H (observation, experimentation, modeling)
  - Inverse Filtering
- **Degradation H and Noise η**
  - Wiener Filter: good guess for H and estimate/know power spectra for η and undegraded image
  - Constrained Least Squares: find mean and variance of η and optimization