Image Restoration and Filtering

Noise

Image Noise: http://en.wikipedia.org/wiki/Image_noise
"***Image noise*** *is random (not present in the object imaged) variation of brightness or color information in images, and is usually an aspect of electronic noise.*"


Noise clearly visible in an image from a digital camera

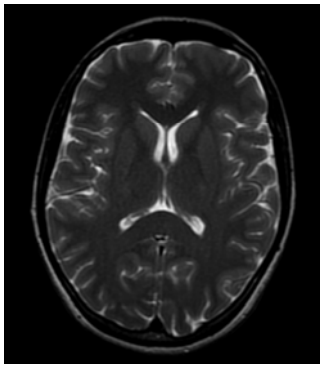
Image Noise:
http://en.wikipedia.org/wiki/Image_noise

- Noise level in CCD-Camera based acquisition can be affected by the light levels, sensor temperature, etc.

- Noise level in magnetic resonance based acquisition can be affected by magnetic field strength, temperature, etc.


Image with salt and pepper noise

- Images are also corrupted during transmission due to interference in the transmission channel.



Normal axial T2-weighted MR image of the brain.
https://en.wikipedia.org/wiki/Magnetic_resonance_imaging_of_the_brain#/media/File:Brain-T2-axial.png

Noise

- We assume that noise (except the periodic noise) is independent of spatial coordinates, and that it is uncorrelated with respect to the image itself.

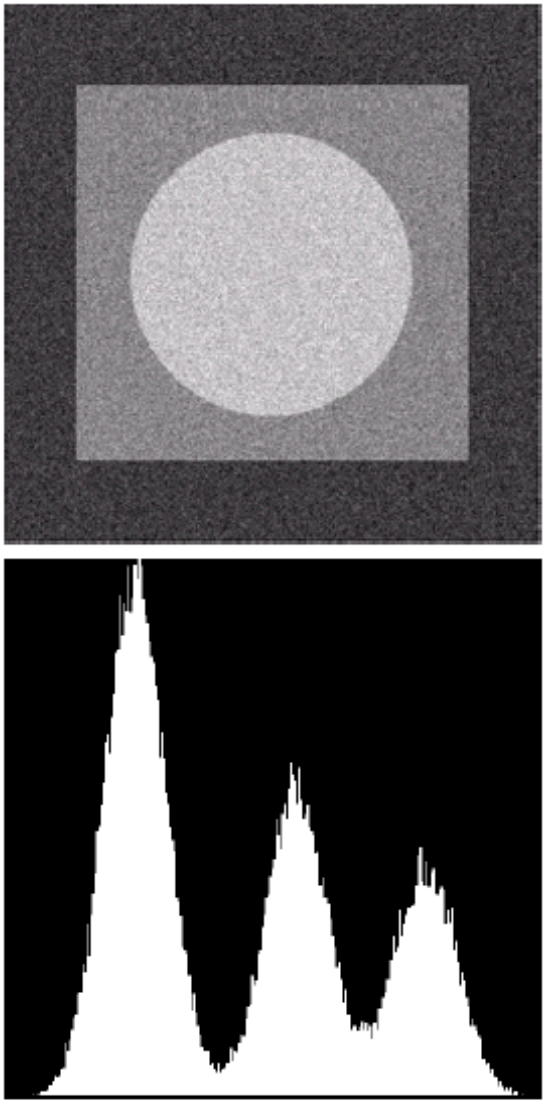
- Also, in some cases, noise can be modelled statistically.

- The statistical behaviour of the grey-level values in the noise component can be described by some probability density functions (PDF). The PDF is identical for all pixels and the noise signal is random.

Gaussian Random Noise

- Also called Normal noise.
- z = random variable
- μ = mean
- σ = standard derivation

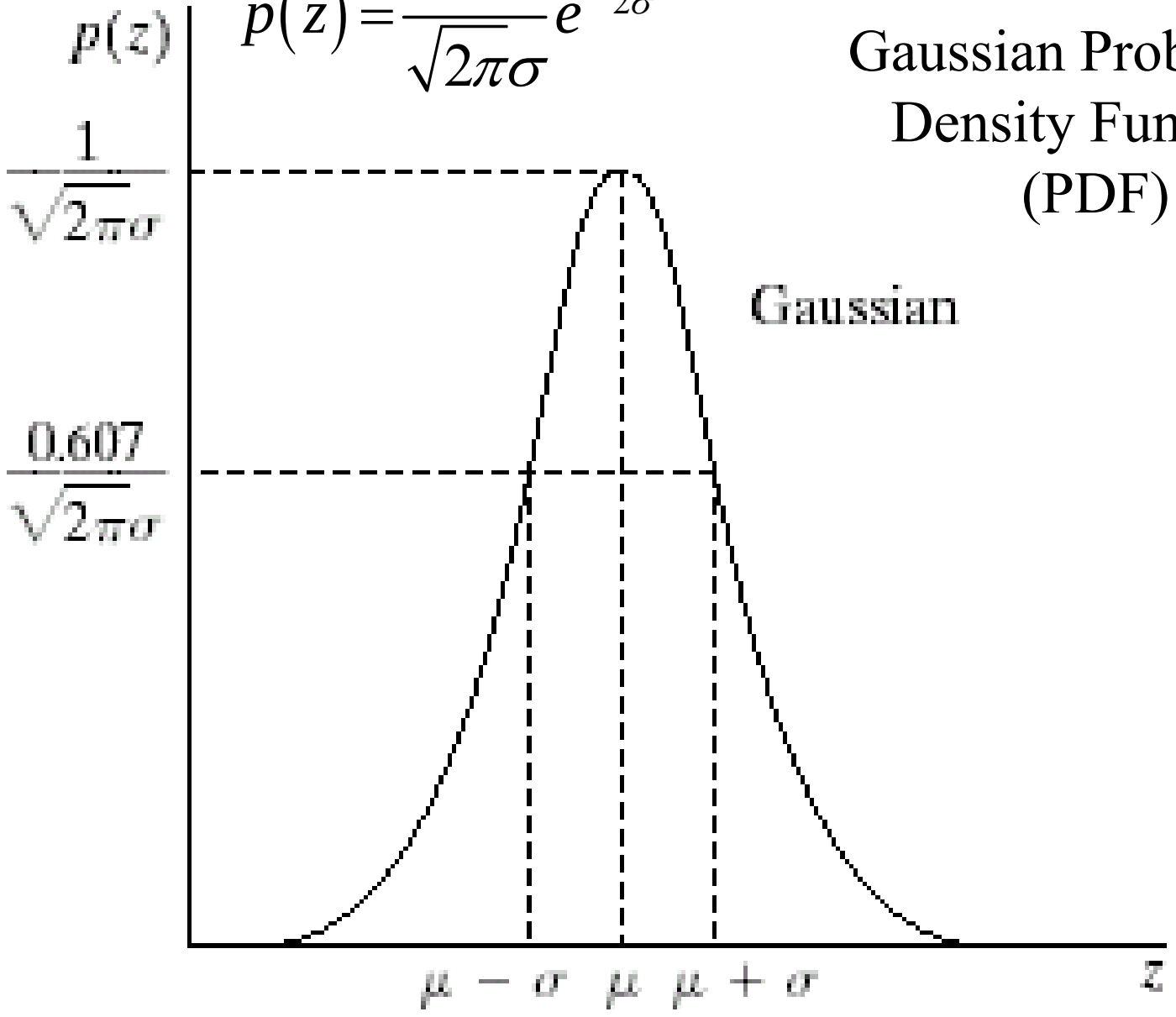
$$p(z) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-(z-\mu)^2}{2\sigma^2}}$$



Gaussian

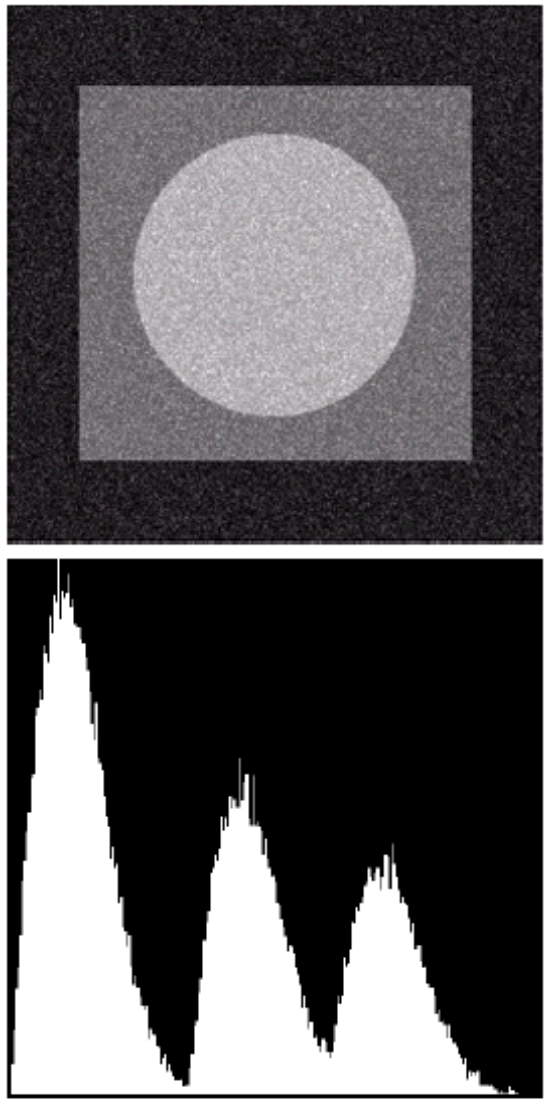
Gaussian Probability Density Function (PDF)

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



Rayleigh Random Noise

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



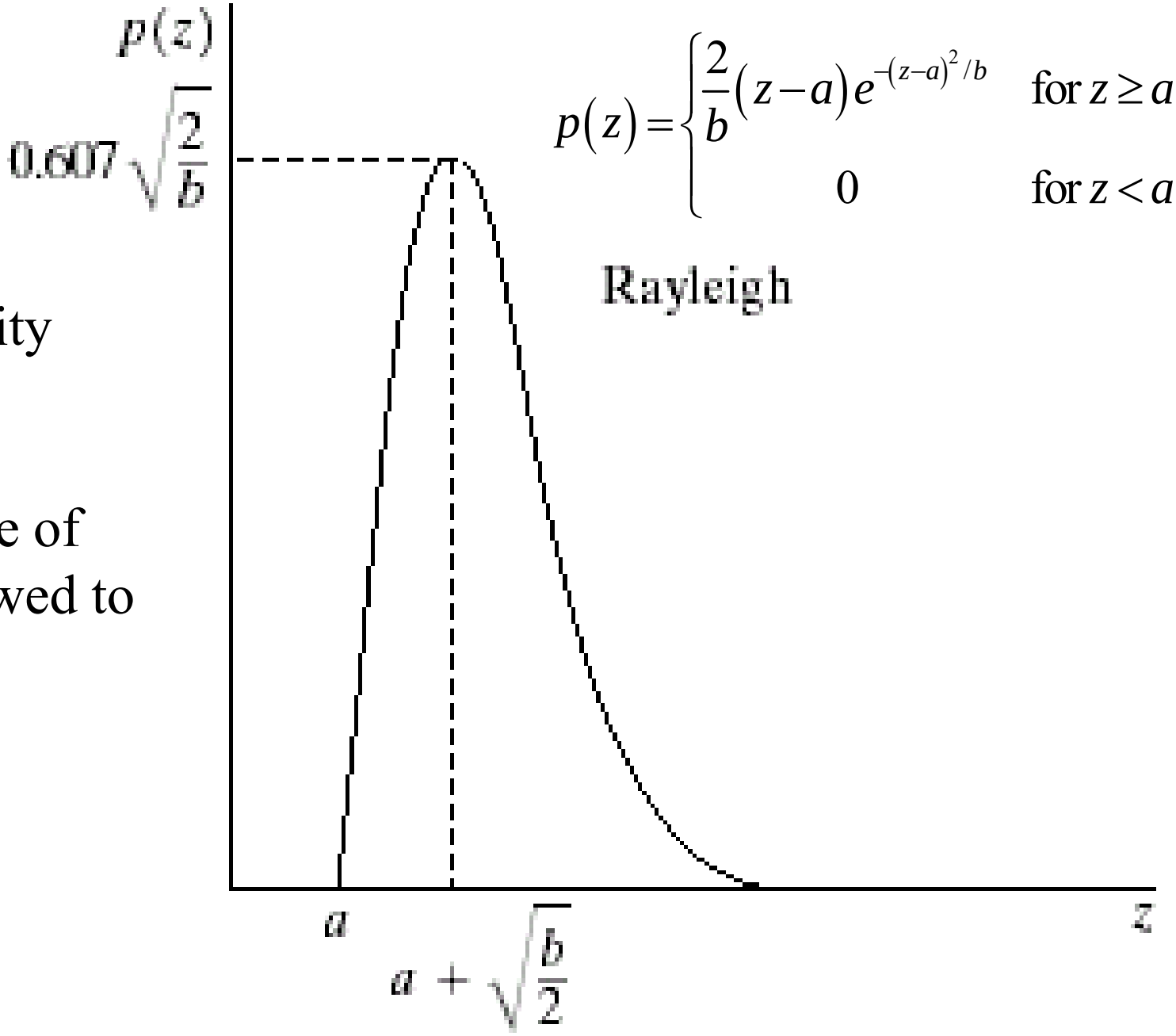
Mean $\mu=a+\sqrt{\frac{\pi b}{4}}$

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

Rayleigh

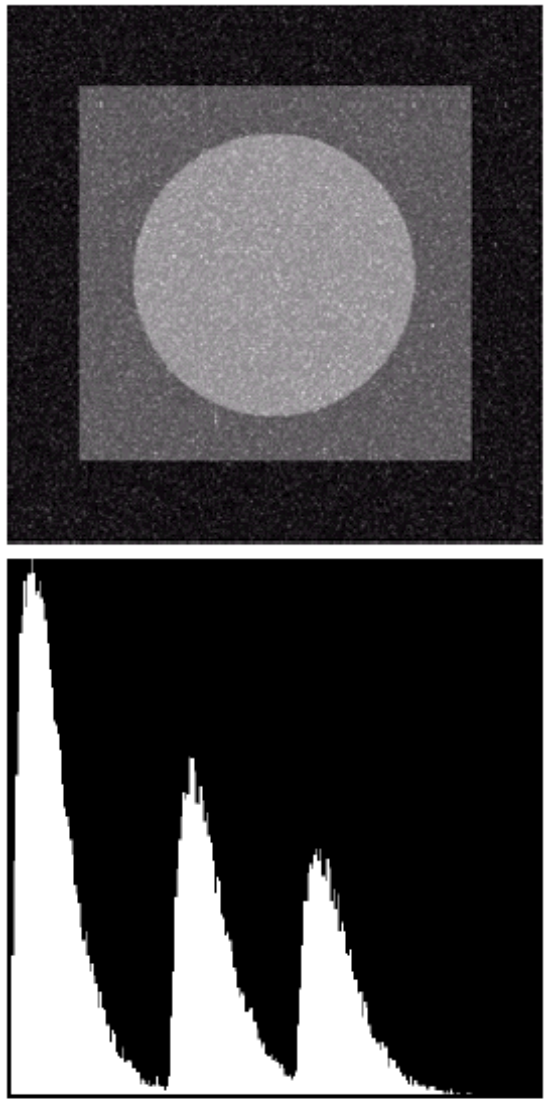
Rayleigh Probability Density Function

- The basic shape of this density is skewed to the right.



Gamma Random Noise

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



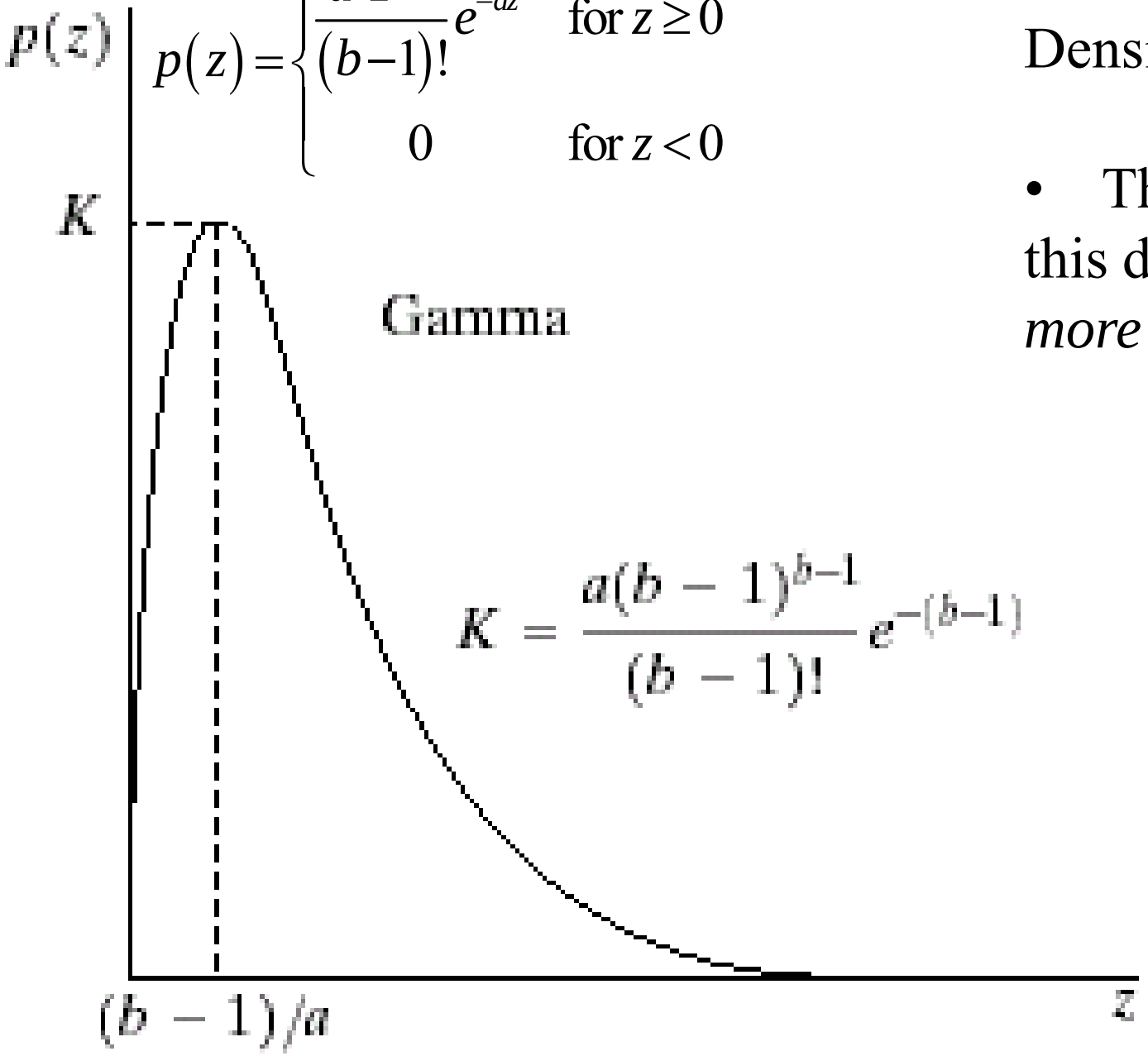
Mean $\mu=\dfrac{b}{a}$

Variance $\sigma^2=\dfrac{b}{a^2}$

Gamma

Gamma Probability Density Function

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



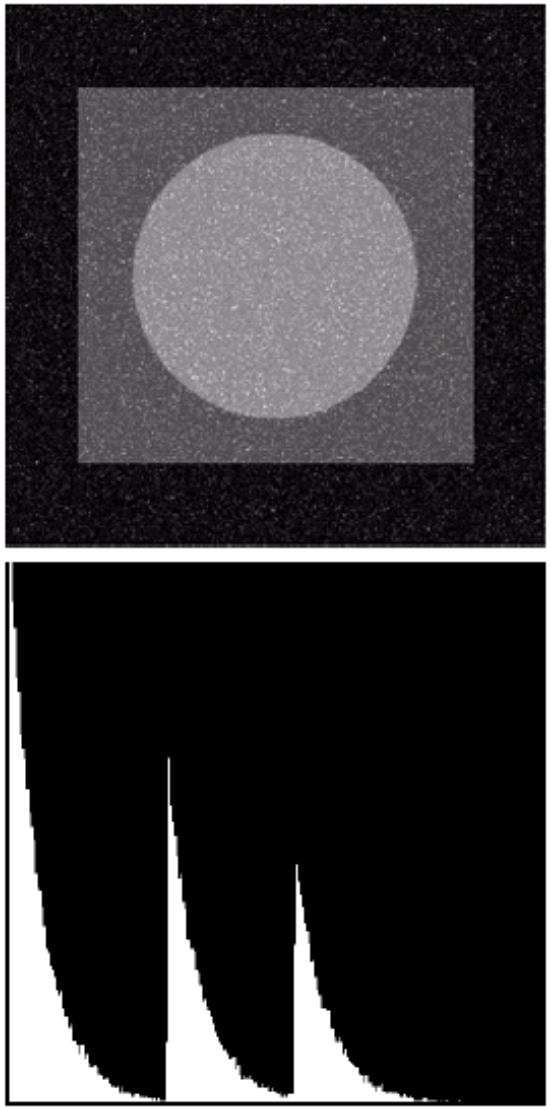
- The basic shape of this density is skewed *more* to the right.

Exponential Random Noise

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

Mean $$\mu = \frac{1}{a}$$

Variance $$\sigma^2 = \frac{1}{a^2}$$

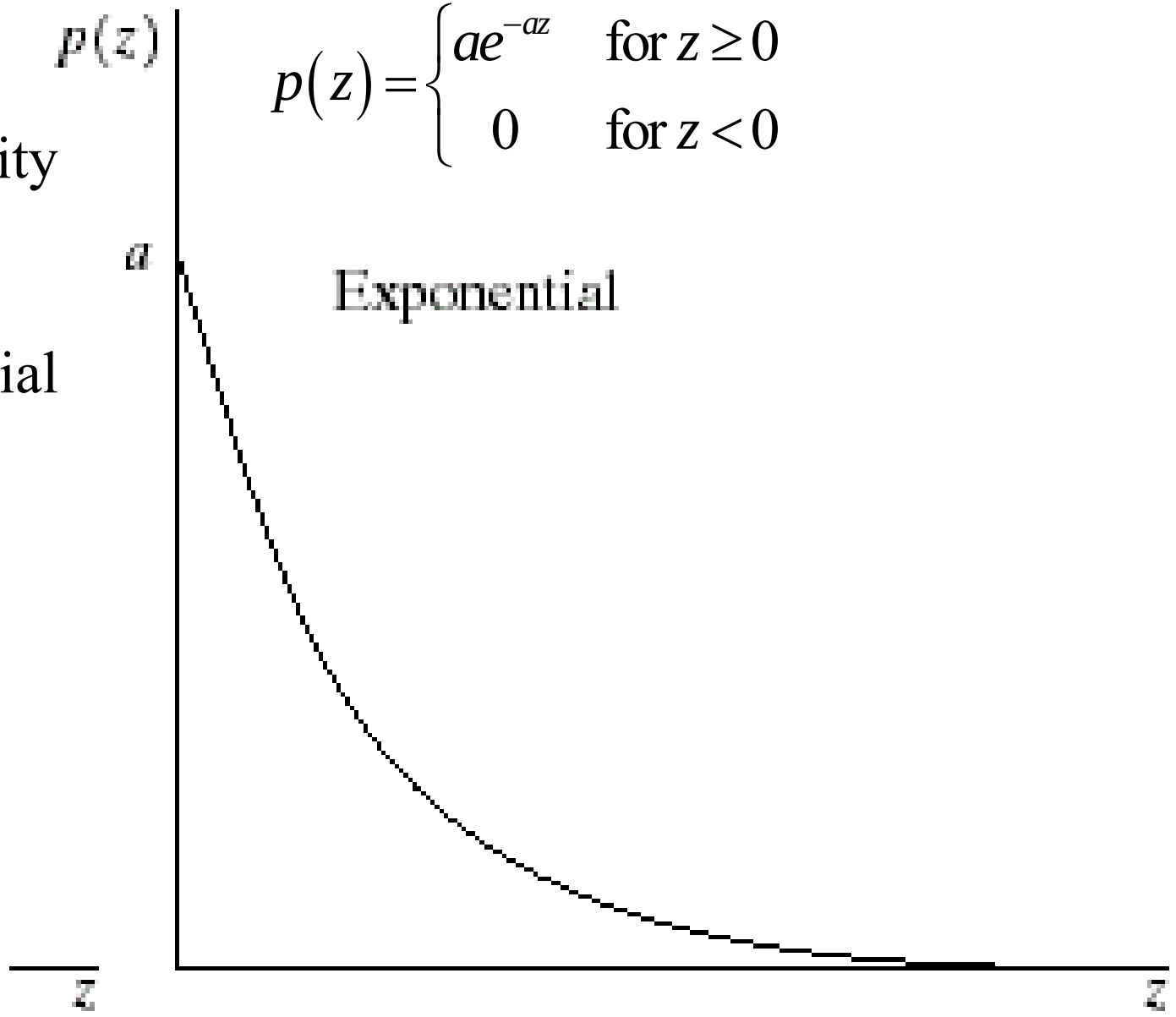


Exponential

Exponential Probability Density Function

- this PDF is a special case when b = 1 in Gamma PDF.

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

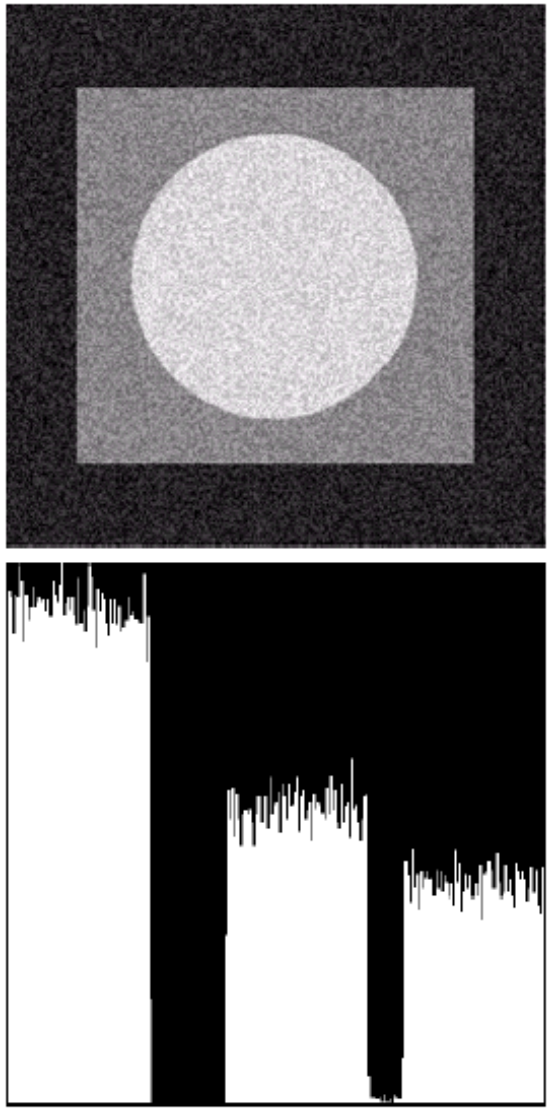


Uniform Random Noise

$$p(z)=\begin{cases}\dfrac{1}{b-a} & \text{if } a\le z\le b\\ 0 & \text{otherwise}\end{cases}$$

Mean $$\mu=\frac{a+b}{2}$$

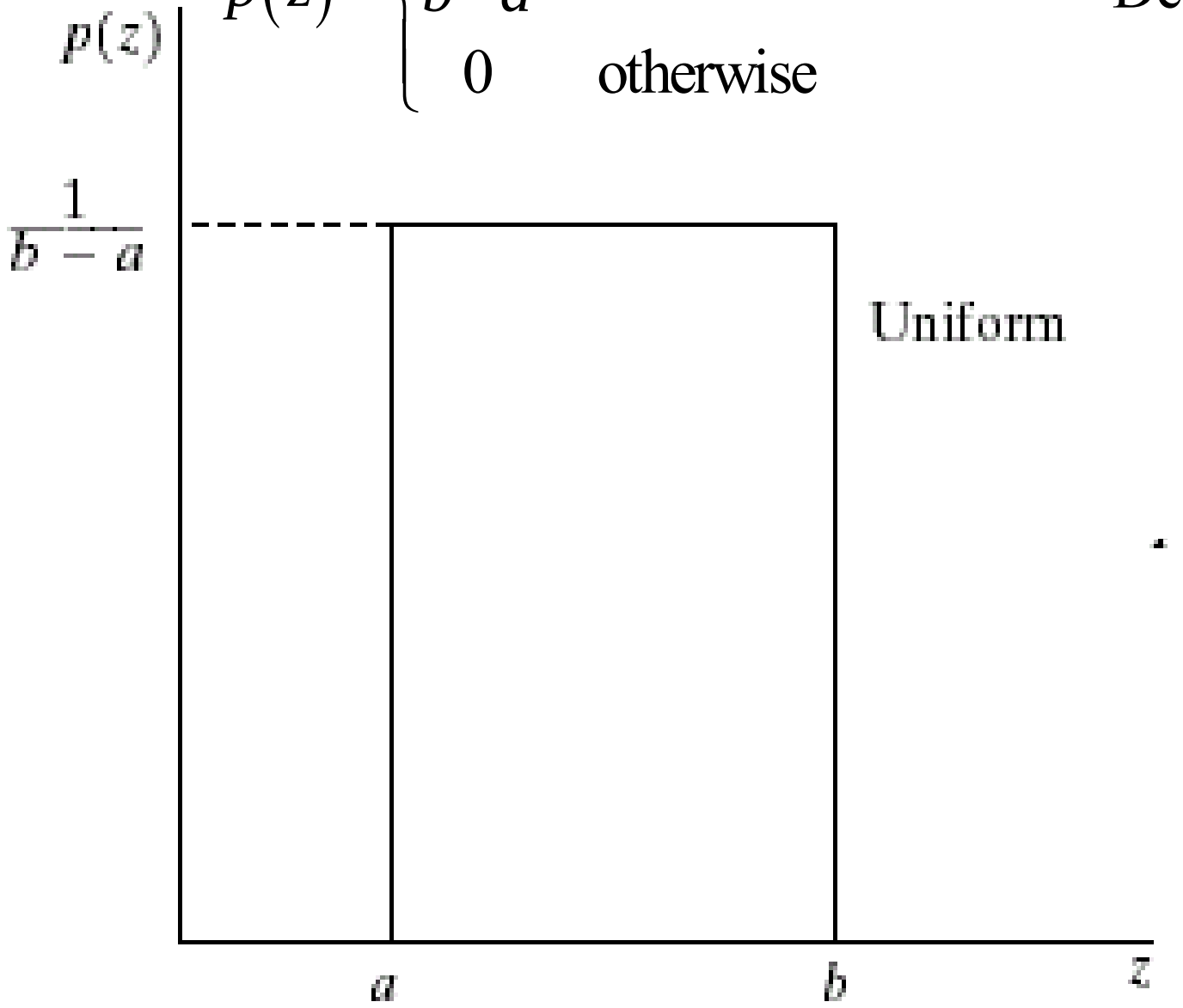
Variance $$\sigma^2=\frac{(b-a)^2}{12}$$



Uniform

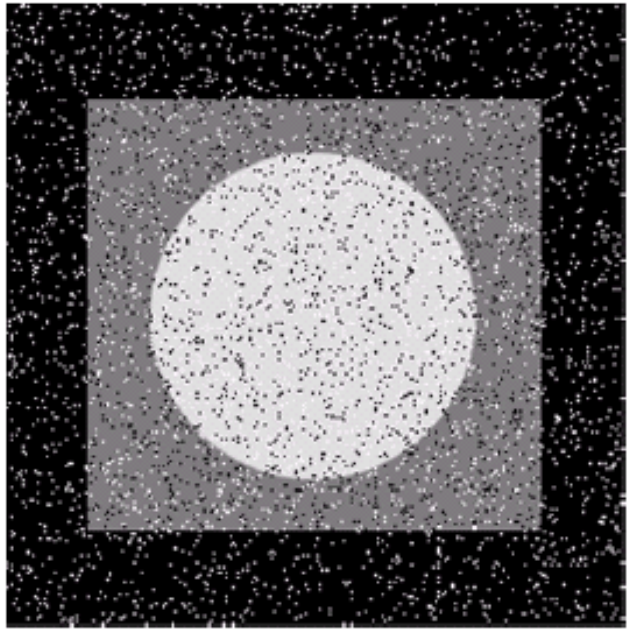
Uniform Probability Density Function

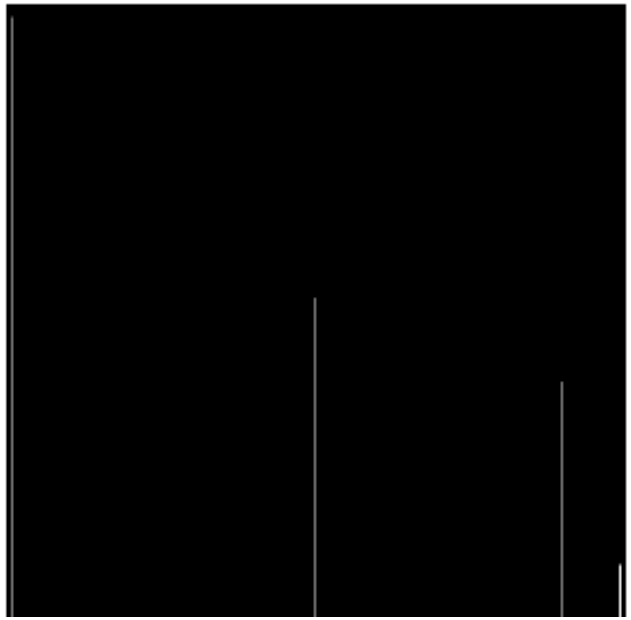
$$p(z) = \begin{cases} \dfrac{1}{b-a} & \text{if } a \le z \le b \\ 0 & \text{otherwise} \end{cases}$$



Impulse (salt-and-pepper) Random Noise

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





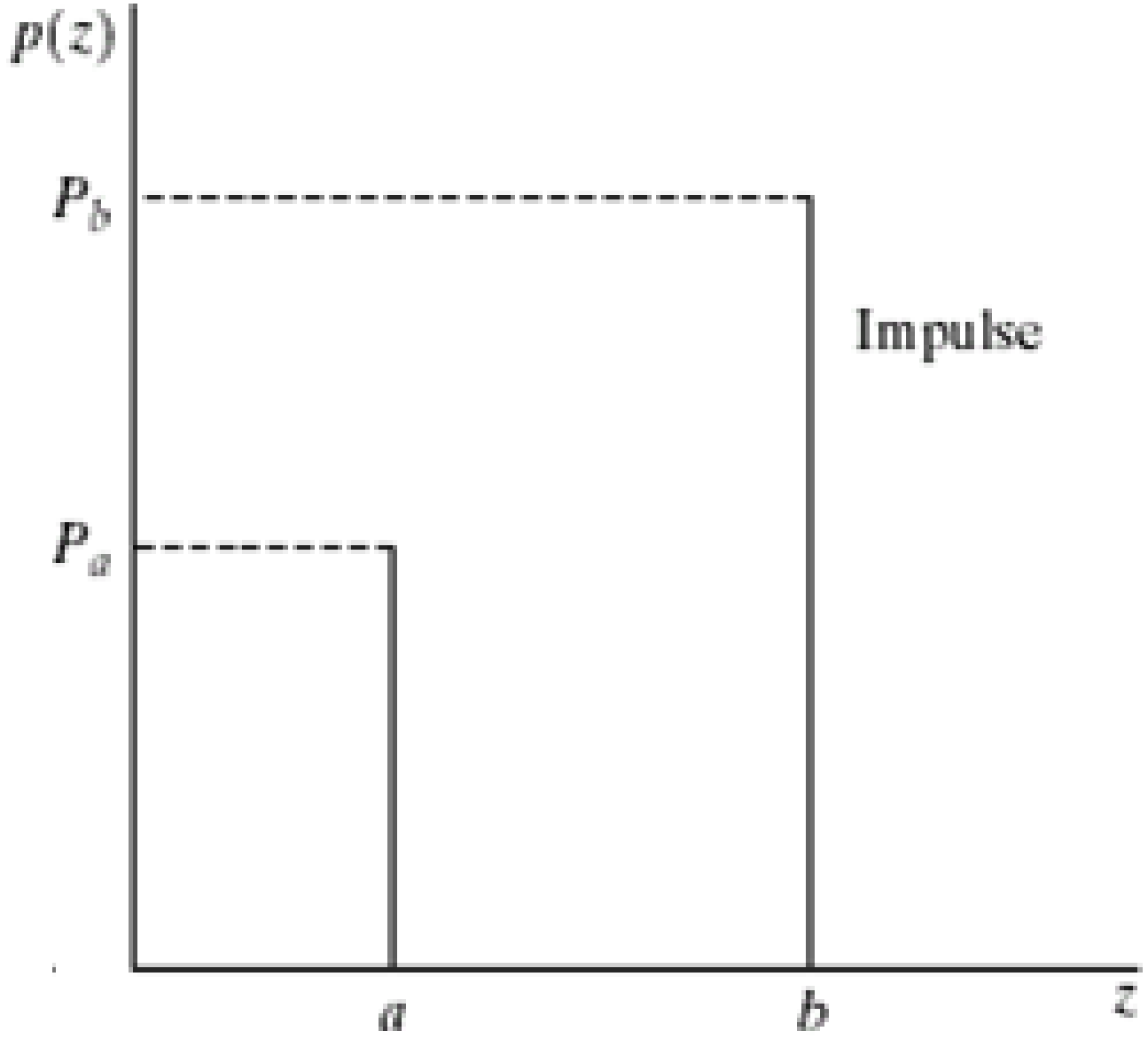
Impulse (salt-and-pepper) Random Noise

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

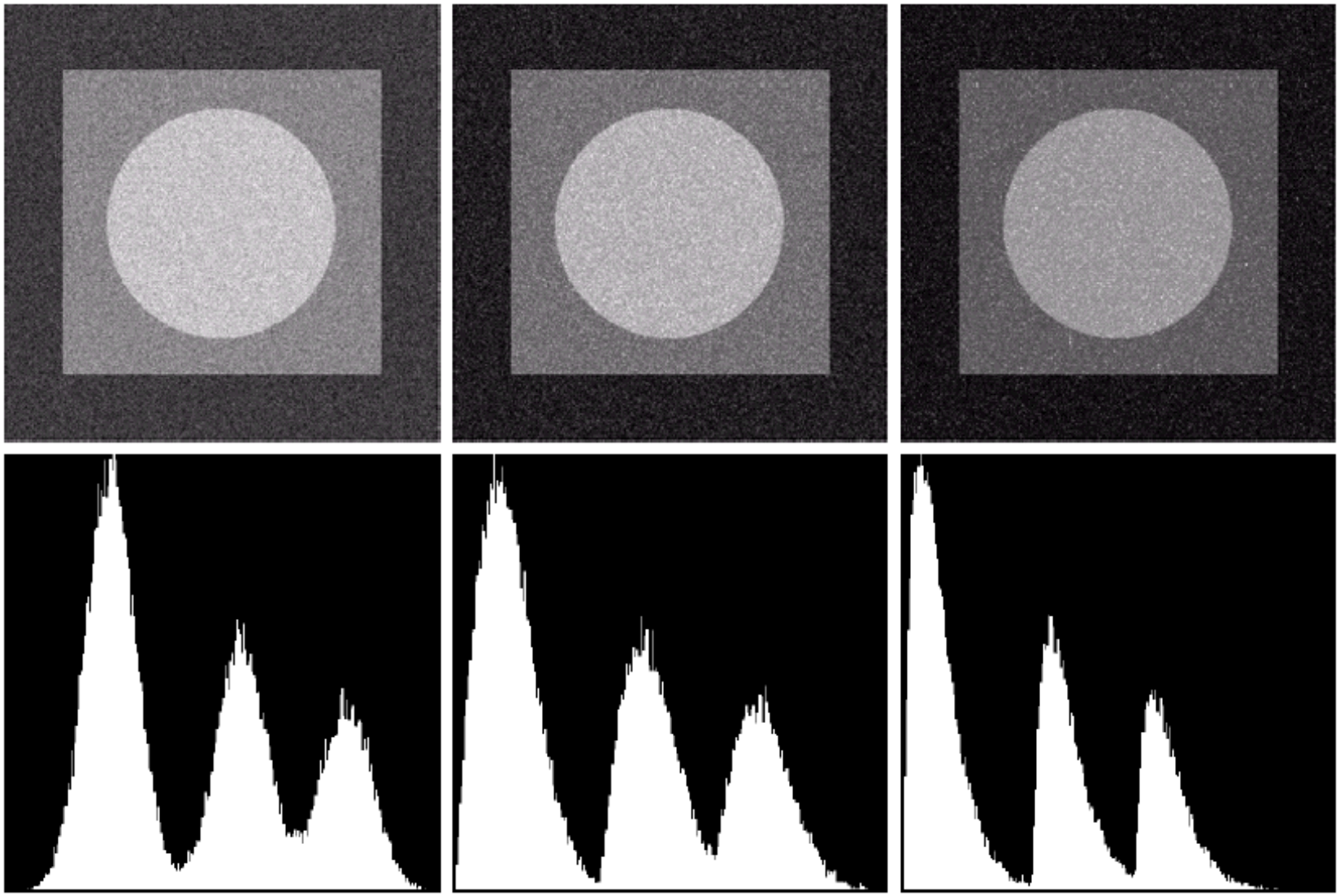
- If b > a, gray-level b will appear as a light dot in the image, and gray-level a will appear like a dark dot.
- The dark dots are called pepper noise.
- The bright dots are called salt noise.
- If either P_a or P_b is zero, the impulse noise is called unipolar.
- If neither probability is zero, and especially if they are approximately equal, impulse noise values will resemble salt-and-pepper granules (small pieces) randomly distributed over the image.

Impulse Probability Density Function

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



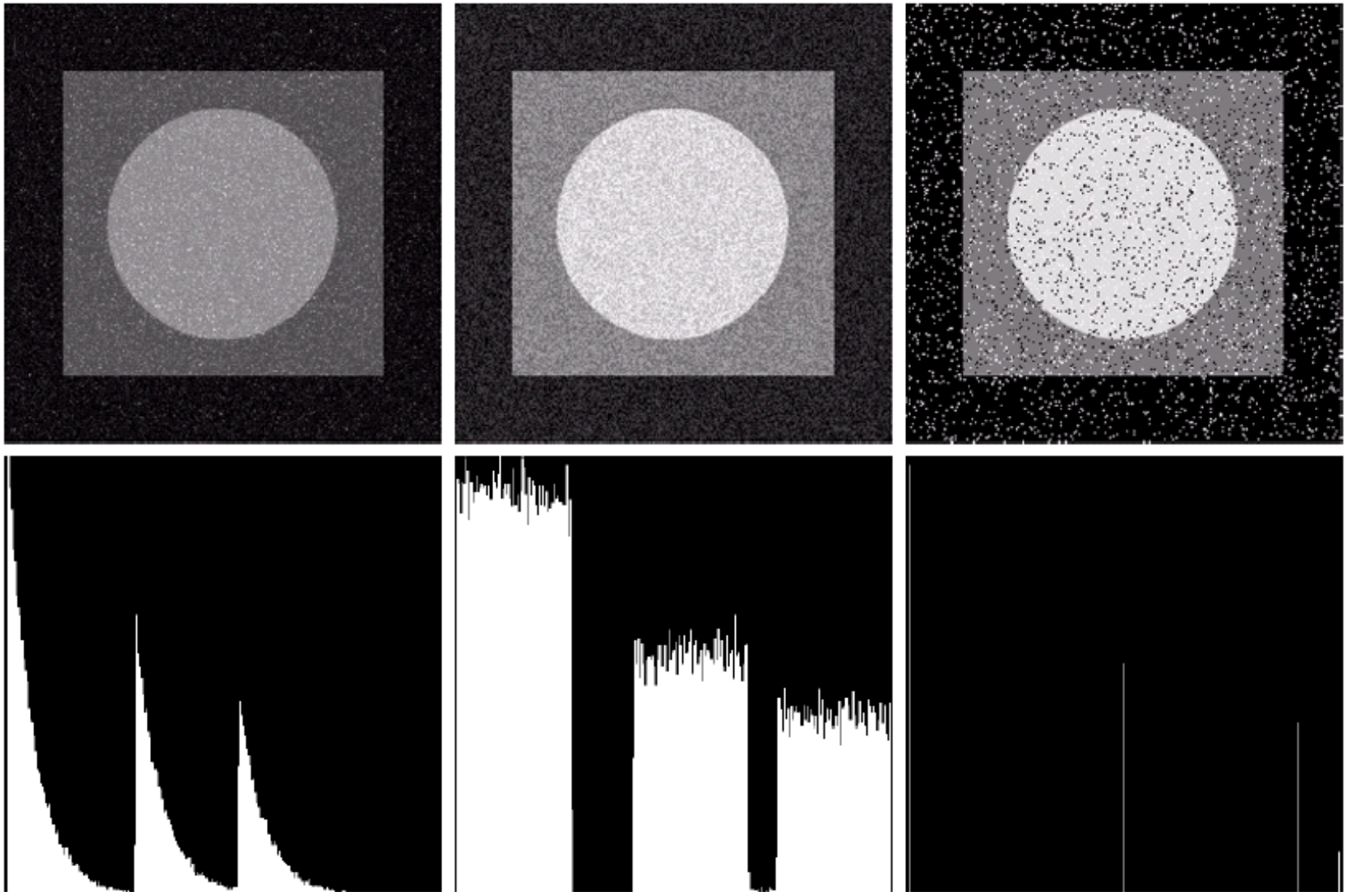
Images and histograms



a b c
d e f

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

Images and histograms

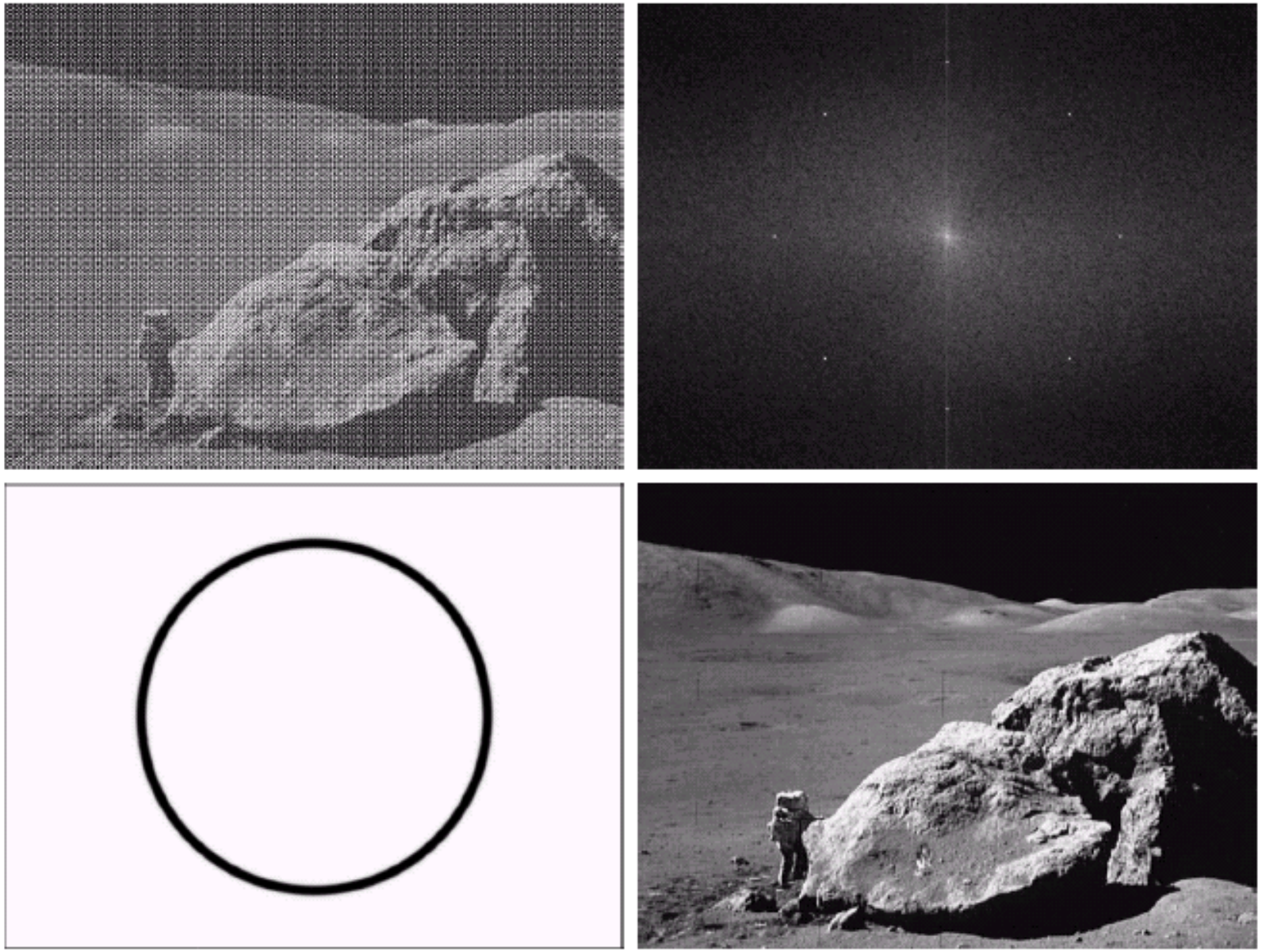


g h i
j k l

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

Periodic Noise

- This is a spatially dependent noise, which can be reduced significantly via frequency domain filtering.



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.)

Estimation of Noise Parameters

- The parameters of periodic noise typically are estimated by inspection of the Fourier representation (spectrum) of the image.
- For example, periodic noise tends to produce frequency spikes that often can be detected even by visual analysis.

- For random noise, it is possible to (a) find the forms of PDF, and (b) estimate the parameters of the PDF from patches of reasonably constant grey level.
- The simplest way to use the data from the image region of interest (ROI) is to calculate the mean and variance of the grey levels.

Mean $$\mu = \sum_{z_i \in S} z_i \, p(z_i)$$

Variance $$\sigma^2 = \sum_{z_i \in S} (z_i - \mu)^2 \, p(z_i)$$

- the z_i's are the grey-level values of the pixels in a ROI (or window) S, and $p(z_i)$ are the corresponding normalised histogram values.

Restoration in the Presence of Noise Only-Spatial Filtering

Mean Filters

1. Arithmetic mean filter
2. Geometric mean filter
3. Harmonic mean filter
4. Contraharmonic mean filter

Arithmetic mean filter

- The arithmetic mean filtering process computes the average value of the corrupted image *g(x,y)* in the area defined by S_{xy}.

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

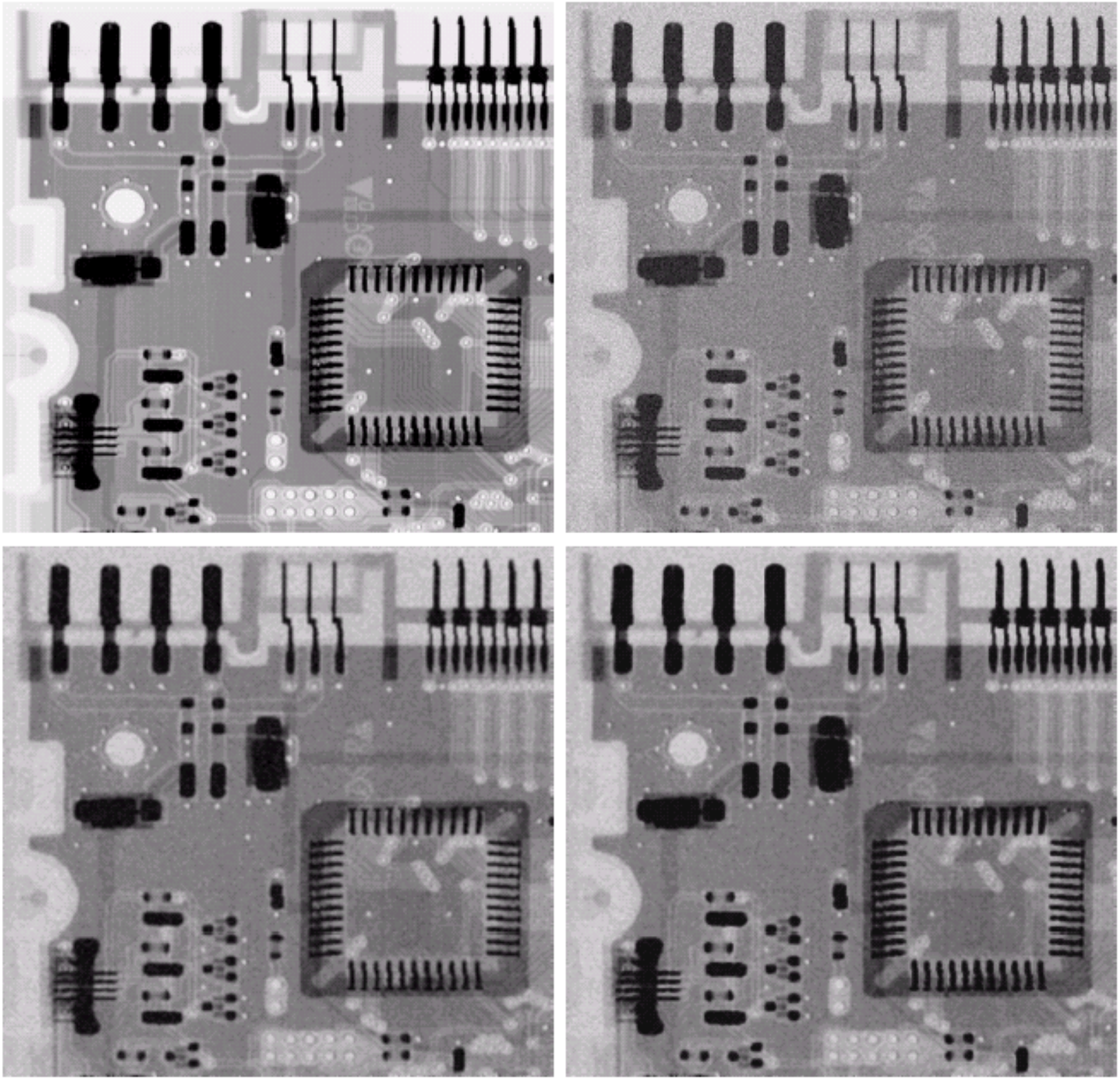
- This can also be implemented by using a mask with all the coefficients equal to 1/(mn).

Geometric mean filter

- The geometric mean filtering process computes the product of the pixels of the corrupted image *g(x,y)* in the area defined by S_{xy}. Then, it raises the product to the power 1/(mn).

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

- A geometric mean filter achieves smoothing comparable to the arithmetic mean filter. It tends to lose less image detail in the process.
- In general, the arithmetic mean filter and geometric mean filter work well for random noise.



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.)

Harmonic mean filter

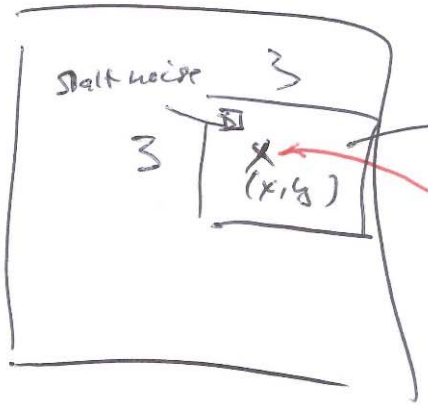
- The harmonic mean filtering process is given by

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

- It works well for salt noise, but fails for pepper noise.
- It does well with other types of noise like Gaussian noise.

Harmonic mean filter

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



9 pixels
$mn = 9$

$$\hat{f}(x,y) = \frac{9}{\frac{1}{g_1} + \frac{1}{g_2} + \ldots + \frac{1}{g_9}}$$

small

g_1	g_2	
		g_9

Intensity values

Salt noise → intensity value is high

→ g_1 is corrupted by salt noise

→ $\frac{1}{g_1}$ is small & large

pepper noise

→ intensity value is low

→ g_1 is low

→ $\frac{1}{g_1}$ is high

Contraharmonic mean filter

- The contraharmonic mean filtering process is given by

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

- Q is called the order of the filter.
- For 0 < Q, it eliminates pepper noise.
- For Q < 0, it eliminates salt noise.
- For Q = 0, it becomes arithmetic mean filter.
- For Q = -1, it becomes harmonic mean filter.

a b
c d

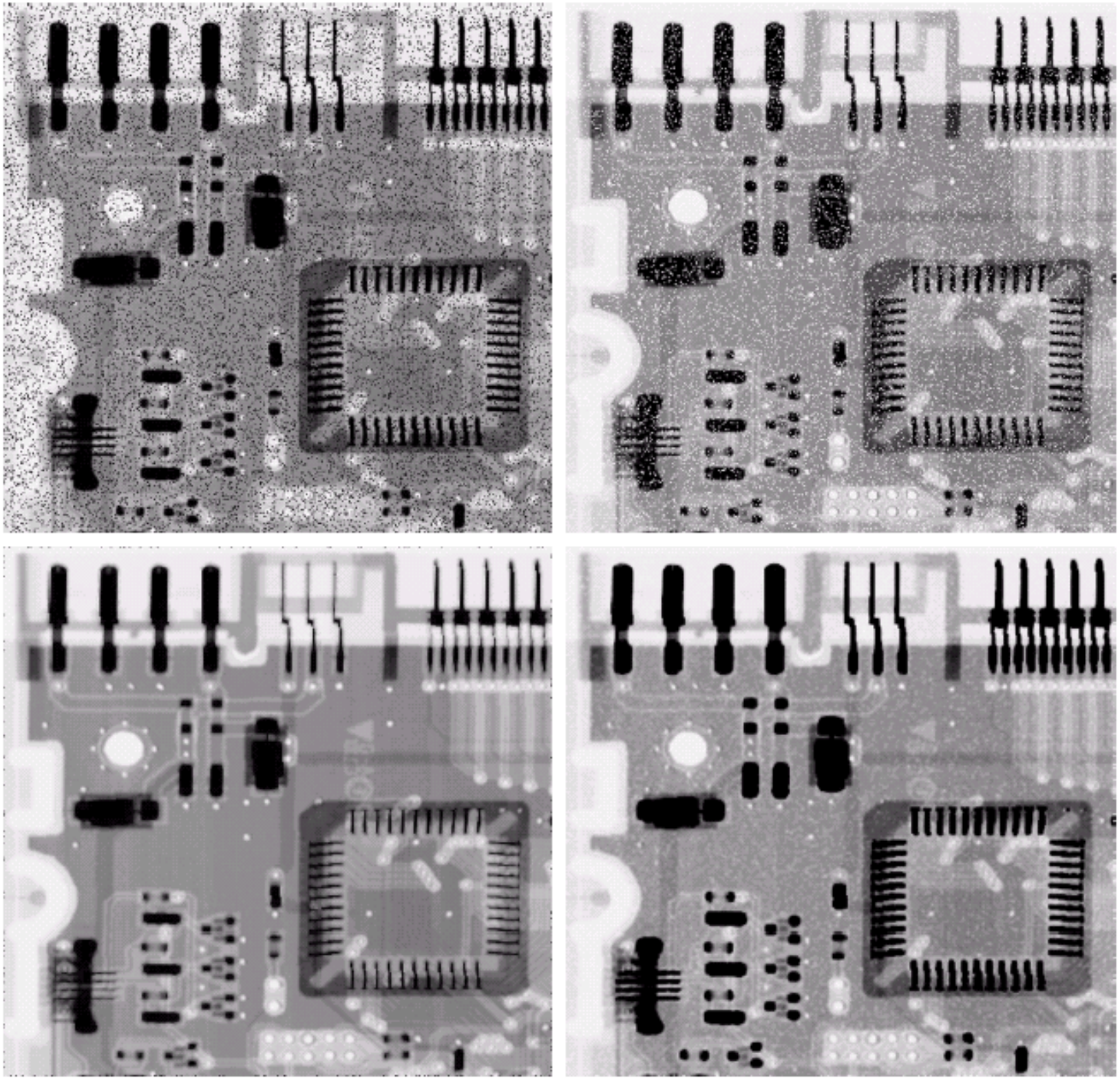
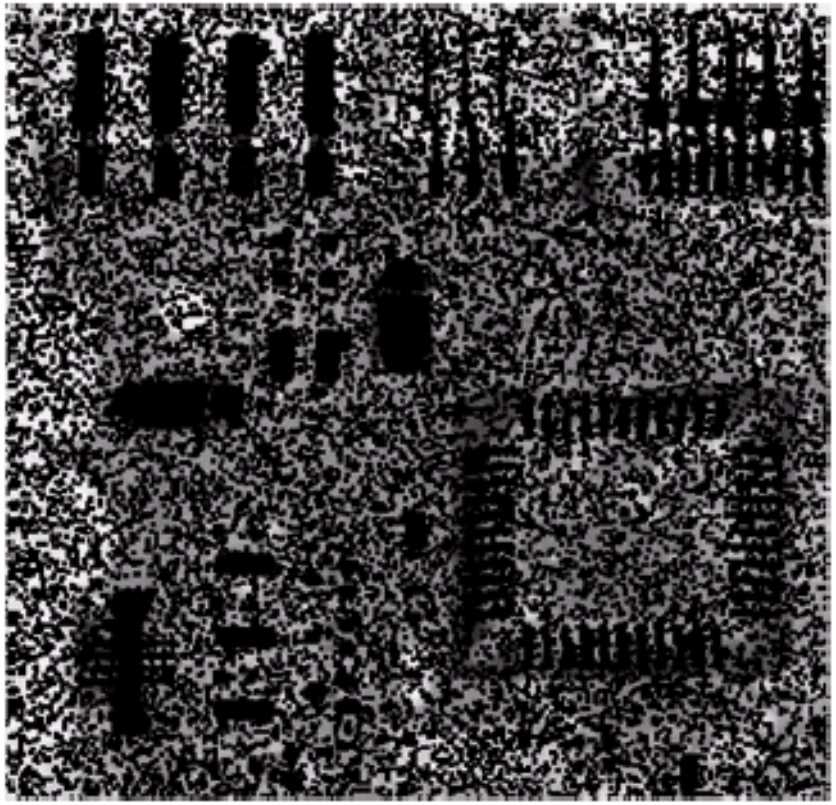
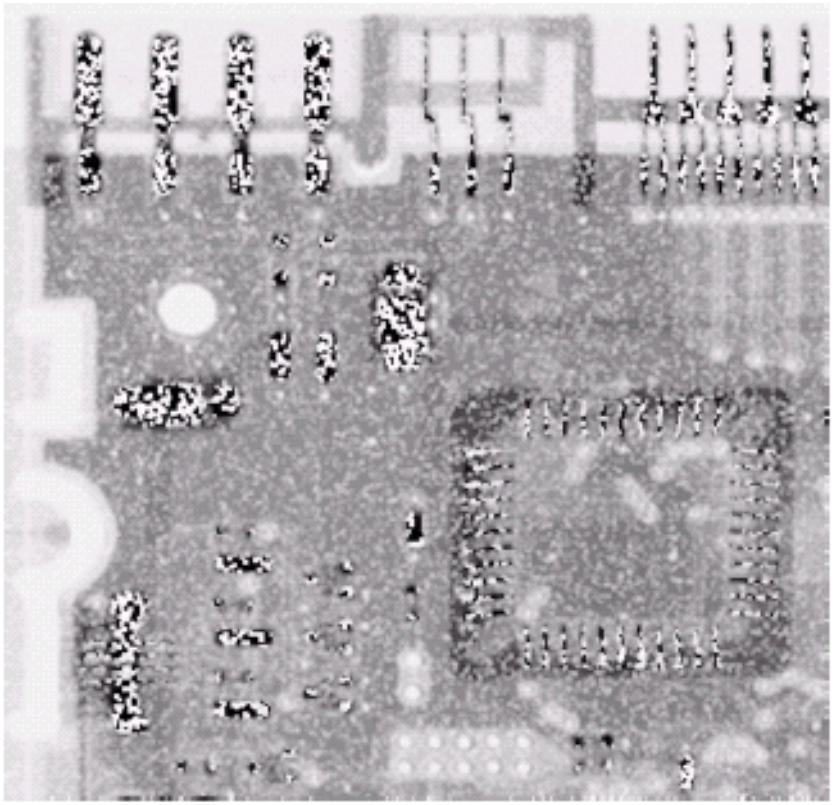


FIGURE 5.8
(a) Image corrupted by pepper noise with a probability of 0.1. (b) Image corrupted by salt noise with the same probability. (c) Result of filtering (a) with a 3×3 contraharmonic filter of order 1.5. (d) Result of filtering (b) with $Q = -1.5$.

Pepper noise

Salt noise





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 $Q = 1.5$.

Q= -1.5

Q=1.5

Order-Statistics Filters

1. Median filter
2. Max and min filters
3. Midpoint filter
4. Alpha-trimmed mean filter

Median filter

- The median filter computes the median of the grey levels in the area S_{xy}.

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

- It provides excellent noise-reduction capabilities, with considerably less blurring than linear smoothing filters of similar size.

- Merge sort/sorting methods is needed for sorting intensity values. https://en.wikipedia.org/wiki/Merge_sort

a b
c d

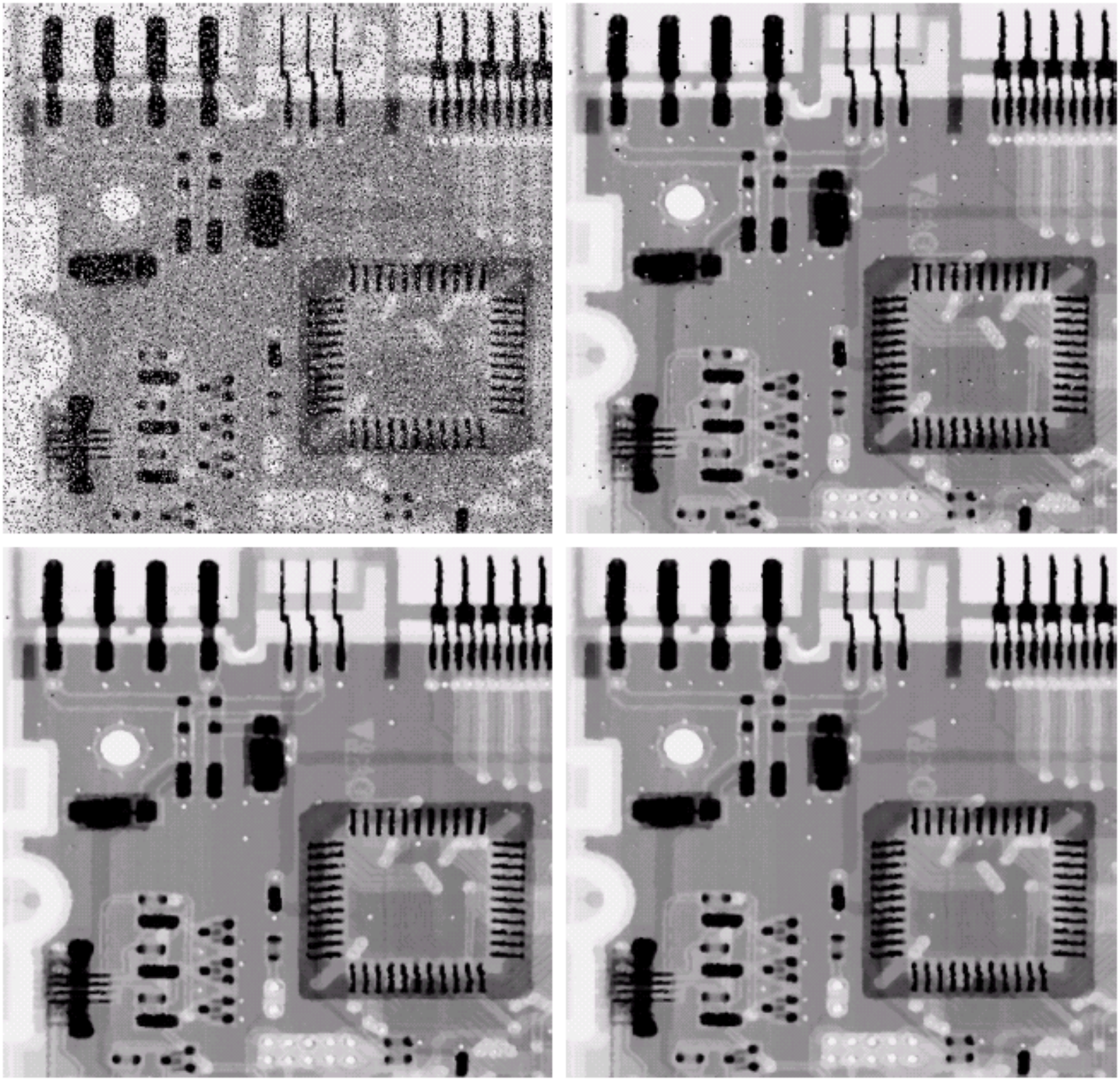


FIGURE 5.10
(a) Image corrupted by salt-and-pepper noise with probabilities $P_a = P_b = 0.1$. (b) Result of one pass with a median filter of size 3×3. (c) Result of processing (b) with this filter. (d) Result of processing (c) with the same filter.

Max and min filters

- The max filter is given by

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

- It finds the brightest point in the area *Sxy*.

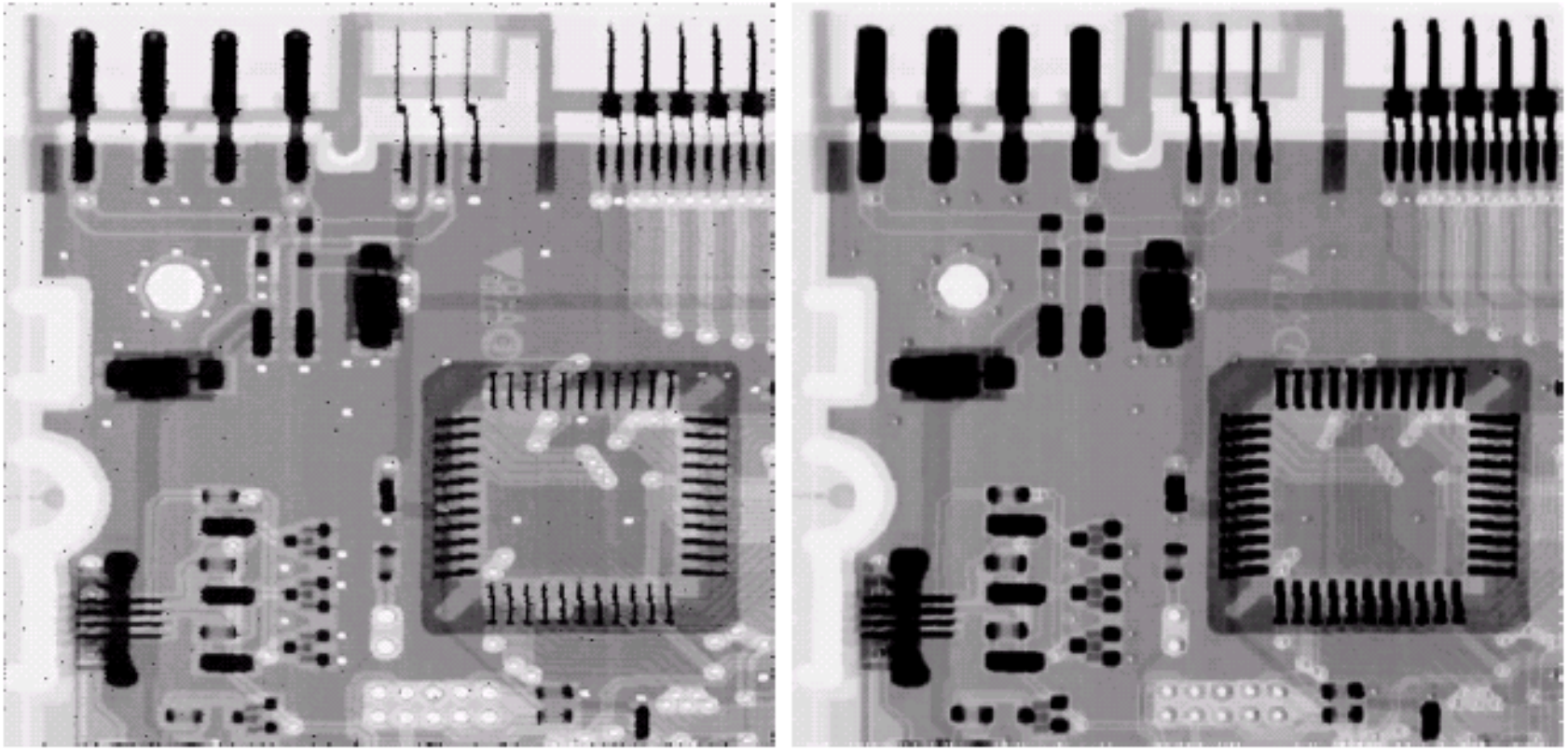
- The min filter is given by

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

- It finds the darkest point in the area *Sxy*.

Max filter result

Min filter result



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.

Midpoint filter

- The midpoint filter computes the midpoint between the max. and min. of the grey level values.

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

- It works best for randomly distributed noise, like Gaussian or uniform noise.

Alpha-trimmed mean filter

- We delete the d/2 lowest and d/2 highest grey-level values of *g(s,t)* in the area S_{xy}.

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

- It works best when multiple types of noise (like salt-and-pepper and Gaussian) are combined in an image.
- If d = (mn – 1), then it becomes a median filter.

Alpha-trimmed mean filter

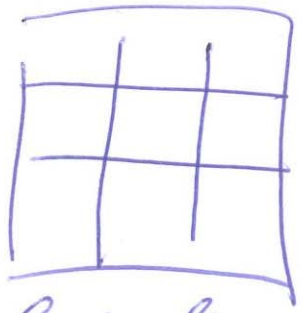
$d = mn - 1$

$= 9 - 1$

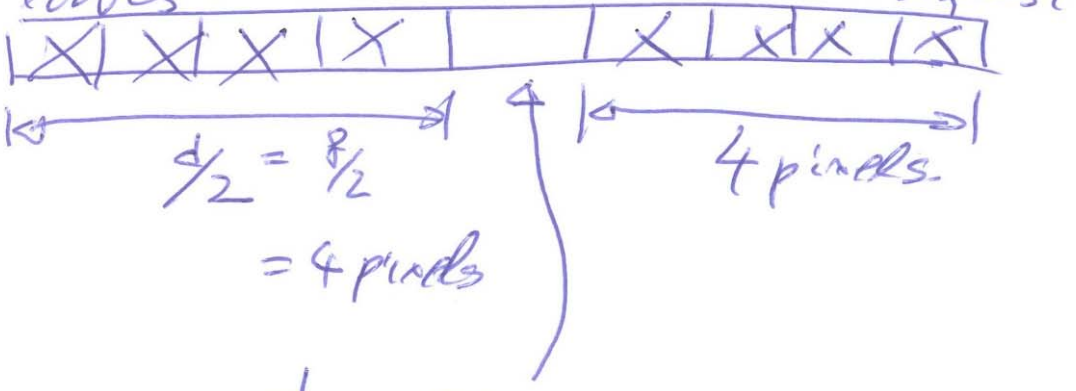
$d = 8$

$m = 3$

$n = 3$



9 pixels.

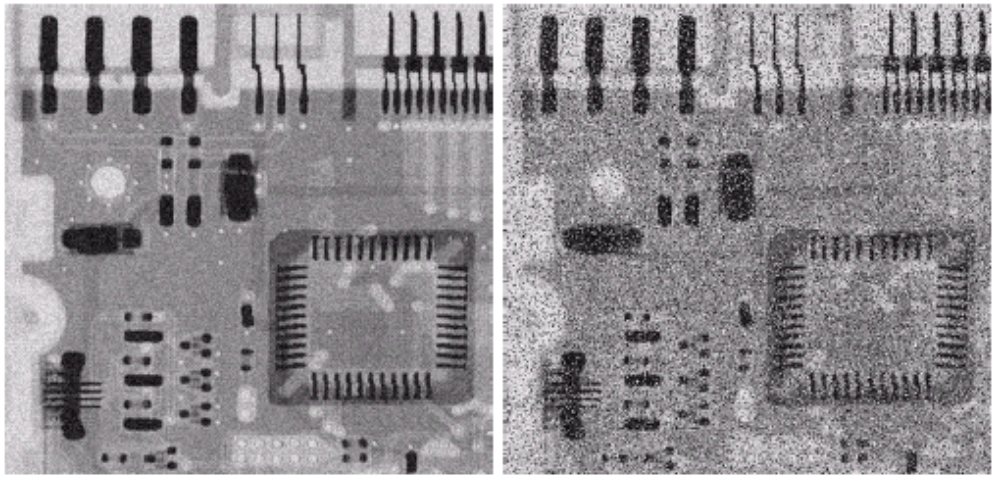


$$\frac{1}{mn-d} \sum g_r$$

$$= \frac{1}{9-8} \sum g_r$$

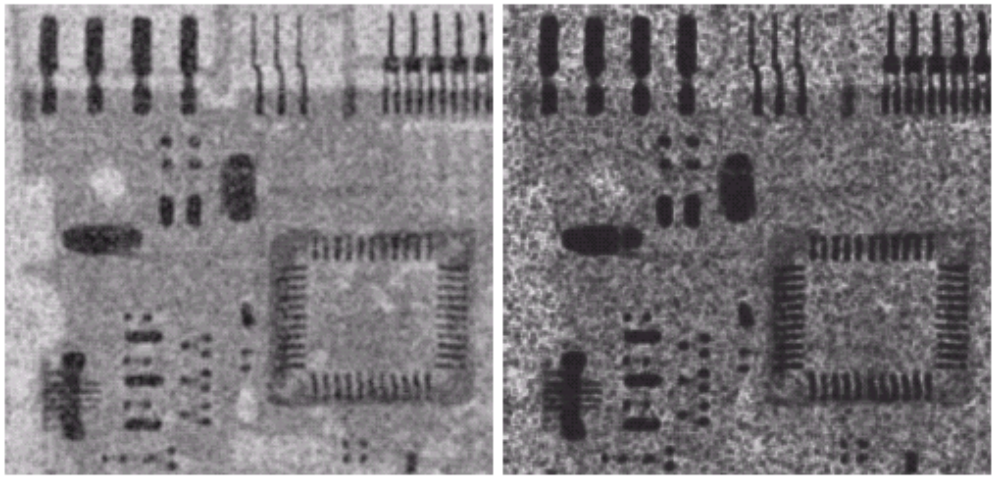
$$= \sum g_r$$

Noisy image (Uniform noise)



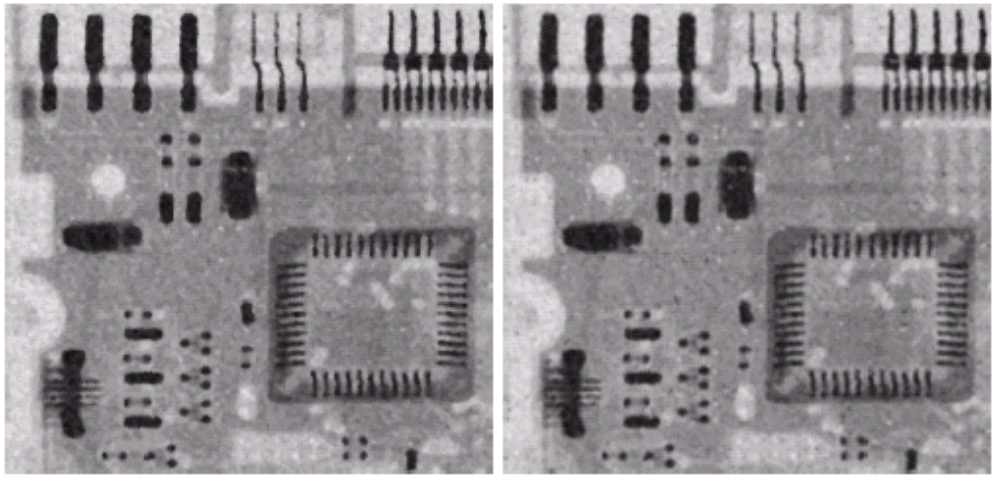
Noisy image (Uniform noise + salt-and-pepper noise)

Arithmetic mean filter



Geometric mean filter

Median filter



Alpha-trimmed mean filter

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

1. Adaptive, local noise reduction filter
2. Adaptive median filter

Adaptive, local noise reduction filter

- Mean and variance are used.

S_{xy} = neighbouring pixels

S'_{xy} = pixels in the pure noise region

$$m_L = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} g(s,t)$$

$$\sigma_L^2 = \frac{1}{mn} \sum_{(s,t) \in S_{xy}} (g(s,t) - m_L)^2$$

$$m_\eta = \frac{1}{m'n'} \sum_{(s,t) \in S'_{xy}} g(s,t)$$

η = eta

$$\sigma_\eta^2 = \frac{1}{m'n'} \sum_{(s,t) \in S'_{xy}} (g(s,t) - m_\eta)^2$$

Adaptive, local noise reduction filter

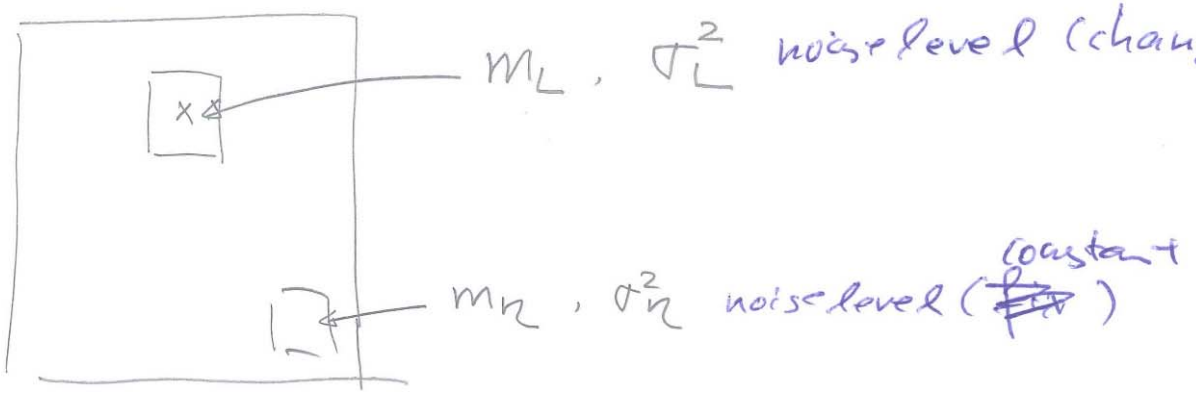
- The filter is given by

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

- If $\sigma_\eta^2 = 0$, then the filter gives the observed, corrupted image.
- If $\sigma_\eta^2 = \sigma_L^2$, then the filter gives the local arithmetic mean.

- If $\sigma_\eta^2 << \sigma_L^2$, then the filter gives a value close to *g(x,y)* in order to preserve edges or boundaries.

Adaptive, local noise reduction filter



If $\sigma_\eta^2 = 0 \rightarrow \hat{f}(x,y) = g(x,y)$

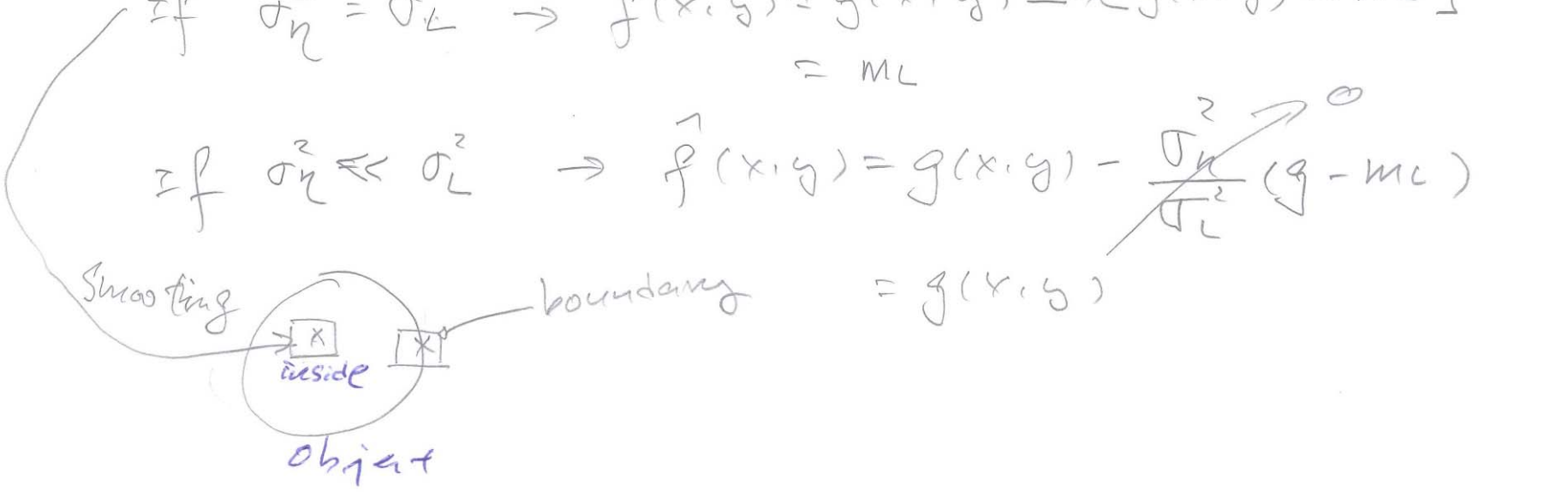
If $\sigma_\eta^2 = \sigma_L^2 \rightarrow \hat{f}(x,y) = g(x,y) - (1)[g(x,y) - m_L]$

$= m_L$

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

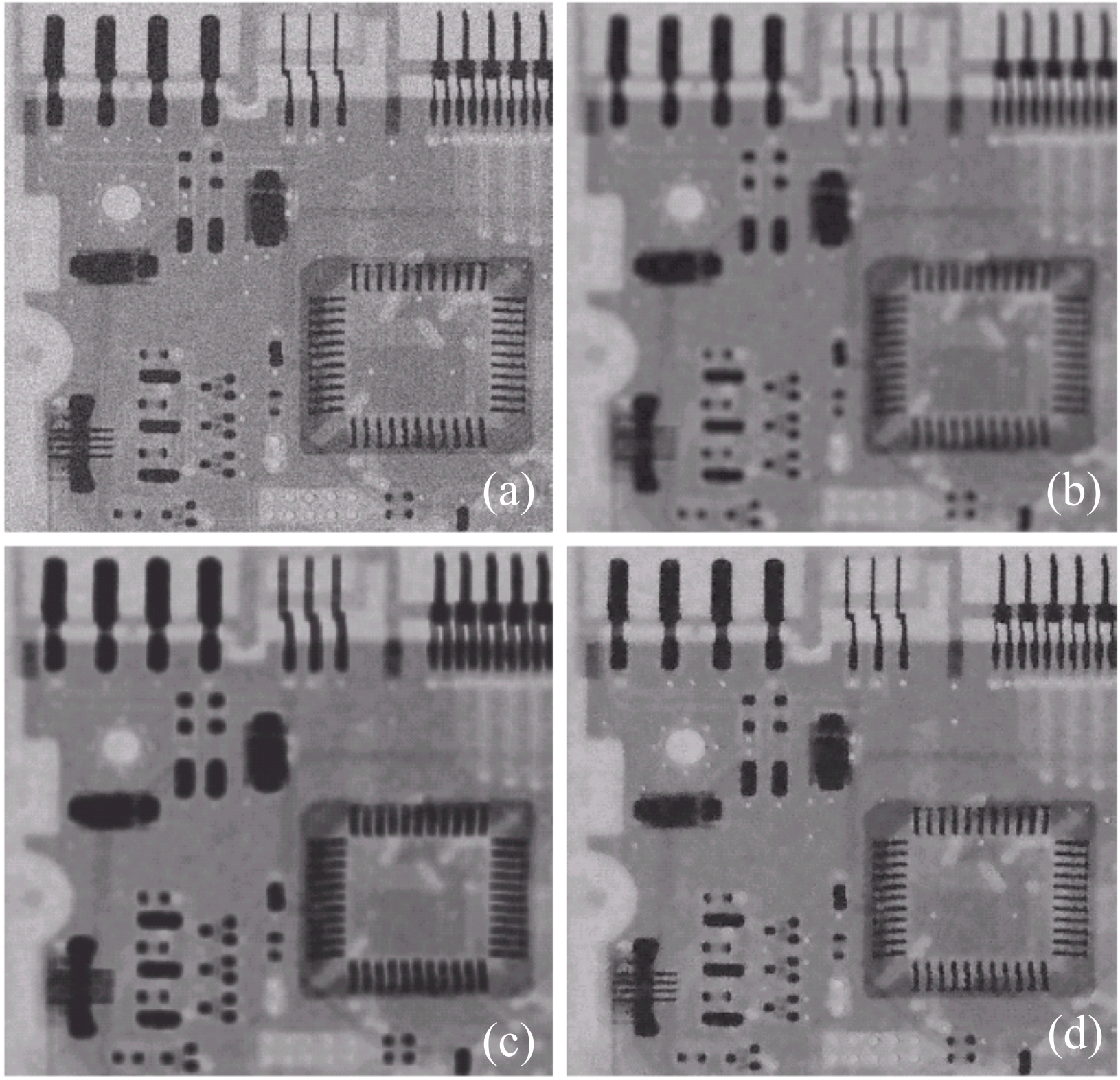
$= g(x,y)$

$\sigma_\eta^2 / \sigma_L^2$ (local average)



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.



Adaptive median filter

- Z_{min} and Z_{max} = min. and max. grey level values in S_{xy} respectively
- Z_{med} = median of grey level values in S_{xy}
- z_{xy} = grey level at coordinates (x,y)
- S_{max} = max. allowed size of S_{xy}

Level A: $A1 = Z_{med} - Z_{min}$; $A2 = Z_{med} - Z_{max}$

If $A1 > 0$ and $A2 < 0$, Go to 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}.

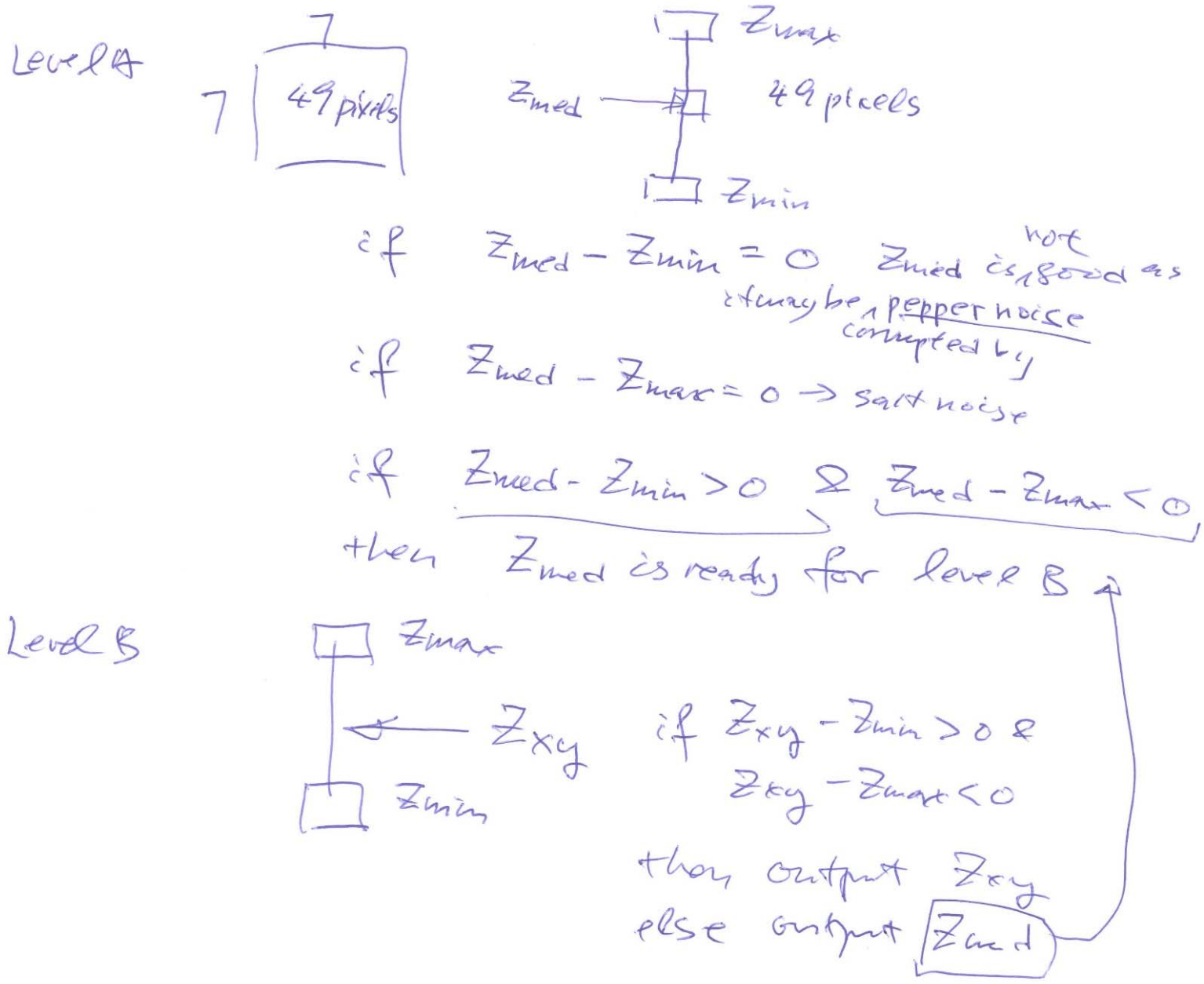
Else output Z_{med}.

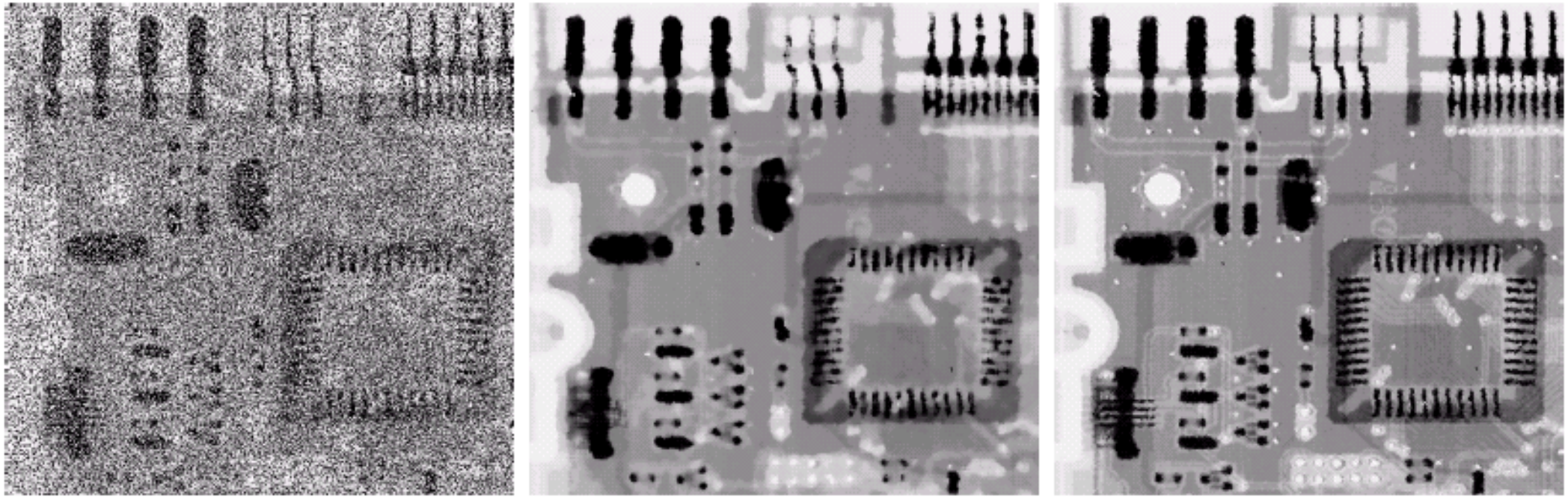
Adaptive median filter

- Level A determines whether the median filter output Z_{med} is an impulse (black or white) or not.
- If the condition $Z_{min} < Z_{med} < Z_{max}$ holds (i.e. $A1 > 0$ and $A2 < 0$), then Z_{med} cannot be an impulse.
- Otherwise, increase the window size.

- Level B determines whether the image intensity value Z_{xy} is an impulse (black or white) or not.
- If Z_{xy} is not an impulse, then output the original value (no change).
- Otherwise, the filter gives the local median filtered output.

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 $S_{max} = 7$.

Adaptive median filter example

Question 3

Adaptive median filter (8 points)

Assumptions:

1. Z_{xy} represents gray level at coordinates (x,y).
2. Z_{med} represents median of gray levels in a window S_{xy}
   (e.g., if $S_{xy} = 3$, then it is a 3 x 3 window with center at Z_{xy}).
3. Z_{min} and Z_{max} are minimum and maximum gray levels in a window S_{xy} respectively.
4. S_{max} denotes the maximum allowed window size of S_{xy}.

The adaptive median filter is defined as follows.

$S_{max} = 9$

For each pixel Z_{xy} in an image,

$S_{xy} = 3$

Level A:

$A1 = Z_{med} - Z_{min}$; $A2 = Z_{med} - Z_{max}$

If $A1 > 0$ and $A2 < 0$, then go to Level B,

else increase the window size, i.e., $S_{xy} = S_{xy} + 2$.

If window size $\leq S_{max}$, then repeat level A,

else output Z_{xy}.

Level B:

$B1 = Z_{xy} - Z_{min}$; $B2 = Z_{xy} - Z_{max}$

If $B1 > 0$ and $B2 < 0$, then out Z_{xy},

else output Z_{med}.

Adaptive median filter example

21	0	27	0	19	35	28	31	21	0
26	39	21	36	34	21	21	0	12	14
0	12	26	34	25	54	0	30	0	27
16	43	23	0	24	36	32	25	18	33
41	40	0	33	27	0	41	27	30*	39
0	18	39	0**	0	22	28	13	24	17
49	41	0	31	0	32	12	0	32	0
25	29	21	38	35	19	26	35	25	36
21	24	42	0	27	21	41	27	0	14
0	34	0	28	16	24	0	35	22	0
24	20	17	0	26	44	31	27	21	35

Find the filtered values of pixels marked with asterisks.

Filtered value for * = 30 (4)

Filtered value for ** = 24 (4)

~~ End of Written Assignment 3 ~~

z_{med}

13 17 18 24 25 27 30 33 39

z_{min} z_{max}

$A1 = 25 - 13 = 12$

$A2 = 25 - 39 = -14$

$A1 > 0$ & $A2 < 0$

$B1 = 30 - 13 = 17$

$B2 = 30 - 39 = -9$

6

$B1 = 17$

$B2 = -9$

$\Rightarrow z_{xy}$ is the output

Periodic Noise Reduction by Frequency Domain Filtering

Filters for periodic noise reduction:

1. Bandreject Filters
2. Bandpass Filters
3. Notch Filters
4. Optimum Notch Filtering

Bandreject filters

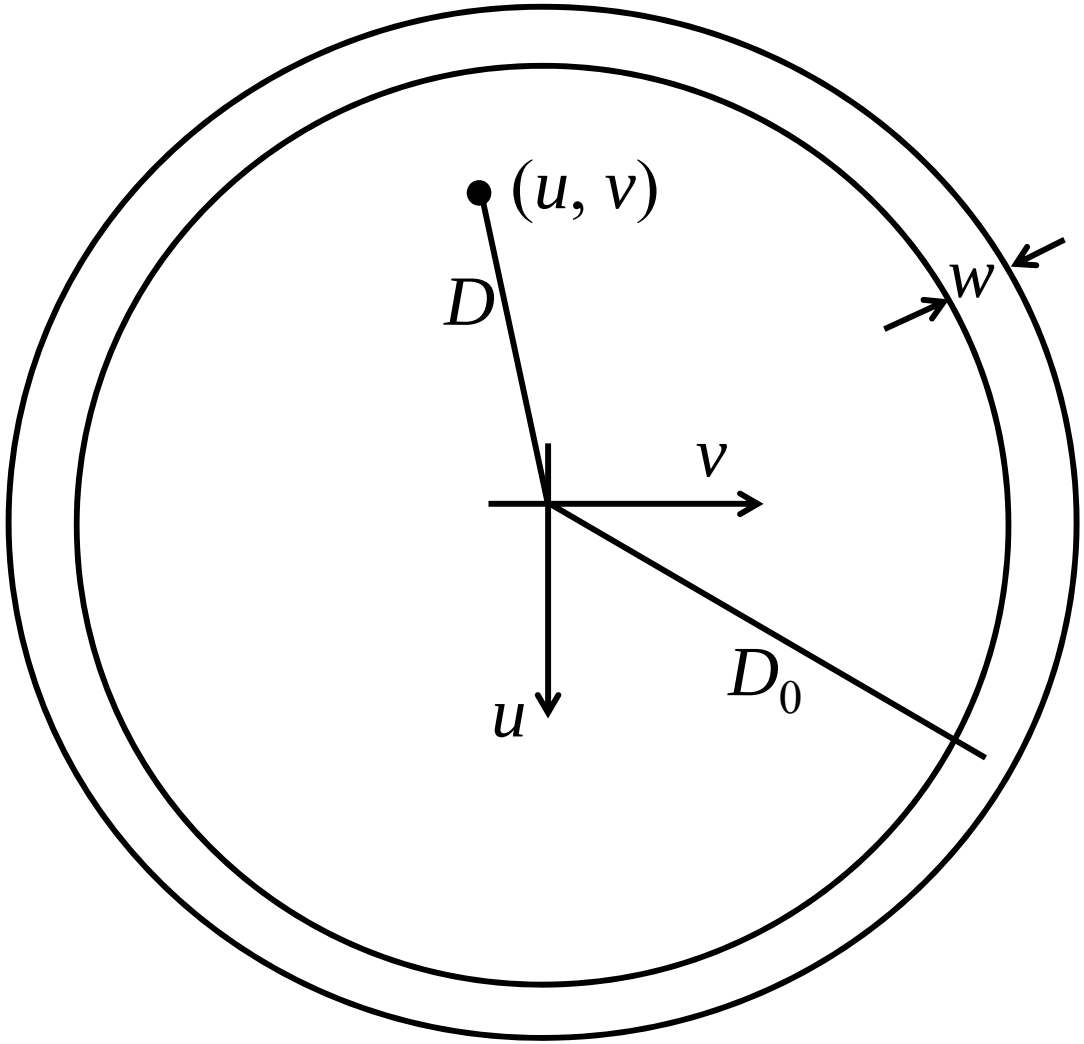
- Bandreject filters remove a band of frequencies about the origin of the frequency representation (or Fourier spectrum).
- Ideal Bandreject Filter

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

W = width of the band
D_0 = radial centre

Bandreject filters

H(u, v)



Bandreject filters

- Butterworth Bandreject Filter

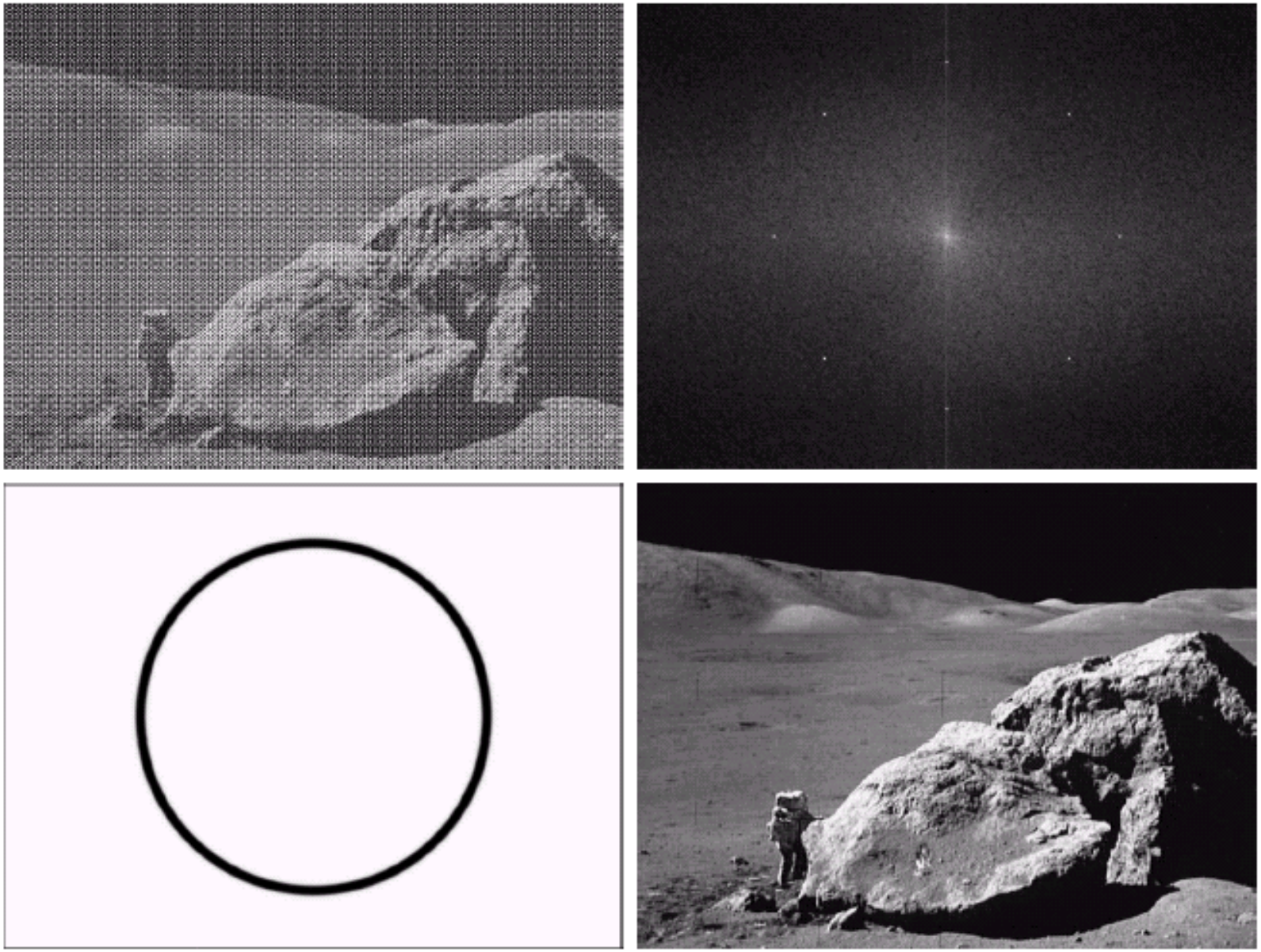
$$H(u,v) = \frac{1}{1 + \left[\frac{D(u,v)W}{D^2(u,v) - D_0^2} \right]^{2n}}$$

- Gaussian Bandreject Filter

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

Butterworth Bandreject Filter

- It is a sharp, narrow filter in frequency domain.
- order n = 4



a b
c d

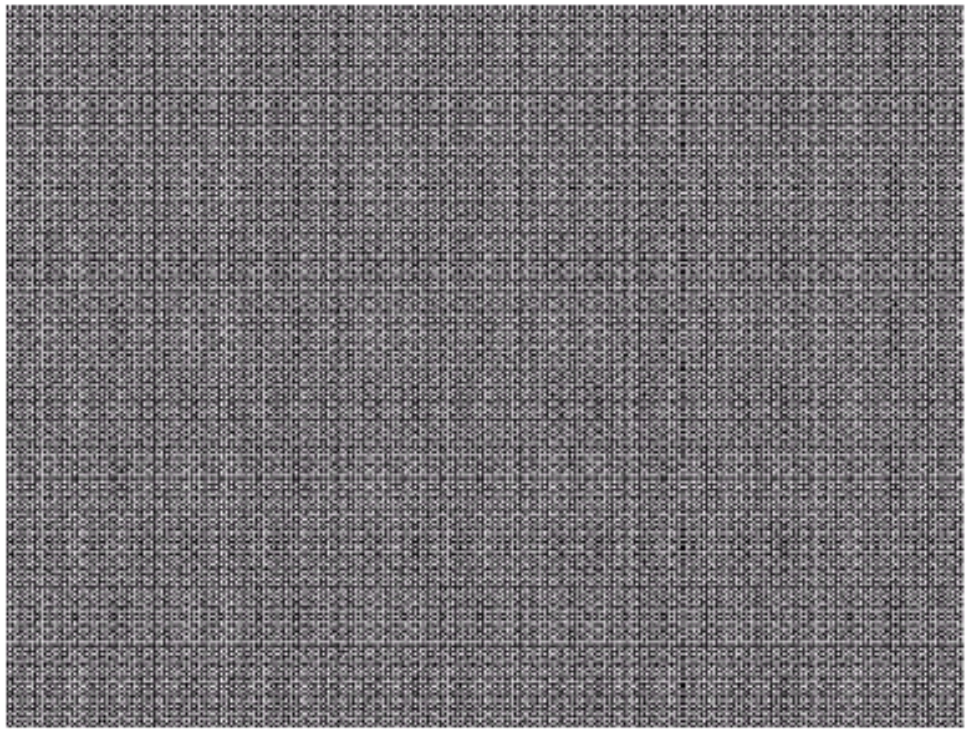
FIGURE 5.16
(a) Image corrupted by sinusoidal noise. (b) Spectrum of (a). (c) Butterworth bandreject filter (white represents 1). (d) Result of filtering. (Original image courtesy of NASA.)

Bandpass filters

- Bandpass filters perform opposition operation of the bandreject filters.

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

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



Notch filters

- Two types: notch reject filter H_{nr} and notch pass filter H_{np}
- Notch reject filter removes frequencies in a predefined neighbourhood about a centre frequency (u_0,v_0).
- Notch pass filter passes frequencies in a predefined neighbourhood about a centre frequency (u_0,v_0).
- They are related by

$$H_{np} = 1 - H_{nr}$$

Notch reject filters

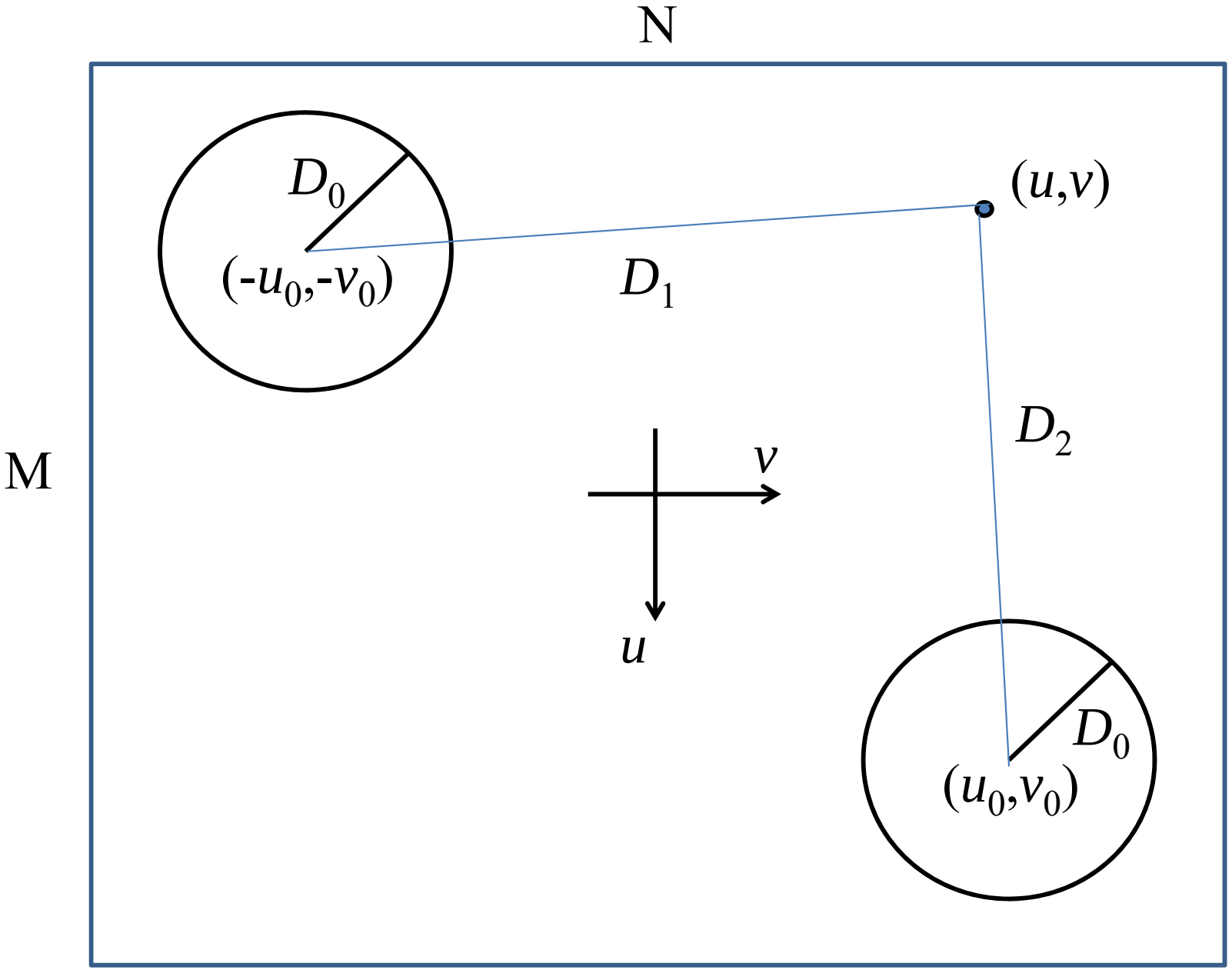
- Ideal notch reject filters

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

where $$D_2(u,v) = \sqrt{\left(u - u_0\right)^2 + \left(v - v_0\right)^2}$$

$$D_1(u,v) = \sqrt{\left(u + u_0\right)^2 + \left(v + v_0\right)^2}$$

Notch reject filters



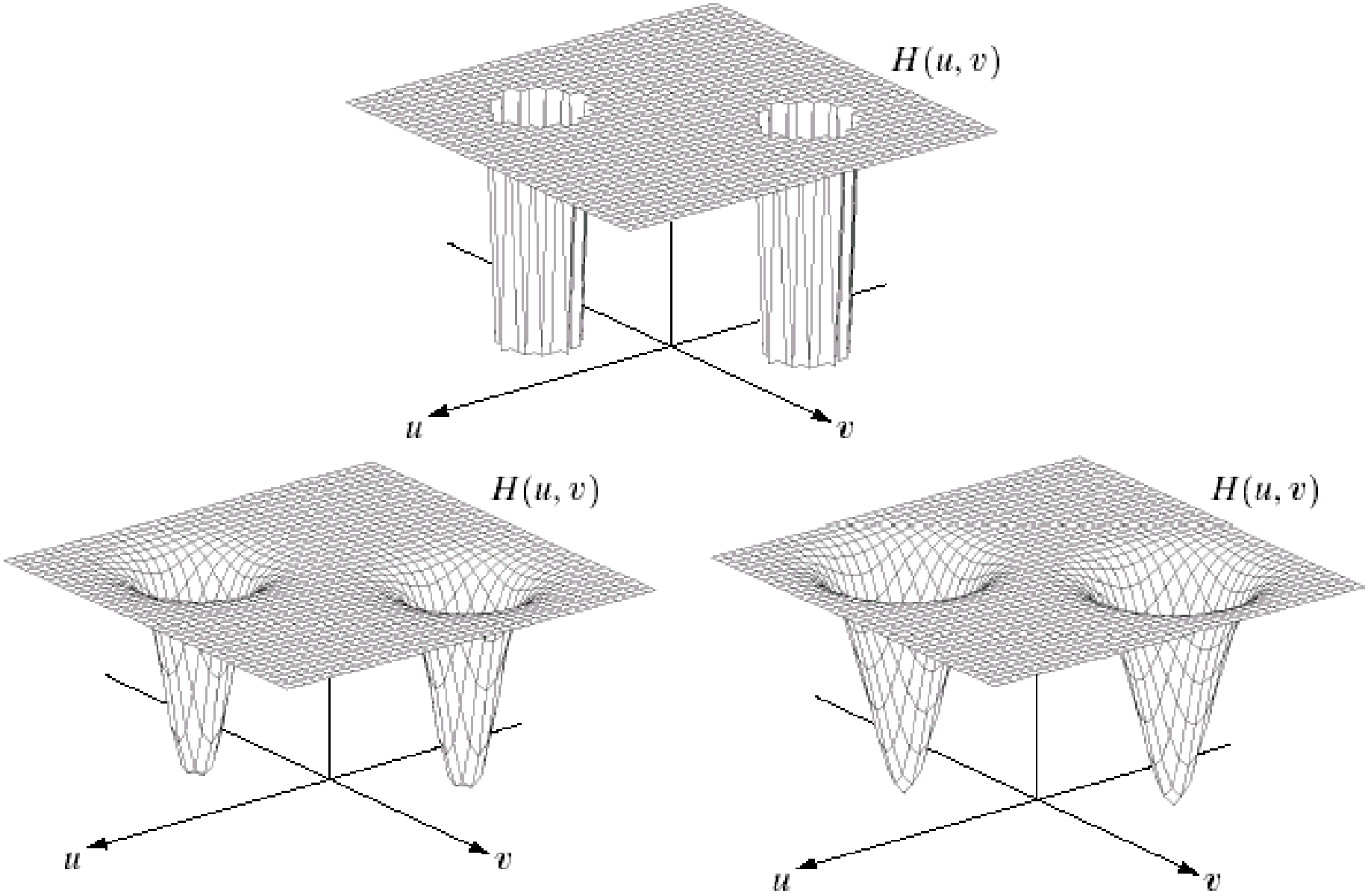
Notch reject filters

- Butterworth notch reject filters

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

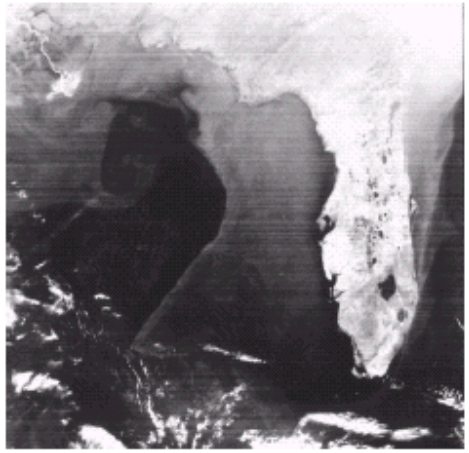
- Gaussian notch reject filters

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



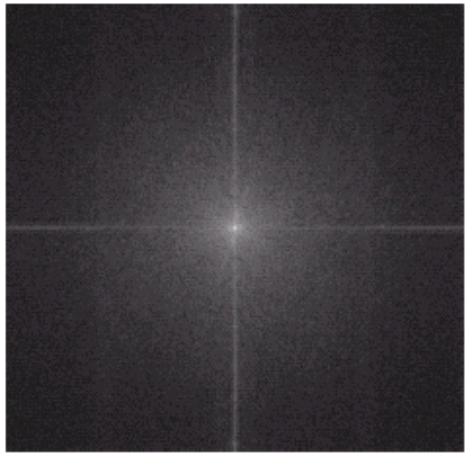
a
b c

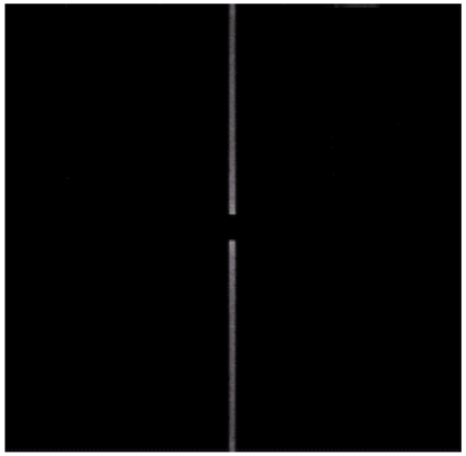
FIGURE 5.18 Perspective plots of (a) ideal, (b) Butterworth (of order 2), and (c) Gaussian notch (reject) filters.



Input image

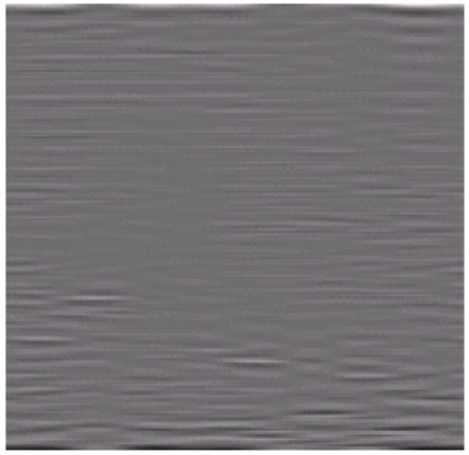
Frequency representation

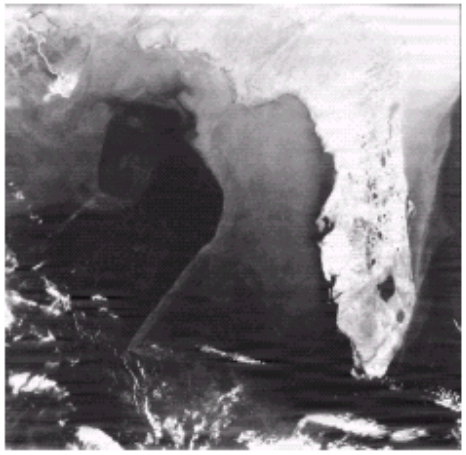




Notch pass filter

Filter output





Enhanced 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.)

A model of the image degradation and restoration process

- The degradation process is modelled as a degradation function H, which operates on an input image to produce a degraded image.
- $f(x,y)$ = input image
- $g(x,y)$ = degraded image
- $\eta(x,y)$ = additive noise (Greek:eta) (introduced before)
- The objective is to obtain an estimate $\hat{f}(x,y)$ of the original image $f(x,y)$.

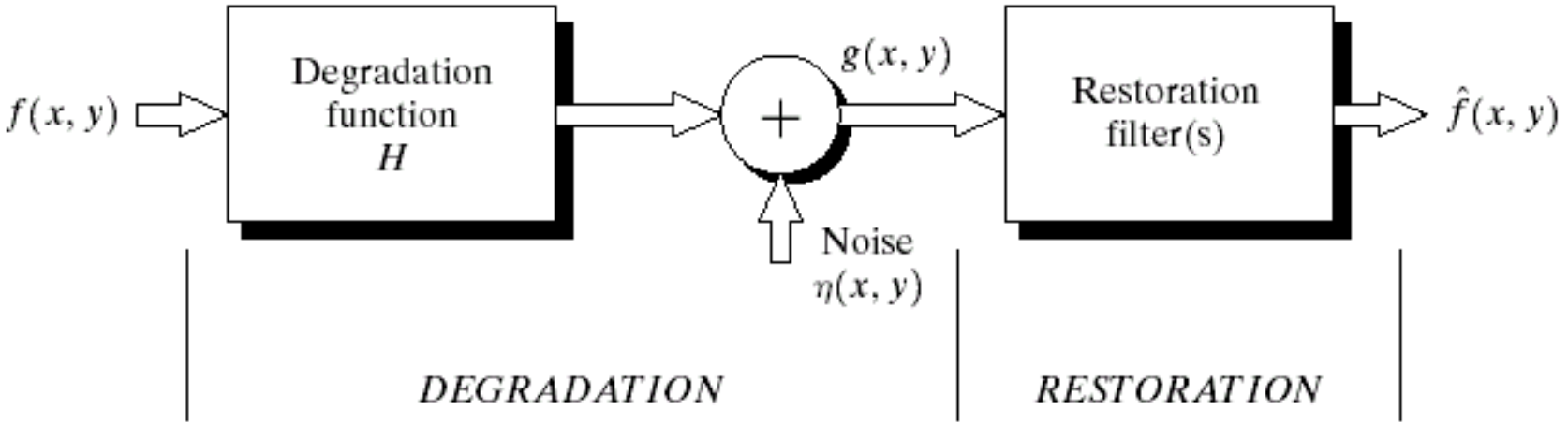


FIGURE 5.1 A model of the image degradation/ restoration process.

Degradation process

- If degradation is a linear, position-invariant process, then the degraded image is given in the spatial domain by

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

- Noise term is additive, rather than multiplicative. * indicates convolution.

- The equivalent frequency representation.

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

- Convolution in the spatial domain is analogous to multiplication in the frequency domain

Linear, Position-Invariant Degradation

A linear, position-invariant degradation function H_d can be modelled as the convolution of a function h in spatial domain.

$$g(x,y) = H_d\left[f(x,y)\right] + \eta(x,y)$$

spatial domain

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

frequency domain

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

Restoration based on Degradation Function *H*

1. Inverse filtering
2. Minimum mean square error (Wiener) filtering
3. Geometric mean filter

Inverse filtering

1. If the degradation function H is known, then the estimated image is given by

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

2. $G(u,v)$ = degraded, observed image.
3. $H(u,v)$ can be estimated by observation, experimentation or modelling.
4. The effect of noise is negligible in the strong-signal area.

4. Since
$$G(u,v) = F(u,v)H(u,v) + N(u,v)$$
5. then
$$\hat{F}(u,v) = F(u,v) + \frac{N(u,v)}{H(u,v)}$$
6. This means that the true image F can never be recovered if noise N is not known exactly.
7. Also, if degradation function H becomes small or zero, then N/H becomes very large and dominates the calculation. It deteriorates the estimation of F.
8. Solution: limit the analysis to frequencies near the origin in order to reduce the probability of having zero or small values of H.

original image

Image degradation based on a model/filter H with parameter k

a b
c d

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

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

The degradation model H is derived based on the physical characteristics of atmospheric turbulence [Hufnagel & Stanley 1964]

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.

The degradation model H is derived based on the physical characteristics of atmospheric turbulence [Hufnagel & Stanley 1964]

$$H(u,v) = e^{-k[(u-M/2)^2+(v-N/2)^2]^{5/6}} \qquad k = 0.0025$$

Minimum mean square error (Wiener) filtering

1. If we minimise the mean square error between the estimated image $\hat{f}(x,y)$ and the true, uncorrupted image $f(x,y)$;
2. we assume that
   a. noise and true image are statistically uncorrelated;
   b. noise or the image has zero mean;
   c. degradation function is linear;
   d. degradation function $H(u, v)$ is known;
   e. power spectrum of the noise $S_\eta(u,v) = |N(u,v)|^2$ is known;
   f. power spectrum of the true, un-degraded, uncorrupted image $S_f(u,v) = |F(u,v)|^2$ is known;

http://en.wikipedia.org/wiki/Wiener_filter
https://en.wikipedia.org/wiki/Wiener_deconvolution

Minimum mean square error (Wiener) filtering

3. then the estimated image is given by

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

4. The Wiener filter incorporates the degradation function and statistical characteristics of noise into the restoration process.
5. It does not have the same problem as the inverse filter with zeros in the degradation function.

http://en.wikipedia.org/wiki/Complex_conjugate

Minimum mean square error (Wiener) filtering

6. If neither N nor F is known, then the solution is to approximate

$$\frac{S_\eta(u,v)}{S_f(u,v)} = K$$

7. The approximated, estimated image is given by

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

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.

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

$$k = 0.0025$$

The degradation model H is derived based on the physical characteristics of atmospheric turbulence [Hufnagel & Stanley 1964]

Noise level

decreasing

Motion blur Inverse filter Wiener filter

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.

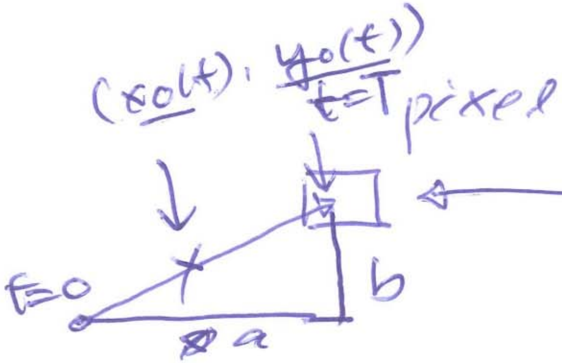
$$H(u,v) = \frac{T}{\pi(ua+vb)} sin[\pi(ua+vb)] e^{-j\pi(ua+vb)}$$

a= b = 0.1

T = 1

j= $\sqrt{-1}$

Motion blur: https://en.wikipedia.org/wiki/Motion_blur



(unknown)
known in degradation

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

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

⋮

$$G(u,v) = F(u,v) \underbrace{\int_0^T e^{-j2\pi[u x_0(t) + v y_0(t)]}\,dt}_{H(u,v)}$$

Geometric mean filter

1. Geometric mean filter

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

2. When alpha = 1, it becomes the inverse filter
3. When alpha = 0 and beta = 1, it becomes the standard Wiener filter.
4. It represents a family of filters combined into a single expression.

Geometric Transformations

1. Also called '*rubber-sheet transformations*'.
2. Useful for image distortion correction.
3. Two things to consider
   a. Spatial Transformations
   b. Grey-level Assignments

      (f = original/corrected image, g = distorted image)

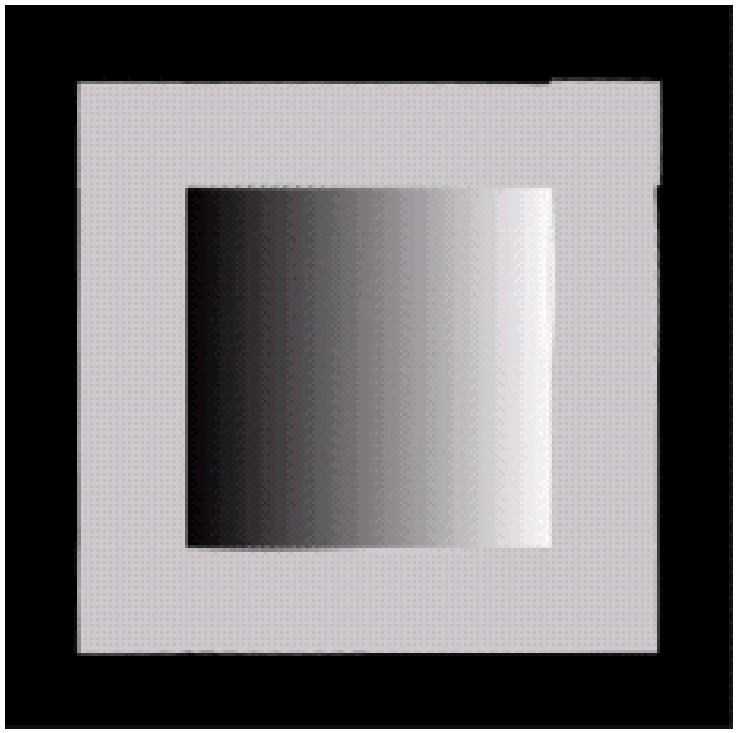
      i. Scenario A: f is unknown and g is known.

      ii. Scenario B: f is known and g is unknown.

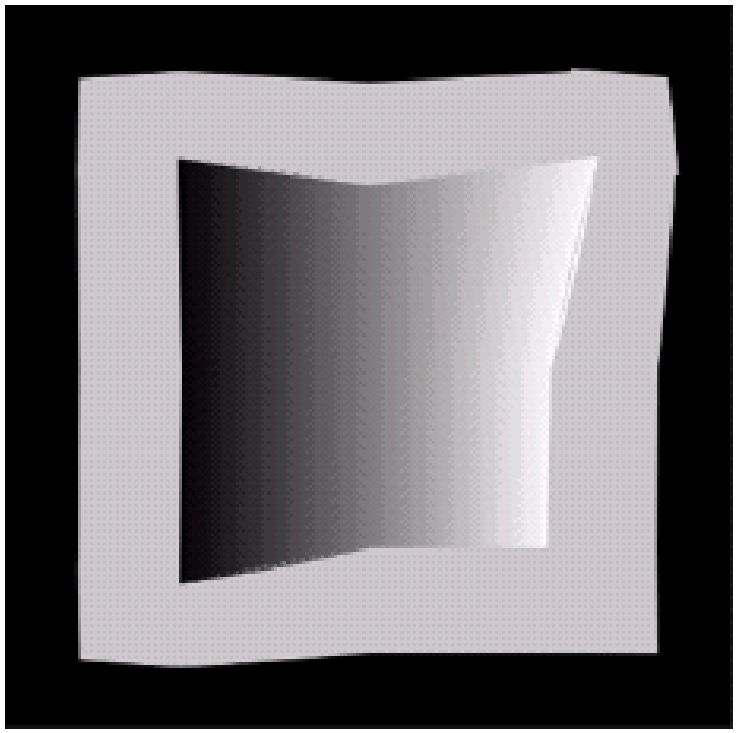
Examples: http://www.corrmap.com/features/rubber-sheeting_transformation.php
https://goo.gl/images/VWlEJ4

Scenario A: f is unknown and g is known

Distortion correction



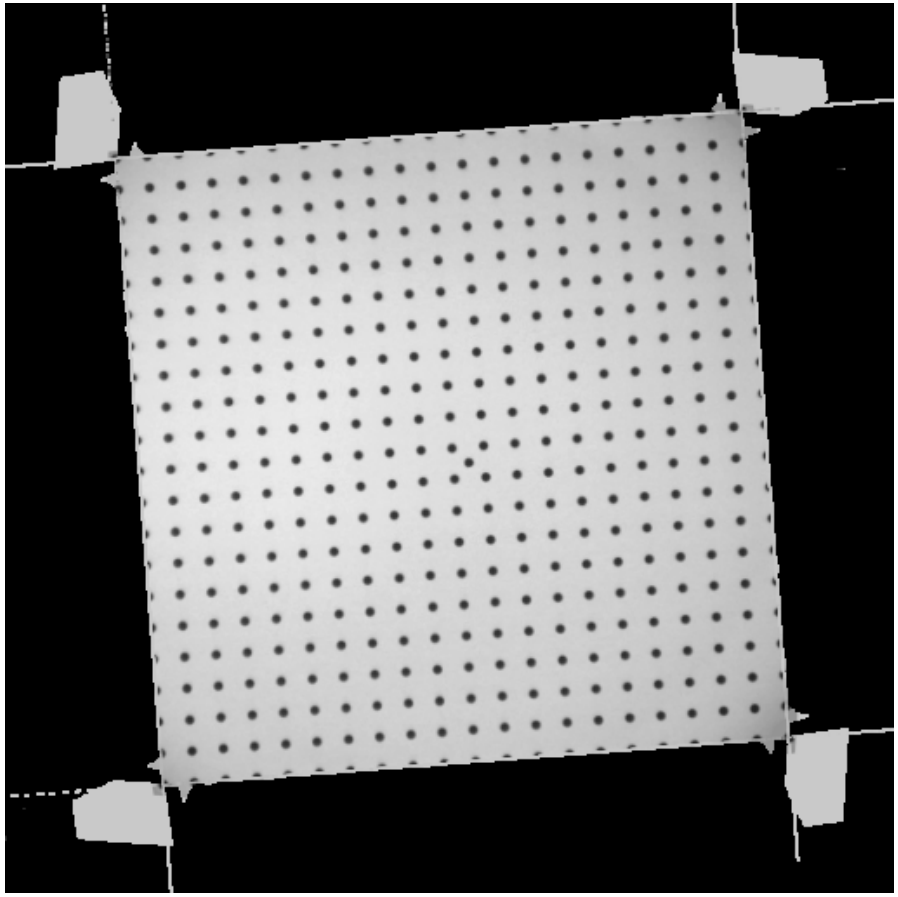
Corrected image
$f(x,y)$



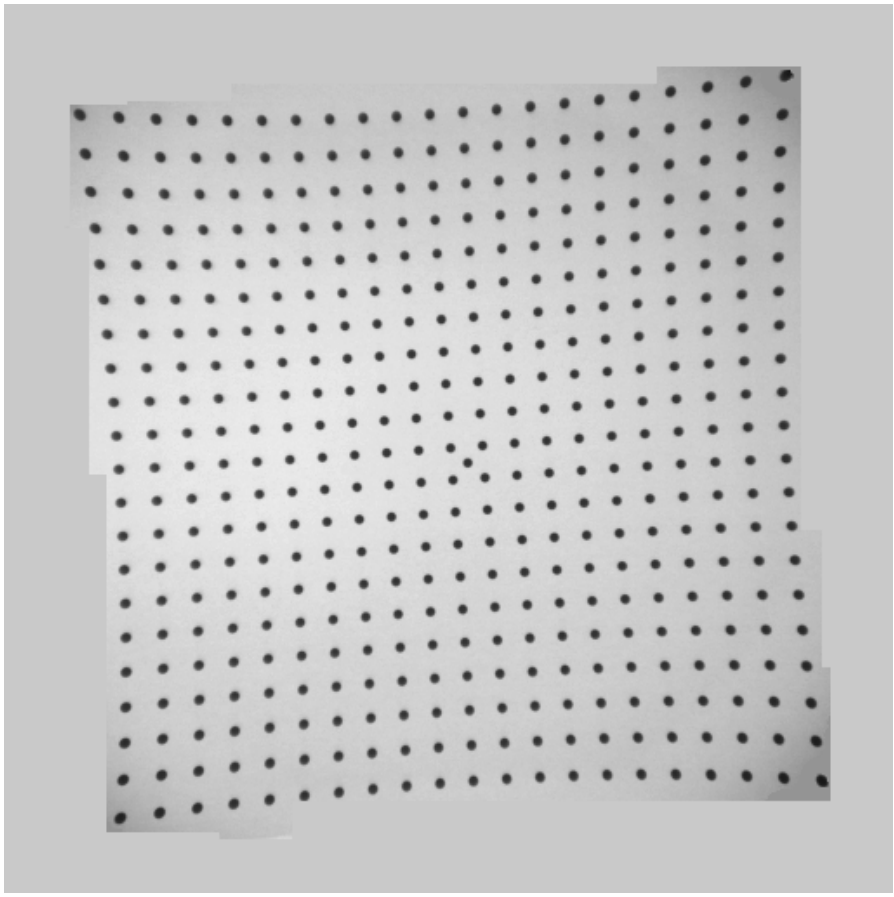
Distorted image
$g(x',y')$

Distortion: http://en.wikipedia.org/wiki/Distortion_(optics)
Lens Distortion: https://helpx.adobe.com/photoshop/using/correcting-image-distortion-noise.html
Correction: http://www.kenrockwell.com/tech/correctinglensdistortion.htm

Example: X-ray image distortion correction



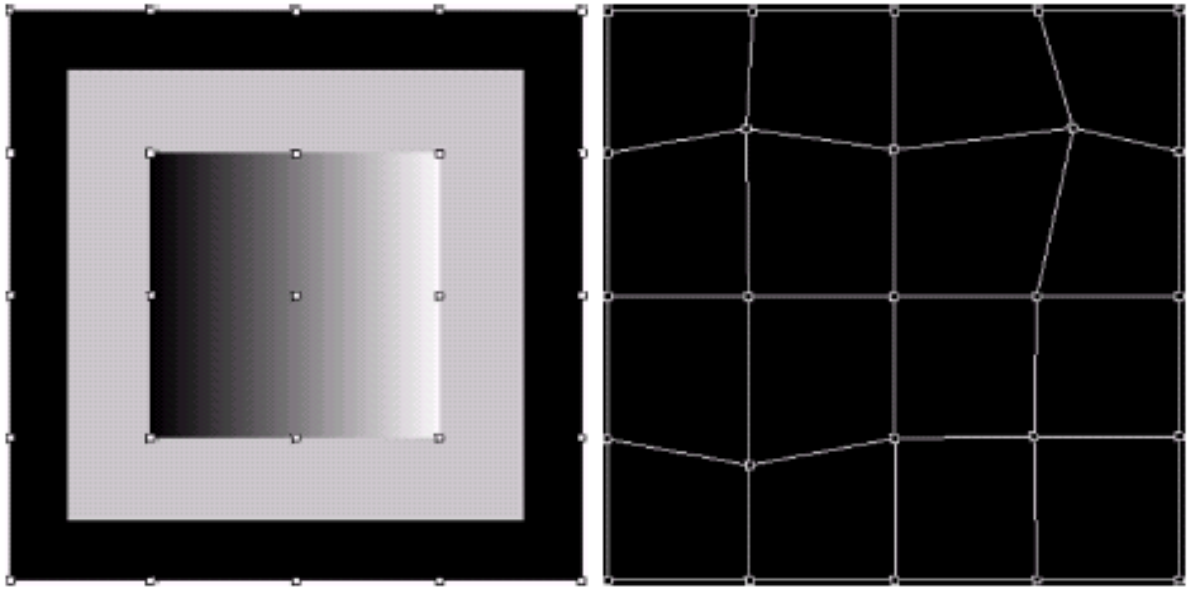
Corrected image
$f(x,y)$



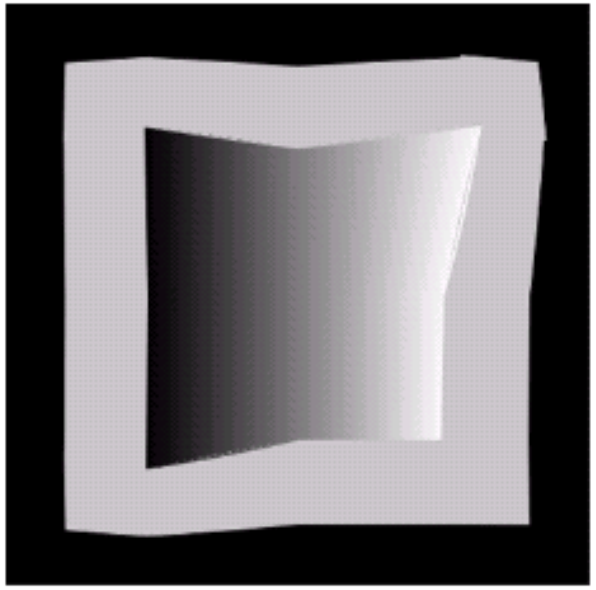
Distorted image
$g(x',y')$

Scenario B: f is known and g is unknown

Distortion



Input image
f(x,y)



Distorted image
g(x’,y’)

Spatial Transformations

1. Assume that $f(x,y)$ is the original image, and $g(x',y')$ is the distorted image.
2. Spatial transformation function models the distortion from the original image f to the distorted image g.

$$x' = r\left(x, y\right)$$

$$y' = s\left(x, y\right)$$

3. Tiepoints/landmarks – a subset of pixels (points) whose locations in the distorted and original images are known precisely.

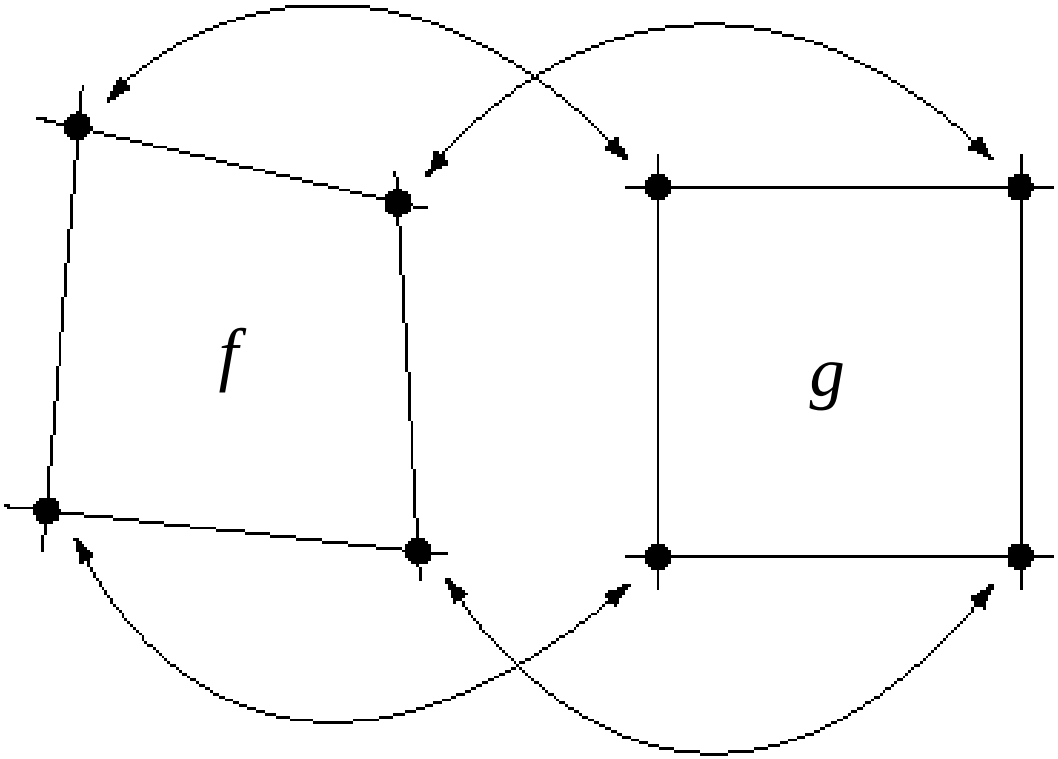


FIGURE 5.32 Corresponding tiepoints in two image segments.

4. If tiepoints are given, then it is possible to estimate the spatial transformation. For example,

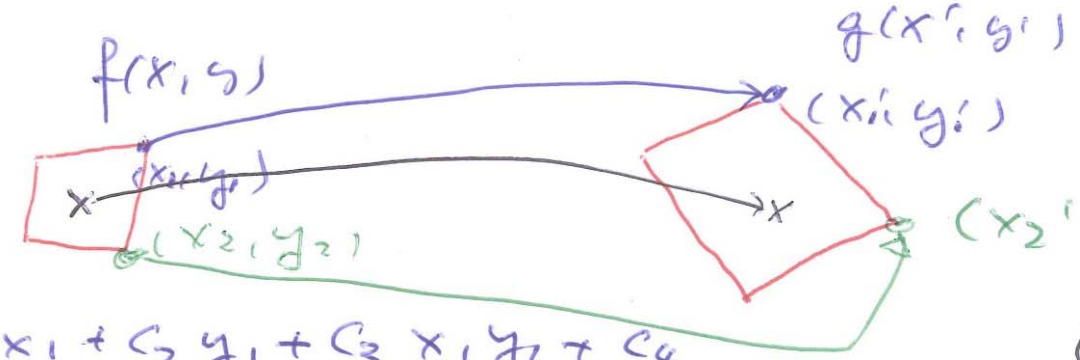
spatial transformation is modelled as

$$x' = r(x, y) = c_1 x + c_2 y + c_3 xy + c_4$$

$$y' = s(x, y) = c_5 x + c_6 y + c_7 xy + c_8$$

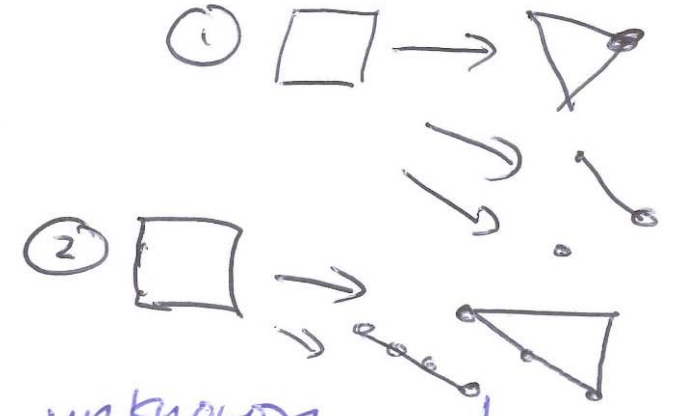
If 4 pairs of $(x', y') \leftrightarrow (x, y)$ are given,

then all coefficients can be estimated.



$$x_1' = c_1 x_1 + c_2 y_1 + c_3 x_1 y_1 + c_4$$
$$y_1' = c_5 x_1 + c_6 y_1 + c_7 x_1 y_1 + c_8$$

$$x_2' = c_1 x_2 + c_2 y_2 + c_3 x_2 y_2 + c_4$$
$$y_2' = c_5 x_2 + c_6 y_2 + c_7 x_2 y_2 + c_8$$



known

$$\begin{bmatrix} x_1' & x_2' & x_3' & x_4' \\ y_1' & y_2' & y_3' & y_4' \end{bmatrix} = \begin{bmatrix} c_1 & c_2 & c_3 & c_4 \\ c_5 & c_6 & c_7 & c_8 \end{bmatrix} \begin{bmatrix} x_1 & \cdots & 1 \\ y_1 & & \\ x_1 y_1 & & \\ 1 & & \end{bmatrix}$$

unknown ↔ C

$\overleftrightarrow{X}$

$\overleftrightarrow{C} = \overleftrightarrow{X}\,\overleftrightarrow{M}^{-1}$

$\overleftrightarrow{M}$

Grey-level Assignments

1. We need to consider grey-level assignments in both scenarios (both assume that the spatial transformation is known),
   a. distorted image g is known, we want to find the original f image; or
   b. original image f is known, we want to find the distorted image g.

Grey-level Assignments

2. Scenario A: after transformation, an integral point (x,y) in the original point may map to a non-integral point (x',y').
3. Solution 1: the non-integral point (x',y') maps to its nearest neighbour (*nearest neighbour approach*), and the intensity value at (x',y') is then mapped/copied to (x,y).

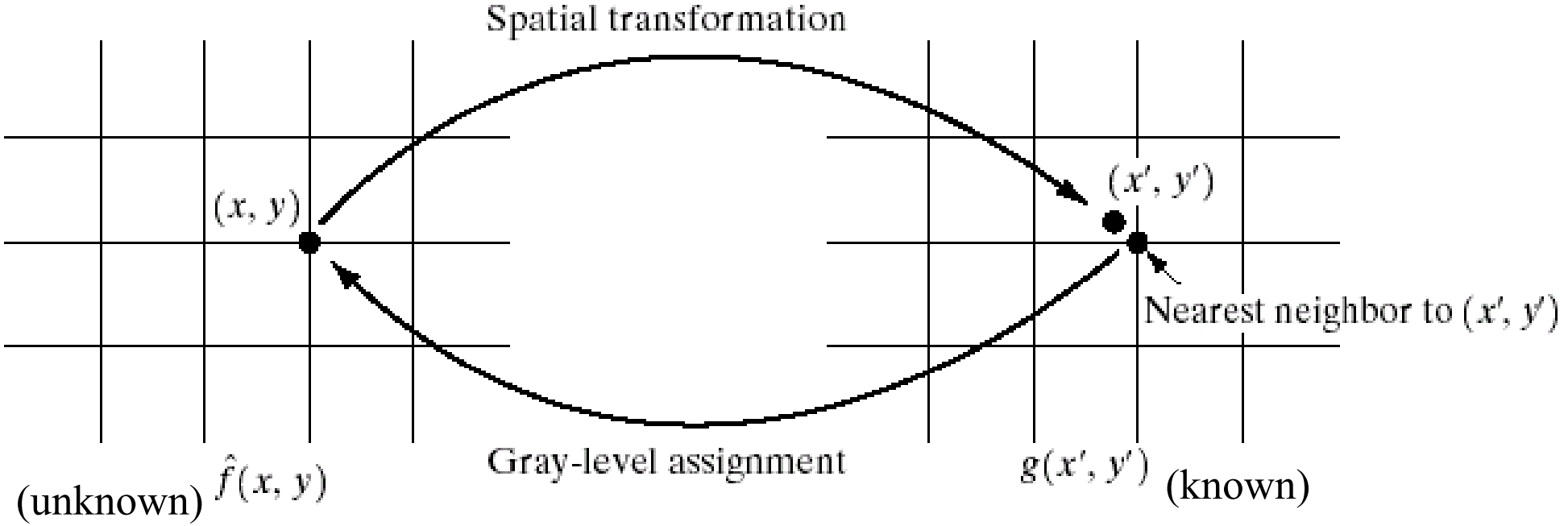
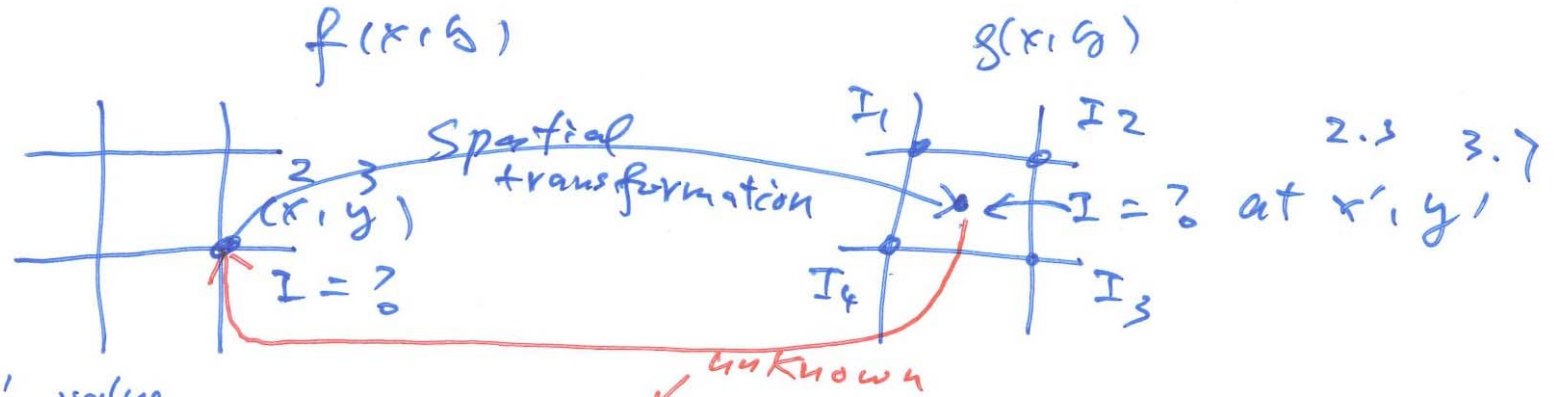


FIGURE 5.33 Gray-level interpolation based on the nearest neighbor concept.



unknown

Given x', y', value of I(x', y') can be estimated

$$I(x', y') = \boxed{a}x' + \boxed{b}y' + \boxed{c}x'y' + \boxed{d}$$

4 eq^n s

4 unknown

$$\begin{bmatrix} I_1 = a x_1' + b y_1' + c x_1' y_1' + d \\ I_2 = a x_2' + b y_2' + c x_2' y_2' + d \\ I_3 = a x_3' + b y_3' + c x_3' y_3' + d \\ I_4 \end{bmatrix}$$

I_4 known

$I_1 \; I_2 \; I_3$ unknown

known

4×4

1×4

1×4

$$\begin{bmatrix} I_1 & I_2 & I_3 & I_4 \end{bmatrix} = \begin{bmatrix} a & b & c & d \end{bmatrix} \begin{bmatrix} x_1' & x_2' & x_3' & x_4' \\ y_1' & & & \\ x_1'y_1' & & M & \\ 1 & & & \end{bmatrix}$$

Values of a, b, c, d are unknown then

$$\begin{bmatrix} a & b & c & d \end{bmatrix} = \underbrace{\begin{bmatrix} I_1 & I_2 & I_3 & I_4 \end{bmatrix}}_{\text{known}} \underbrace{M^{-1}}_{\text{known}}$$

Grey-level Assignments

4. Solution 2: (*bilinear interpolation*). Since the grey levels of the four integral nearest neighbours of a non-integral pair (*x'*,*y'*) are known, the grey value can be interpolated from its neighbour's grey values by using (4 points and 4 equations)

$$v(x', y') = ax' + by' + cx'y' + d$$

5. Then, the interpolated value becomes the intensity value at (*x*,*y*).

Bilinear interpolation: https://en.wikipedia.org/wiki/Bilinear_interpolation
http://supercomputingblog.com/graphics/coding-bilinear-interpolation/
http://www.cambridgeincolour.com/tutorials/image-interpolation.htm

Grey-level Assignments

6. Scenario B: after transformation, an integral point (x,y) in the original point may map to a non-integral point (x',y').
7. Solution: the non-integral point (x',y') maps to its nearest neighbour (*nearest neighbour approach*), and the intensity value at (x,y) is then mapped to (x',y').

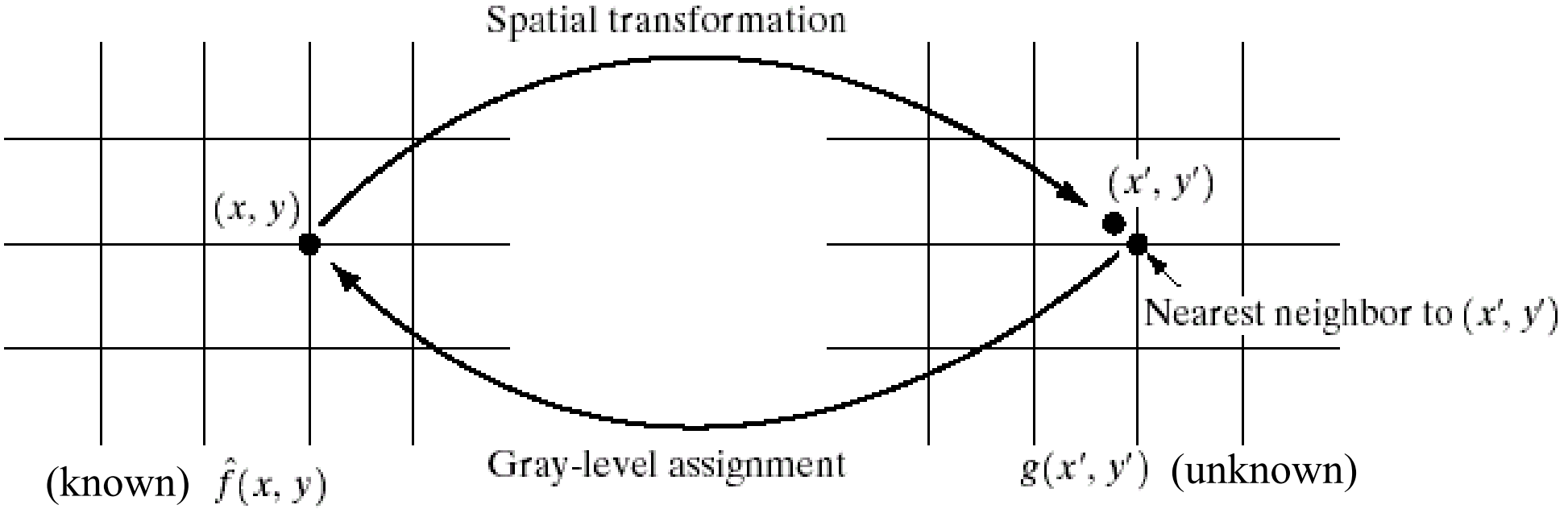
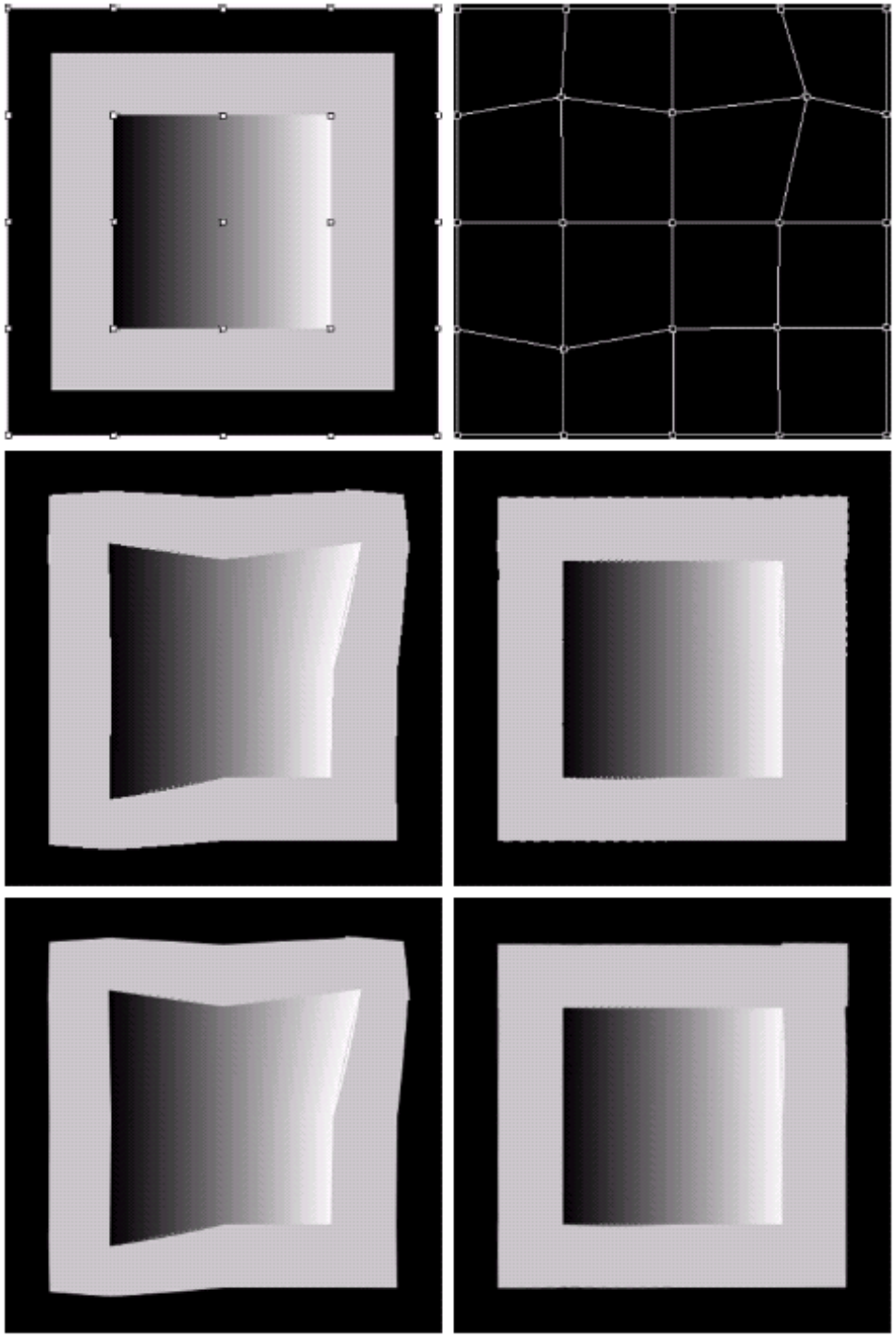


FIGURE 5.33 Gray-level interpolation based on the nearest neighbor concept.



a b
c d
e f

FIGURE 5.34 (a) Image showing tiepoints. (b) Tiepoints after geometric distortion. (c) Geometrically distorted image, using nearest neighbor interpolation. (d) Restored result. (e) Image distorted using bilinear interpolation. (f) Restored image.