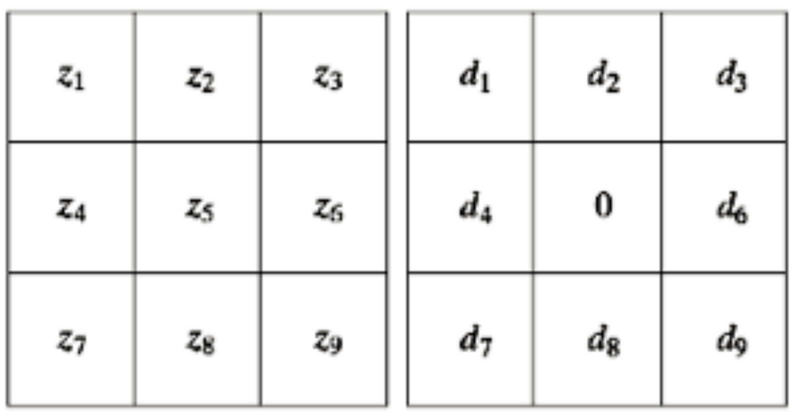
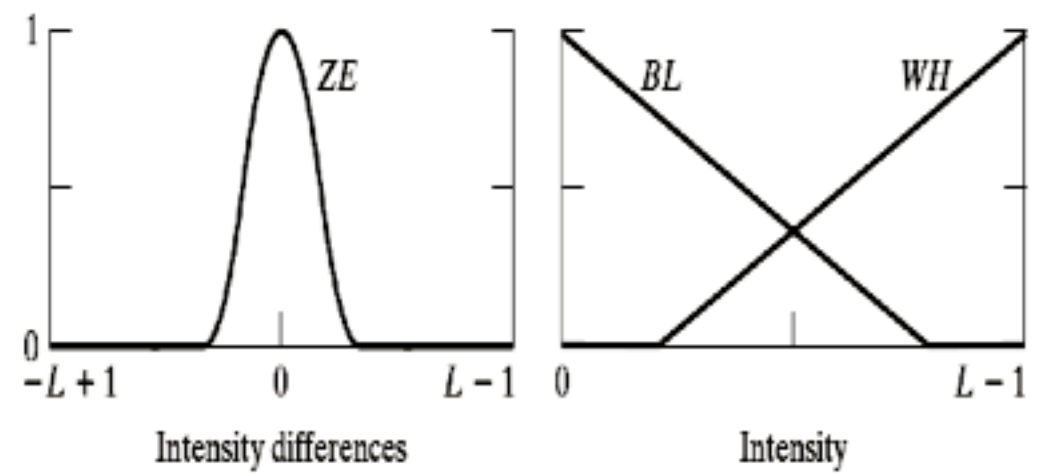
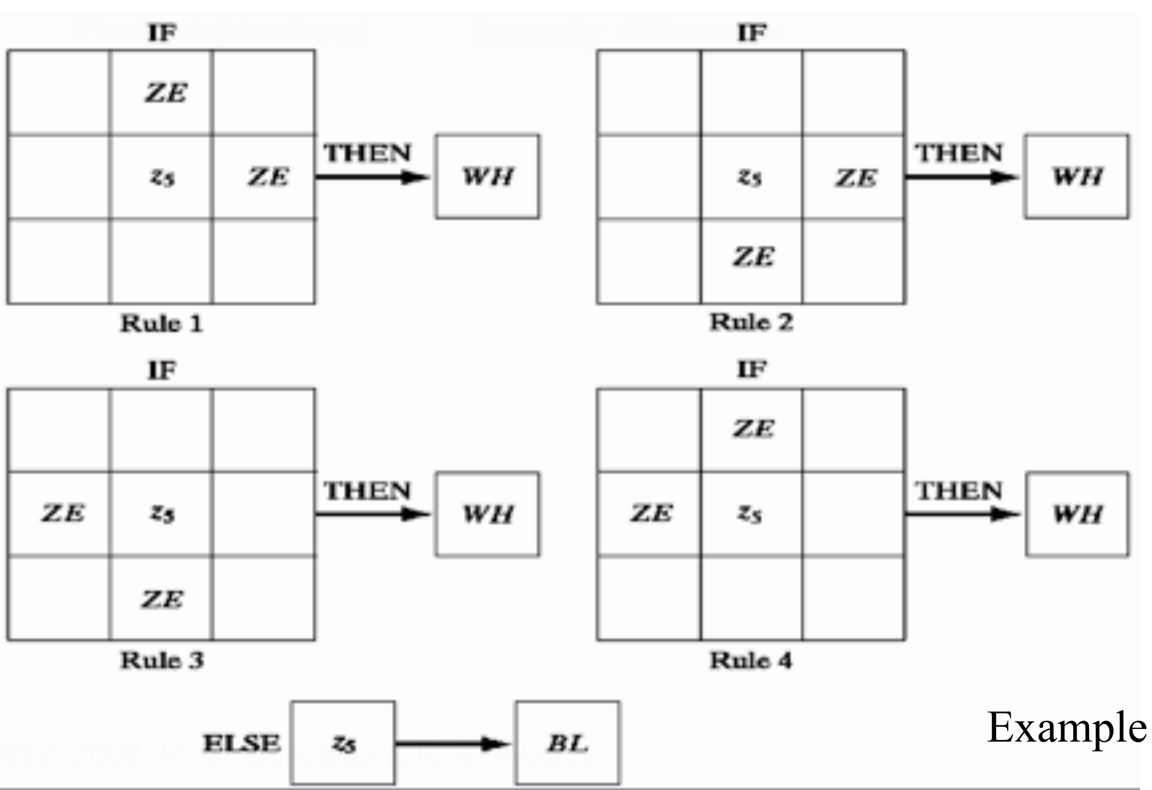
Fuzzy sets for spatial filtering.

The basic approach is to define neighborhood properties that capture the essence of the image.





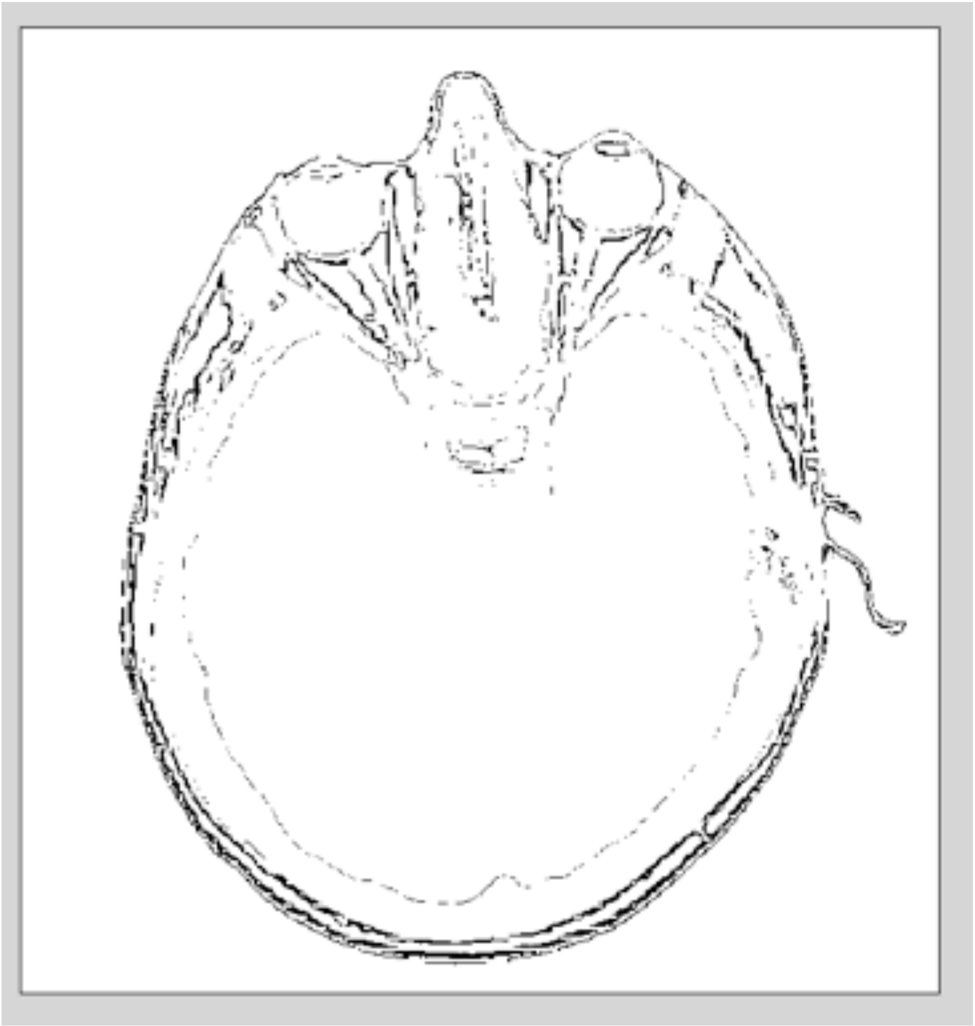


Uniform region: if neighborhood pixels have "zero" intensity difference.

Example of inference rule using 4-neighborhood

a b c

FIGURE 3.59 (a) CT scan of a human head. (b) Result of fuzzy spatial filtering using the membership functions in Fig. 3.57 and the rules in Fig. 3.58. (c) Result after intensity scaling. The thin black picture borders in (b) and (c) were added for clarity; they are not part of the data. (Original image courtesy of Dr. David R. Pickens, Vanderbilt University.)

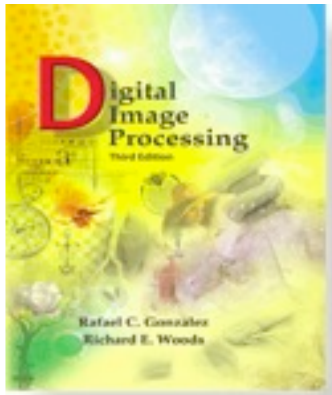


Result of the fuzzy algorithm for spatial filtering as implemented last lecture

Fuzzy techniques are computationally intensive because the rules must be applied to each pixel.

The use of singletons (simplified output) makes it possible to reduce the complexity.

A practical approach is to use fuzzy set techniques to determine the histogram of a well balanced image, then use faster methods like histogram matching to obtain similar results.



Skip chapter 4 (Fourier transforms) for the time being and jump to chapter 5.

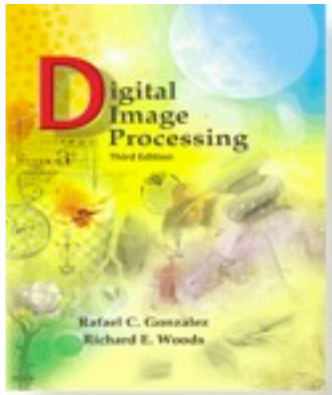
Chapter 5: Image degradation and restoraton

A model of image degradation:

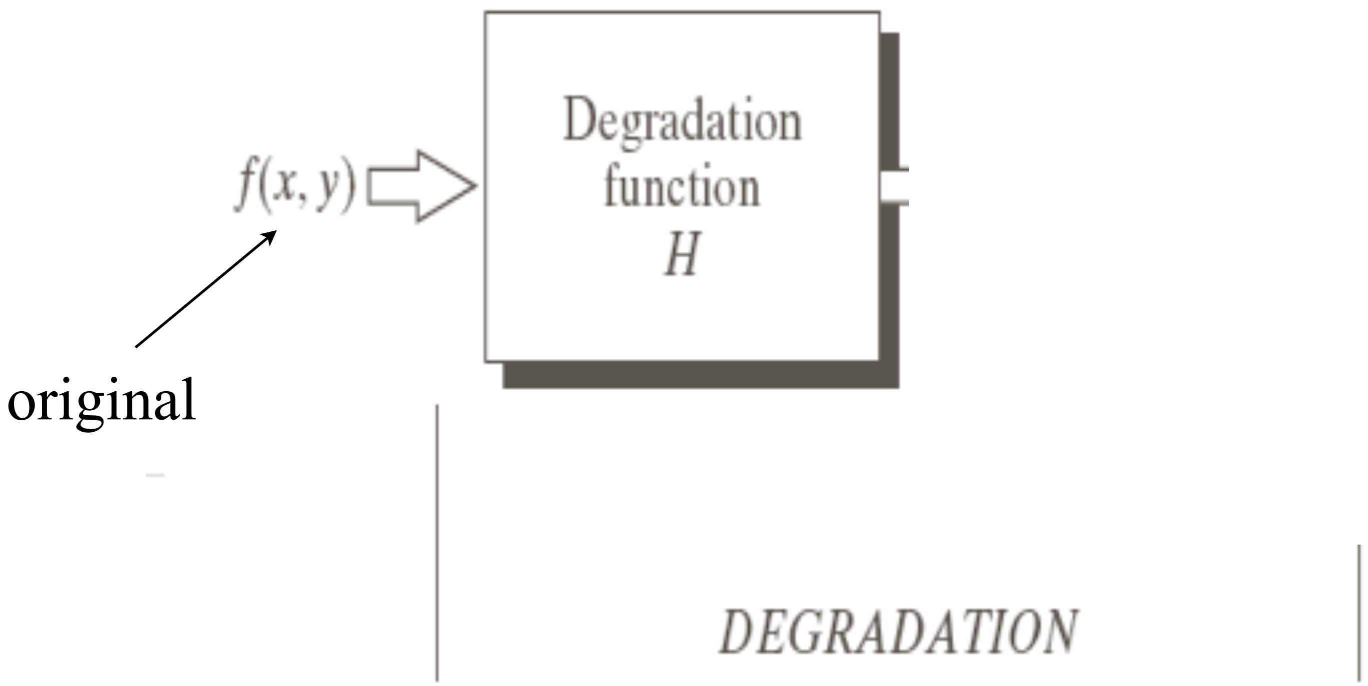
$f(x, y)$

original

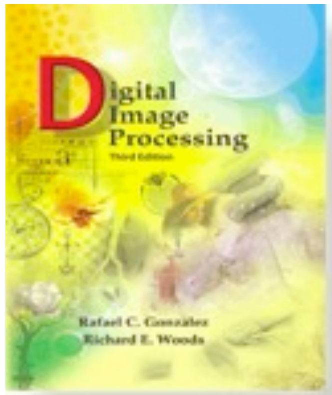
The original image is degraded by some process described by a function *h(x,y)*. In addiction, the image is then corrupted by some noise $\eta\,(x,y)$.



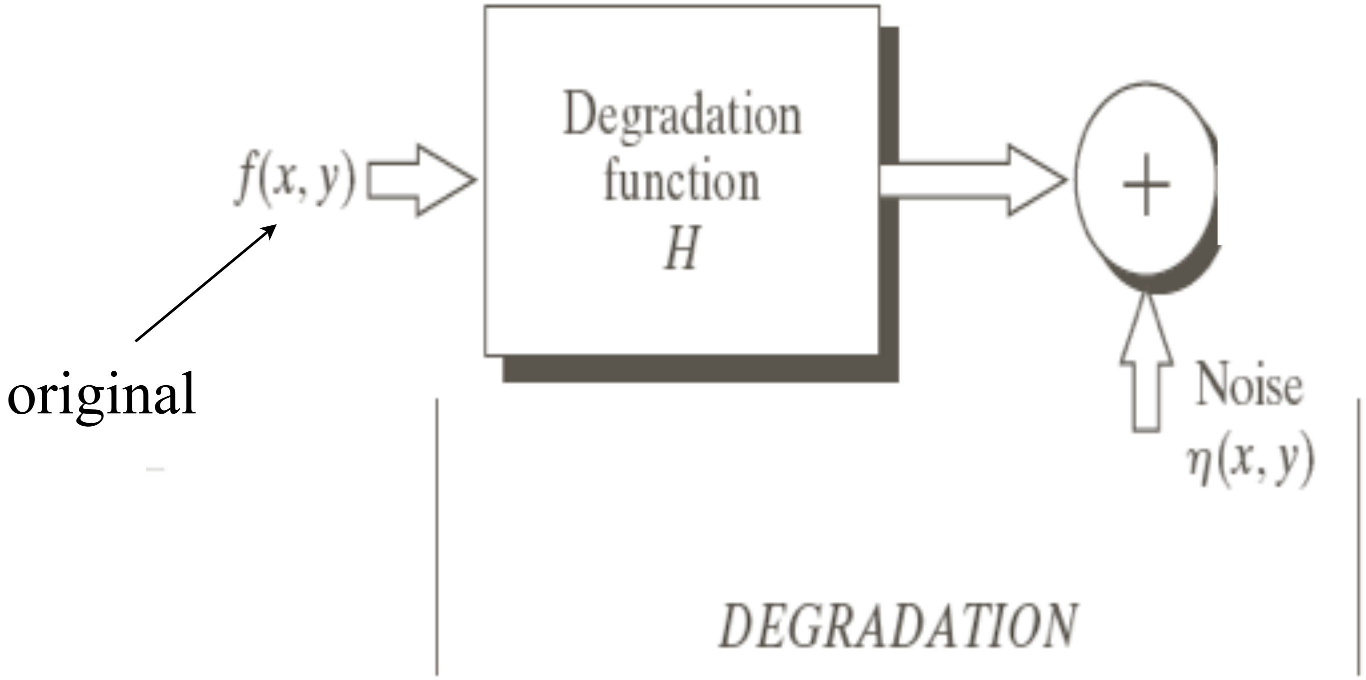
A model of image degradation:



The original image is degraded by some process described by a function *h(x,y)*. In addiction, the image is then corrupted by some noise $\eta\,(x,y)$.

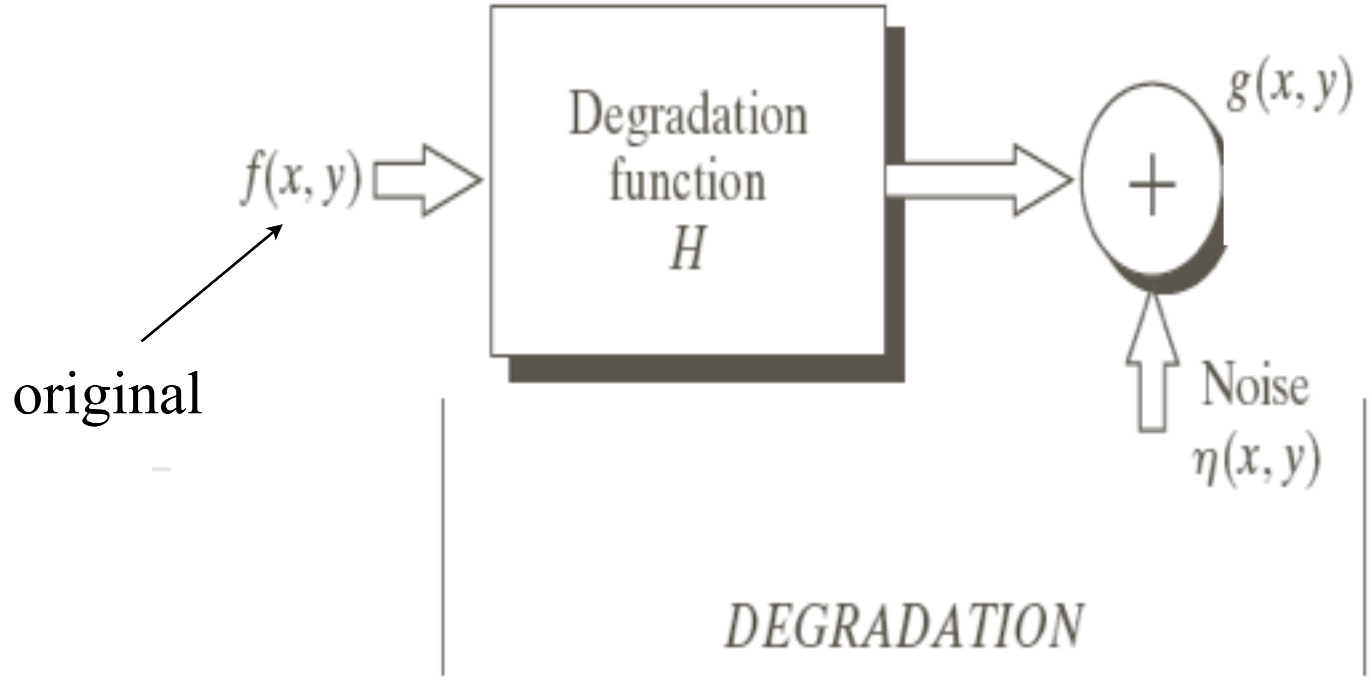


A model of image degradation:

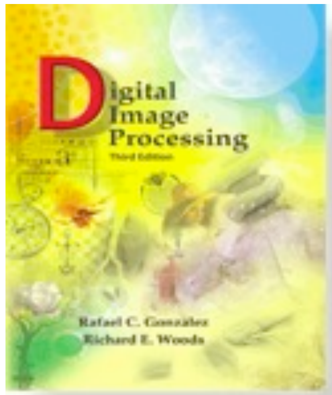


The original image is degraded by some process described by a function *h(x,y)*. In addiction, the image is then corrupted by some noise $\eta\,(x,y)$.

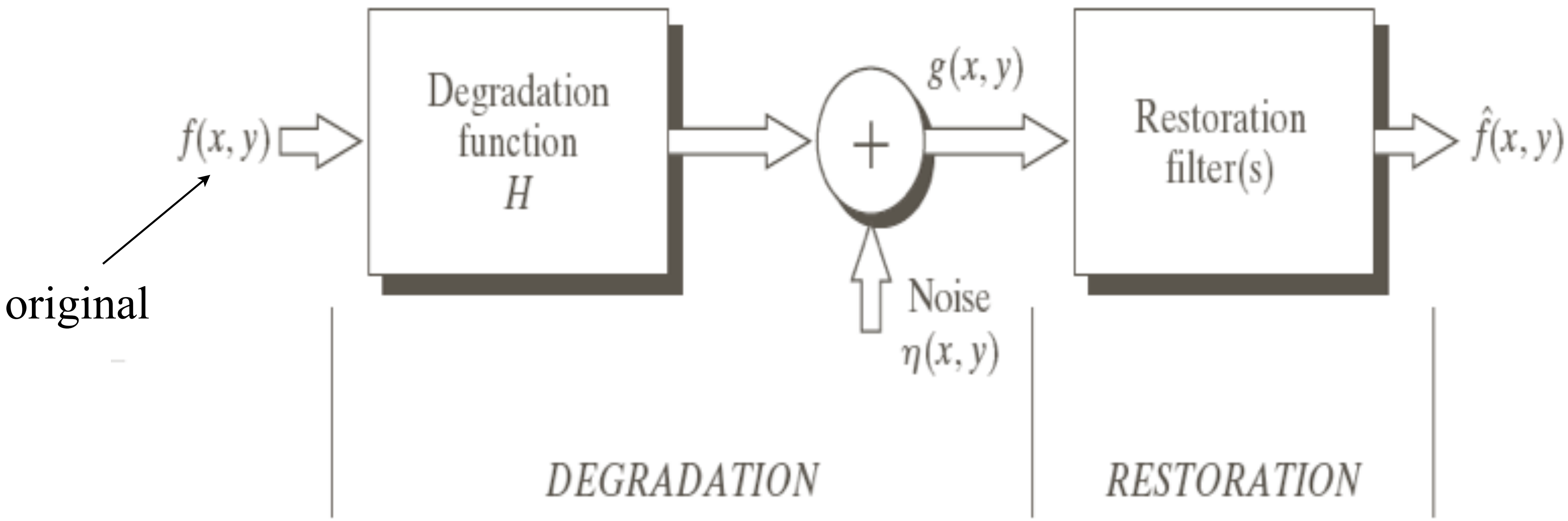
A model of image degradation:



The original image is degraded by some process described by a function $h(x,y)$. In addiction, the image is then corrupted by some noise $\eta(x,y)$.

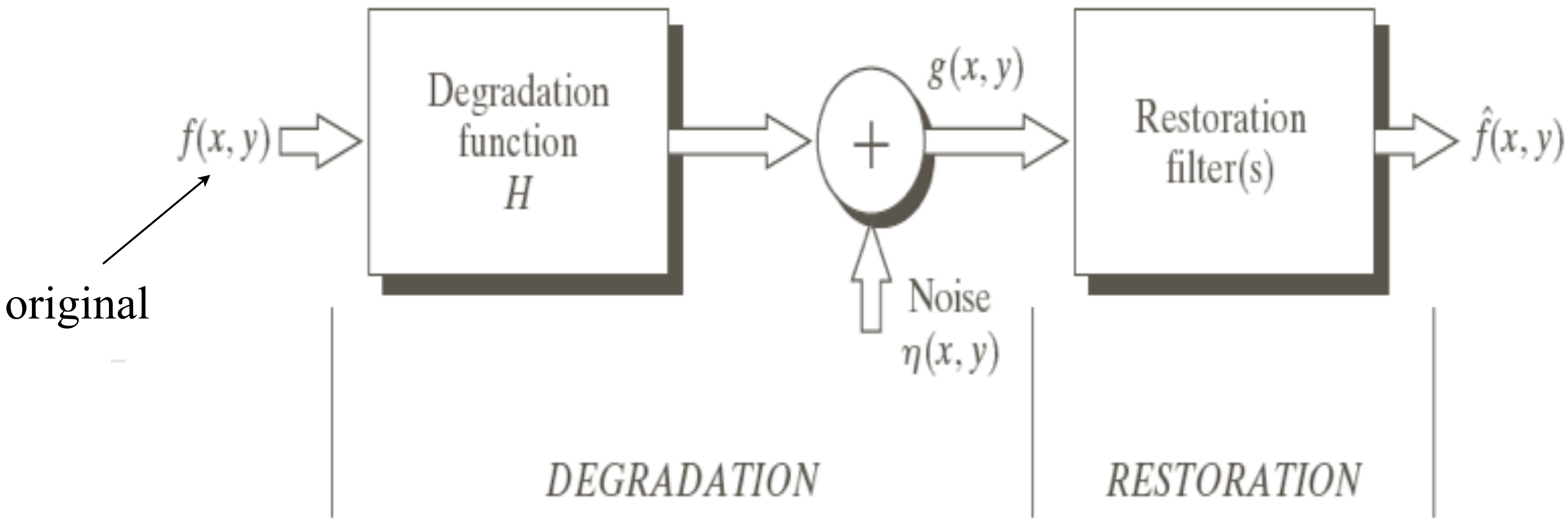


A model of image degradation:



The original image is degraded by some process described by a function $h(x,y)$. In addiction, the image is then corrupted by some noise $\eta(x,y)$.

A model of image degradation:

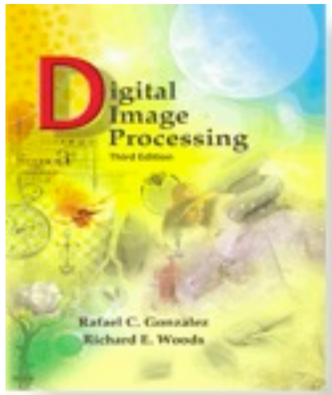


The original image is degraded by some process described by a function *h(x,y)*. In addiction, the image is then corrupted by some noise η *(x,y)*.

convolution

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

corruption original noise



convolution

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

observed corruption original noise

Degradation occurs during image acquisition/transmission.

In Fourier space:

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

Discrete transforms:

$$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M+vy/N)}$$

Spatial noise:

- independent of the spatial coordinates

- uncorrelated with the image

Statistical behavior of the noise: through Probability Density Functions

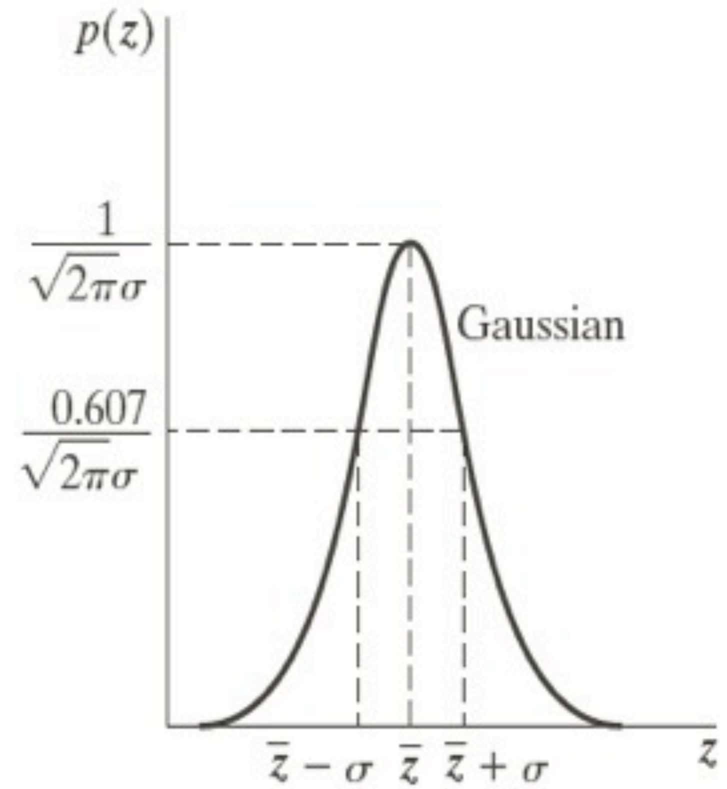
Periodic noise

- is depending on the spatial coordinates

Can be reduced in applying appropriate filters to the frequency domain.

Gaussian (or normal) noise:

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



Mean: $\bar{x}$

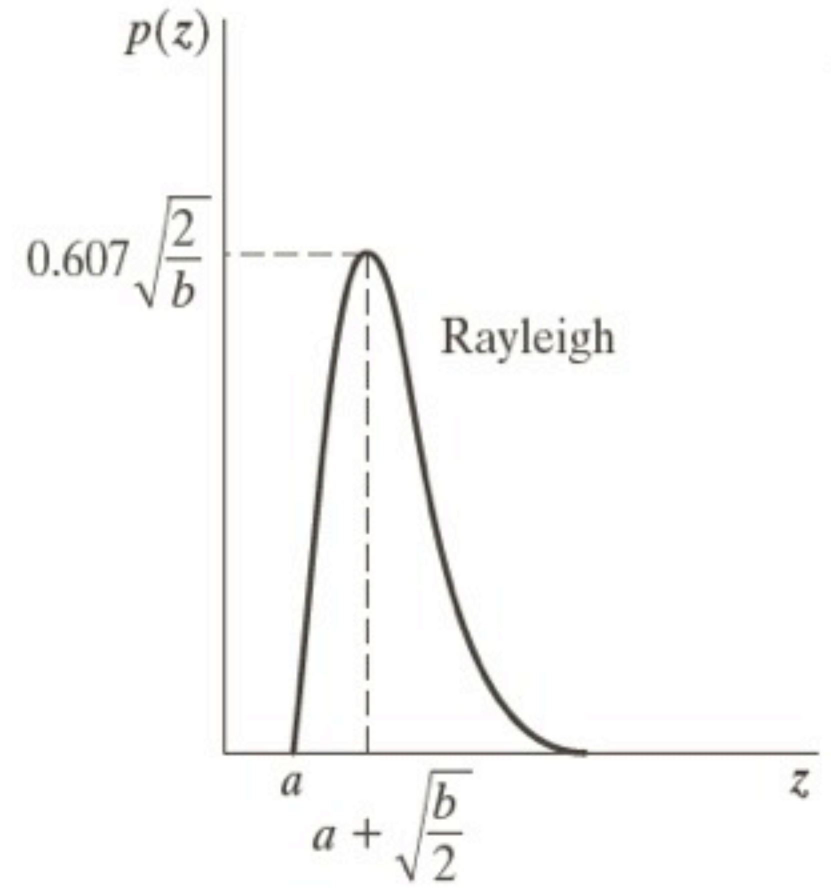
Standard deviation: σ

Variance: σ^2

70% of all the values are between the mean and the standard deviation
95% of all the values are between the mean and 2 σ

Rayleigh noise:

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

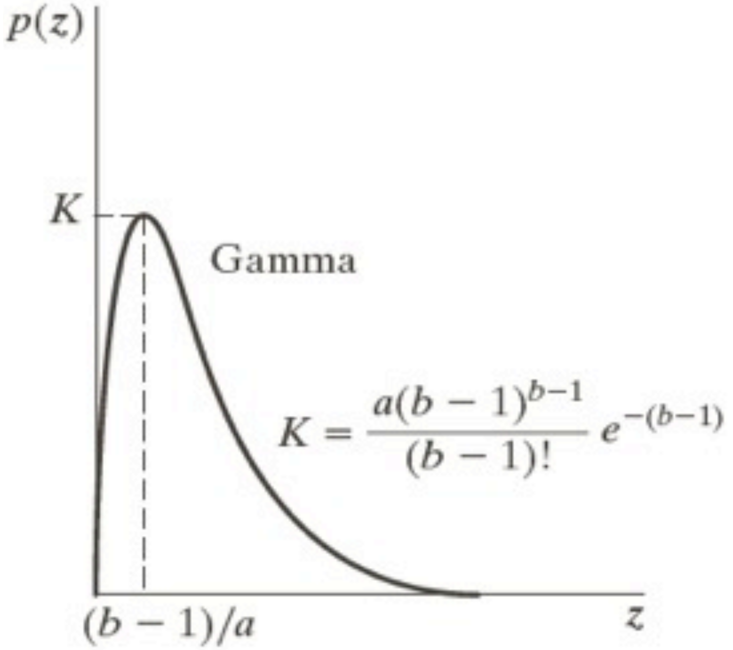


Mean: $\bar{z} = a + \sqrt{\pi b/4}$

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

Erlang (gamma) noise:

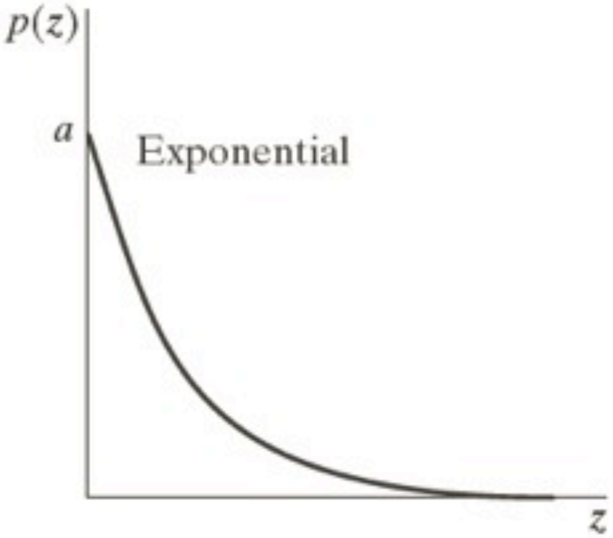
$$p(z) = \begin{cases} \frac{a^b z^{b-1}}{(b-1)!} e^{-az} & z \ge 0 \\ 0 & z < 0 \end{cases}$$



Mean: $\bar{z} = \frac{b}{a}$

Variance: $\sigma^2 = \frac{b}{a^2}$

Exponential noise: *(a>0)*

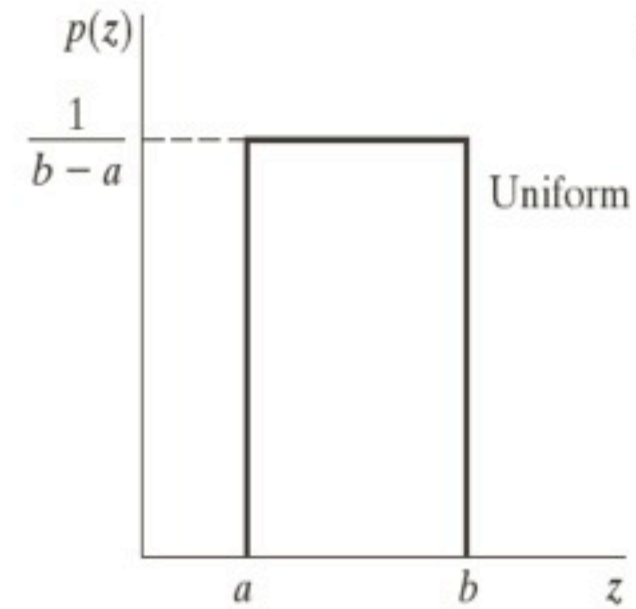


$$p(z) = \begin{cases} ae^{-az} & z \geq a \\ 0 & z < a \end{cases}$$

Mean: $\bar{z} = \dfrac{1}{a}$

Variance: $\sigma^2 = \dfrac{1}{a^2}$

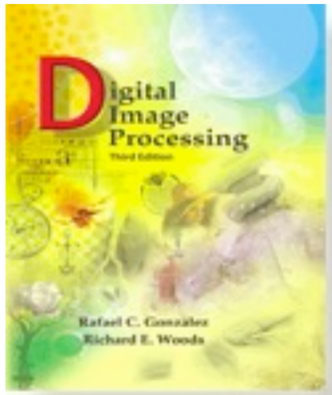
Uniform noise:



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

Mean: $\bar{z} = \dfrac{a+b}{2}$

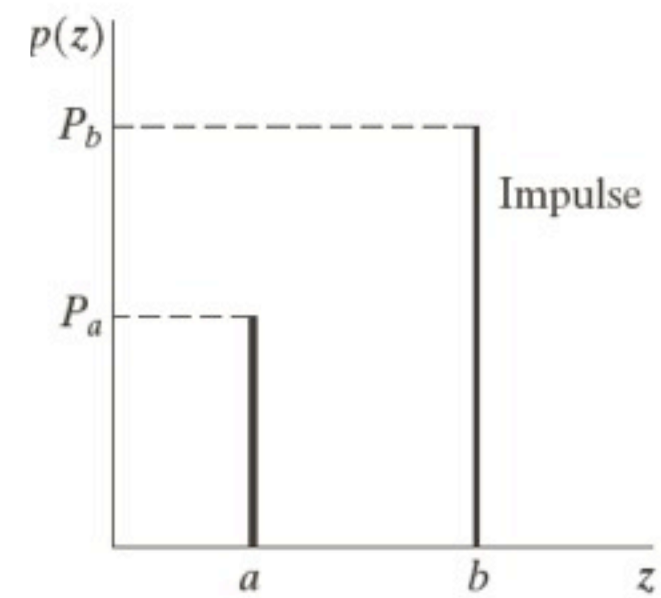
Variance: $\sigma^2 = \dfrac{(b-a)^2}{12}$



Impulse noise:

Bipolar impulse noise

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



assigns the values a (resp. b) with probability P_a (resp. P_b).

If one of the two probabilities is 0, we have a *unipolar* impulse noise. If the two probabilities are approximately equal: *salt and pepper* noise.

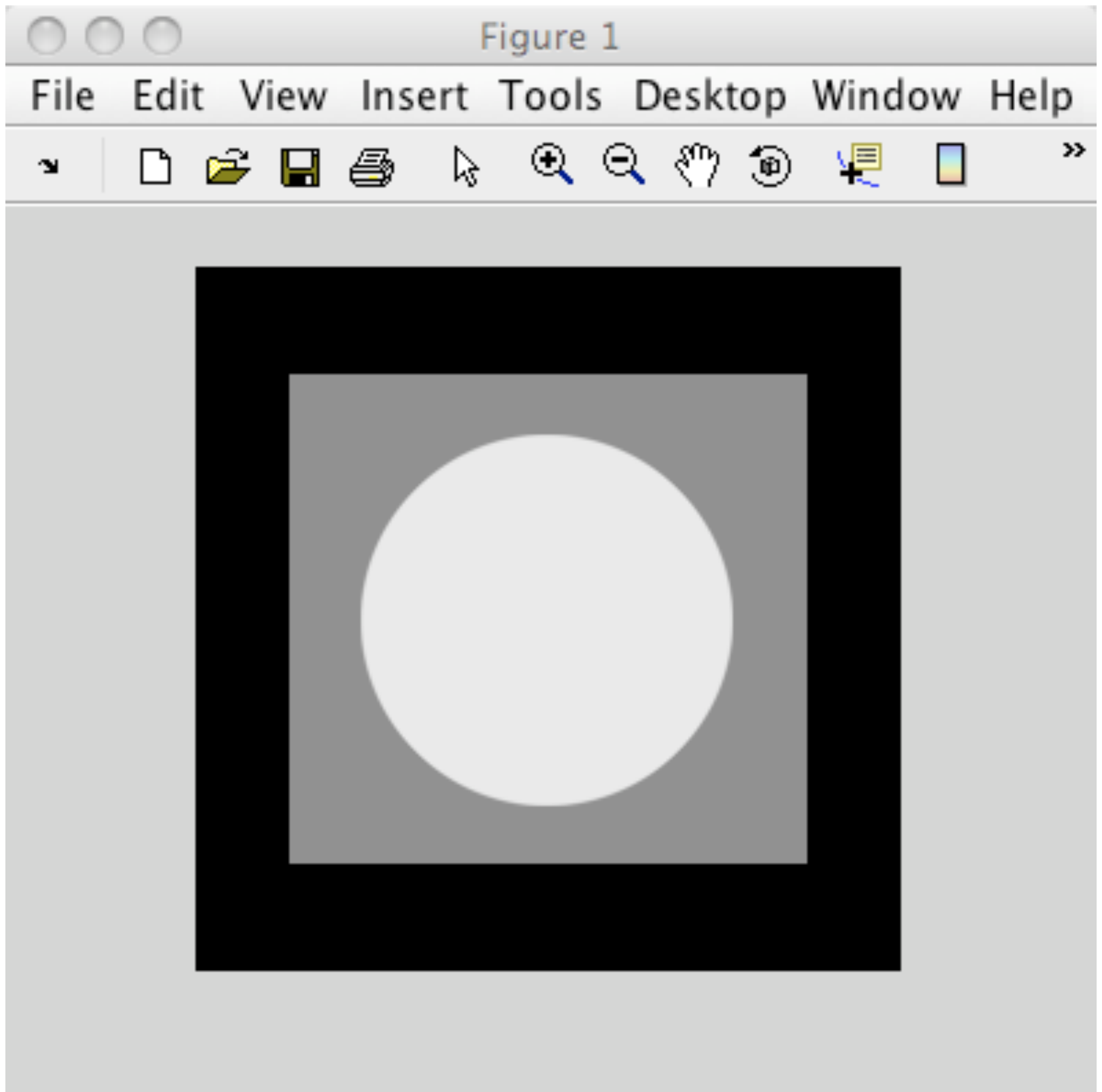
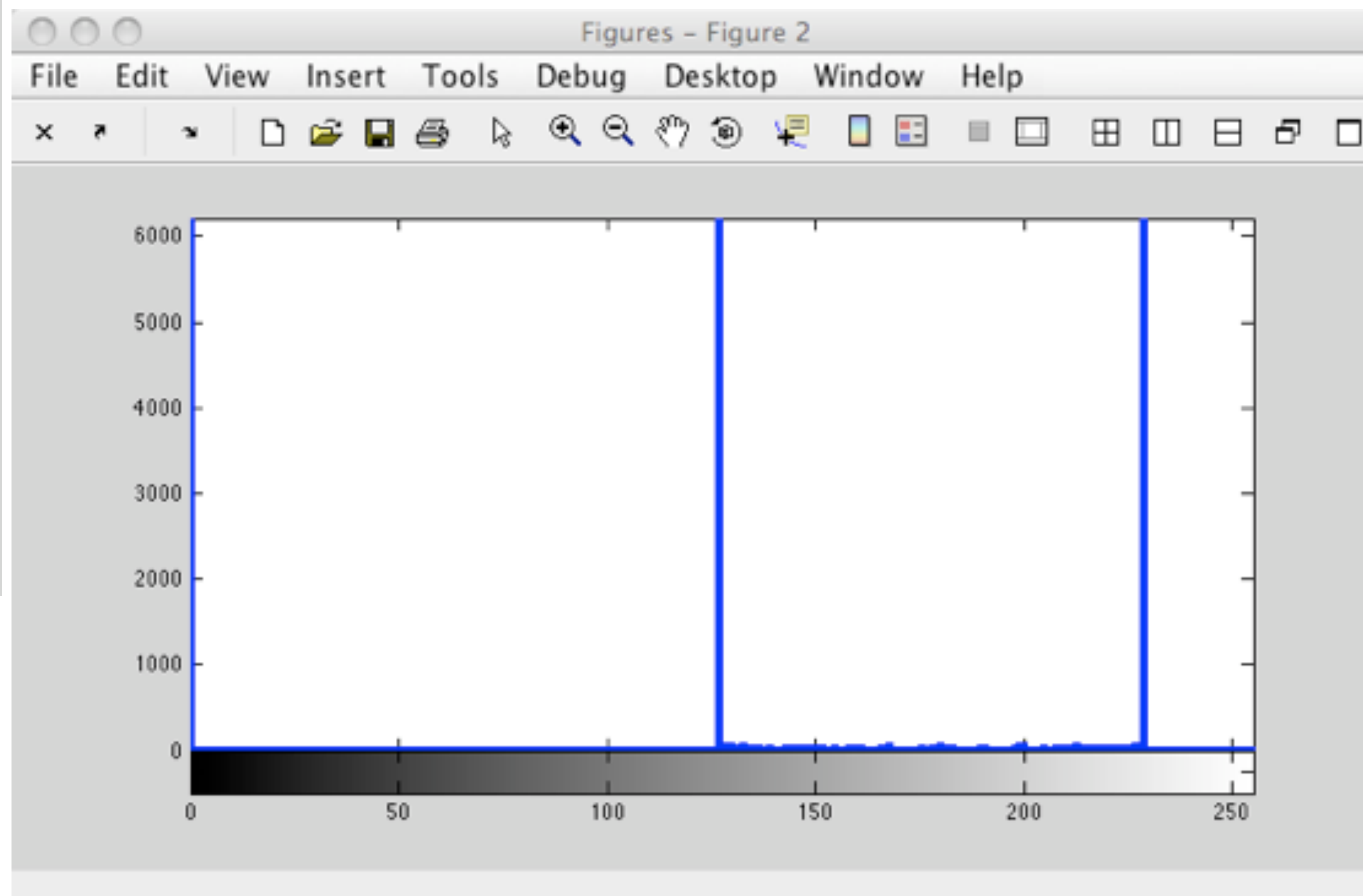


Figure 1
File Edit View Insert Tools Desktop Window Help


Figures – Figure 2
File Edit View Insert Tools Debug Desktop Window Help
6000
5000
4000
3000
2000
1000
0
0
50
100
150
200
250

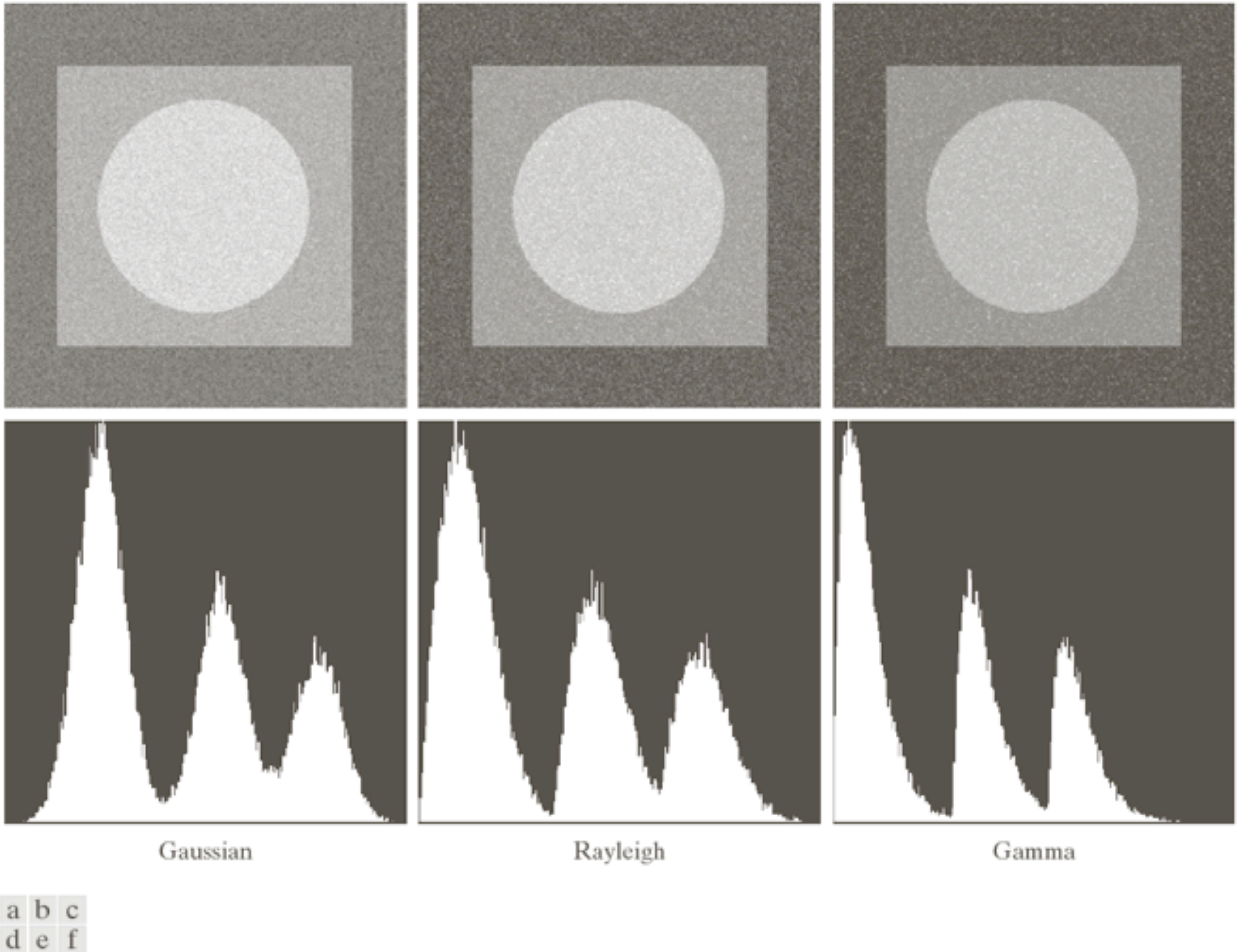


FIGURE 5.4 Images and histograms resulting from adding Gaussian, Rayleigh, and gamma noise to the image in Fig. 5.3.

Chapter 5

Image Restoration and Reconstruction

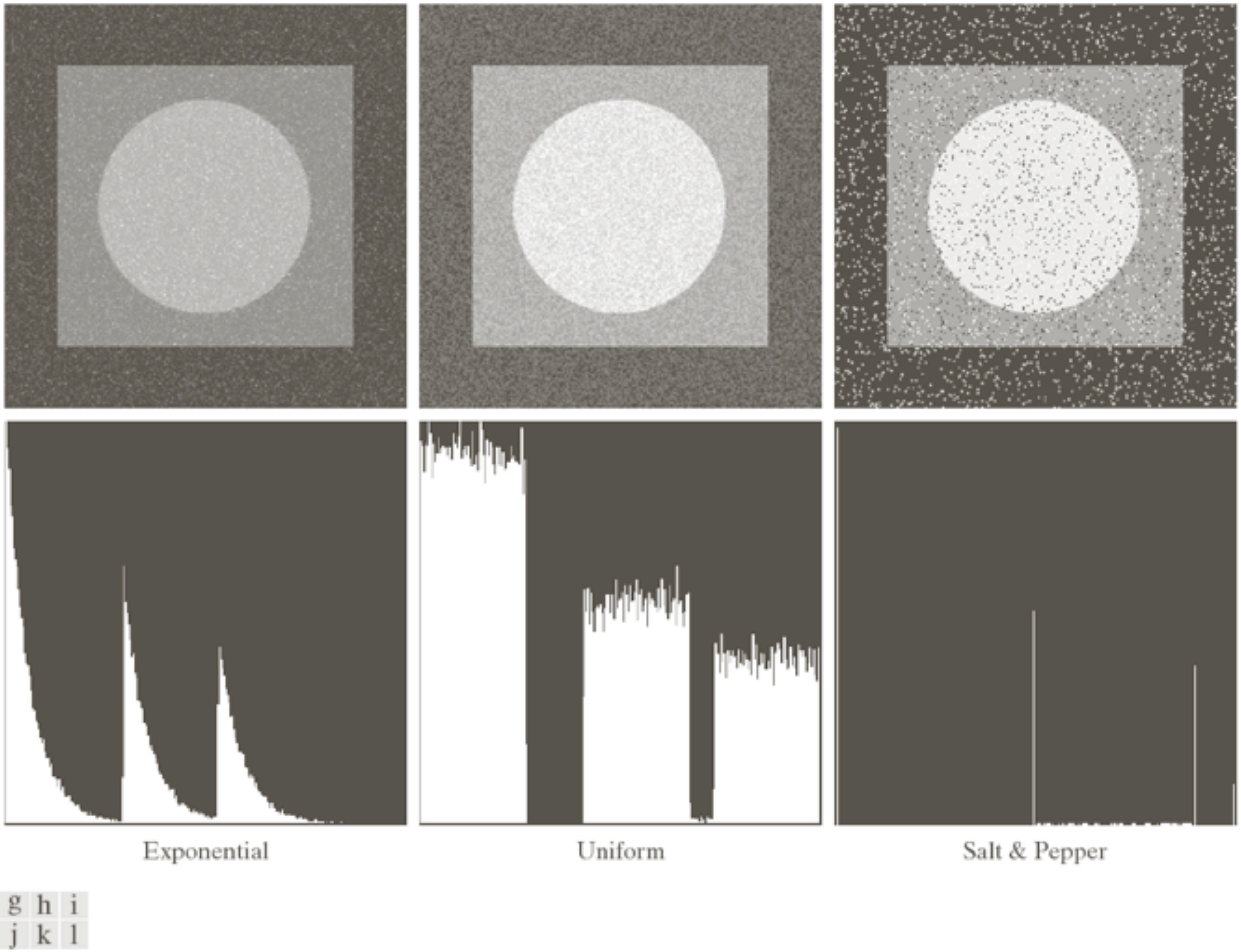
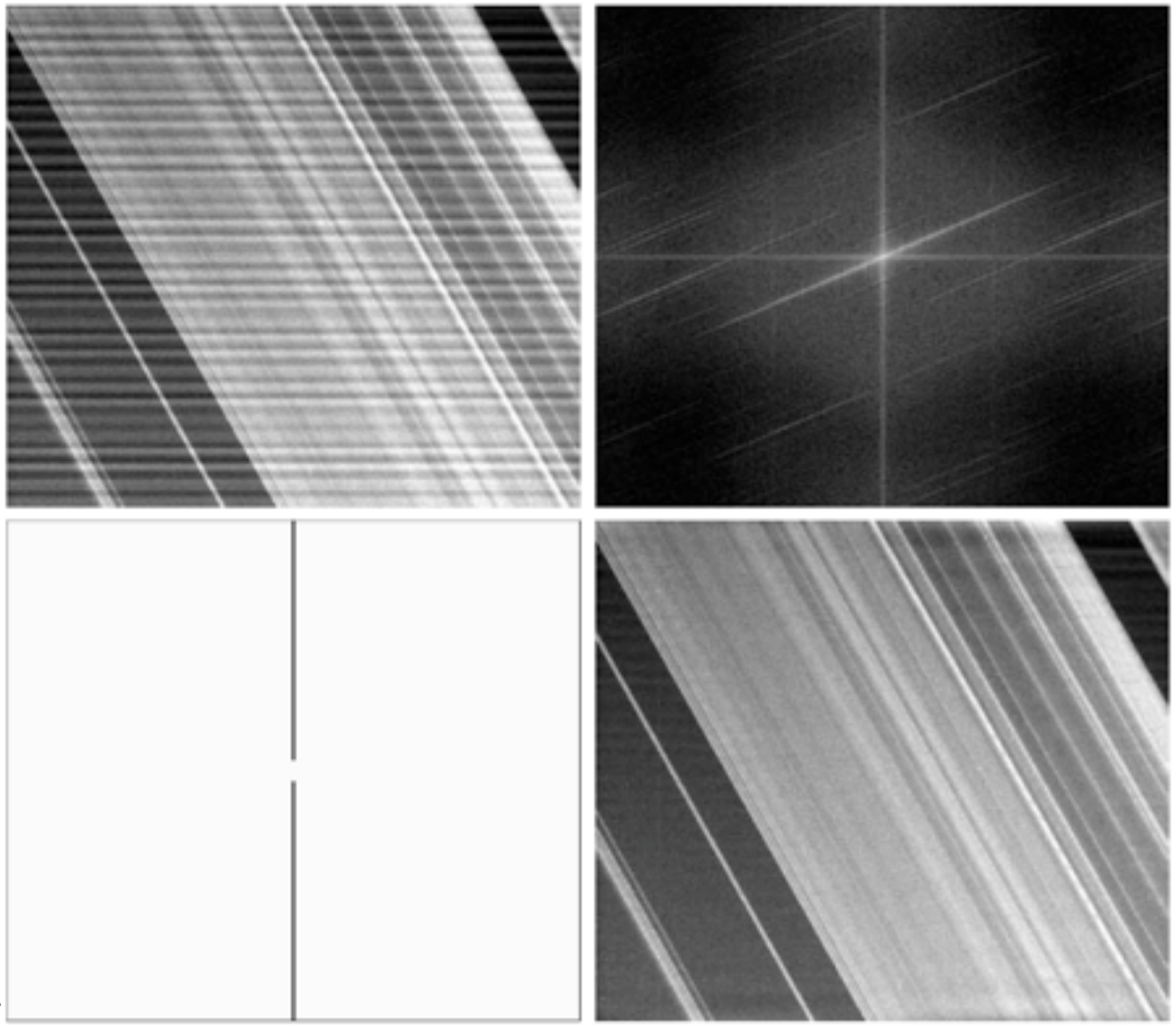


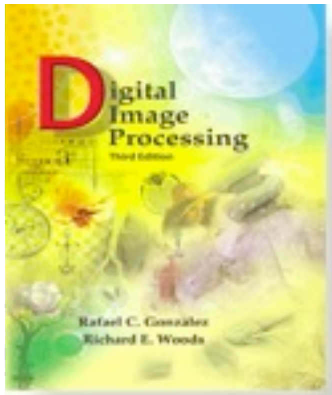
FIGURE 5.4 *(Continued)* Images and histograms resulting from adding exponential, uniform, and salt and pepper noise to the image in Fig. 5.3.

Periodic noise:

Typically arises from electrical interference during image acquisition.



Can be reduced in applying appropriate filters to the frequency domain.



How to estimate the noise parameters?

For periodic noise, we look at the Fourier spectrum for artifacts.
For spatial noise:

- sensor's specifications
- study of known images (phantoms)
- if only the image is available, look at small regions of "constant background intensity"

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.

We pick a strip of constant intensity and we look at its histogram.

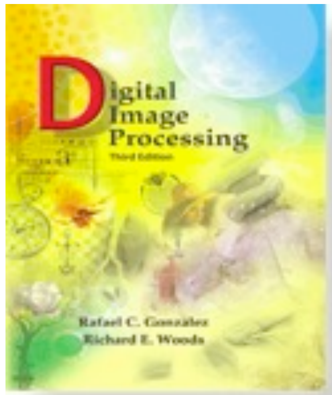
1. Compute the mean $$\bar{z}_S = \sum_{i=1}^{L-1} z_i p_S(z_i)$$

density function obtained normalizing the histogram

2. Compute the variance

$$\sigma_S^2 = \sum_{i=1}^{L-1} (z_i - \bar{z}_S)^2 p_S(z_i)$$

3. Choose the shape and evt. match the parameters *a*,*b*



Restoration in presence of noise only (no degradation)

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

Spatial filtering.

- Arithmetic mean filter

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

$\frac{1}{9} \times$

1	1	1
1	1	1
1	1	1

Simple average mask

- Geometric mean filter

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

Similar to the arithmetic mean filter, loses less detail.

- Harmonic mean filter:

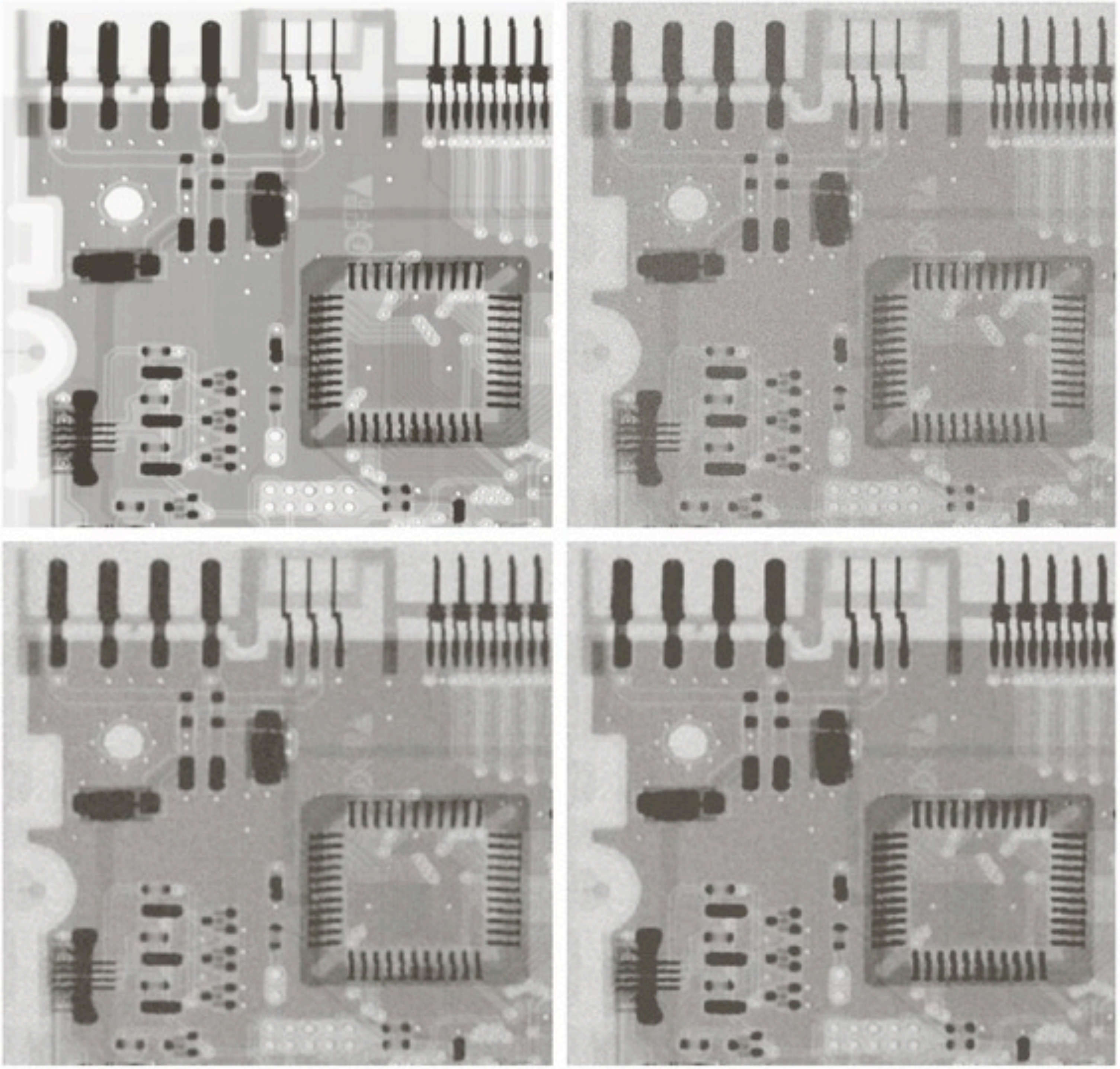
$$\hat{f}(x,y) = \frac{mn}{\sum_{(s,t)\in S_{xy}} \frac{1}{g(s,t)}}$$

Works well for Gaussian noise and salt noise, but not for pepper noise.

- Contra-harmonic mean filter of order *Q*:

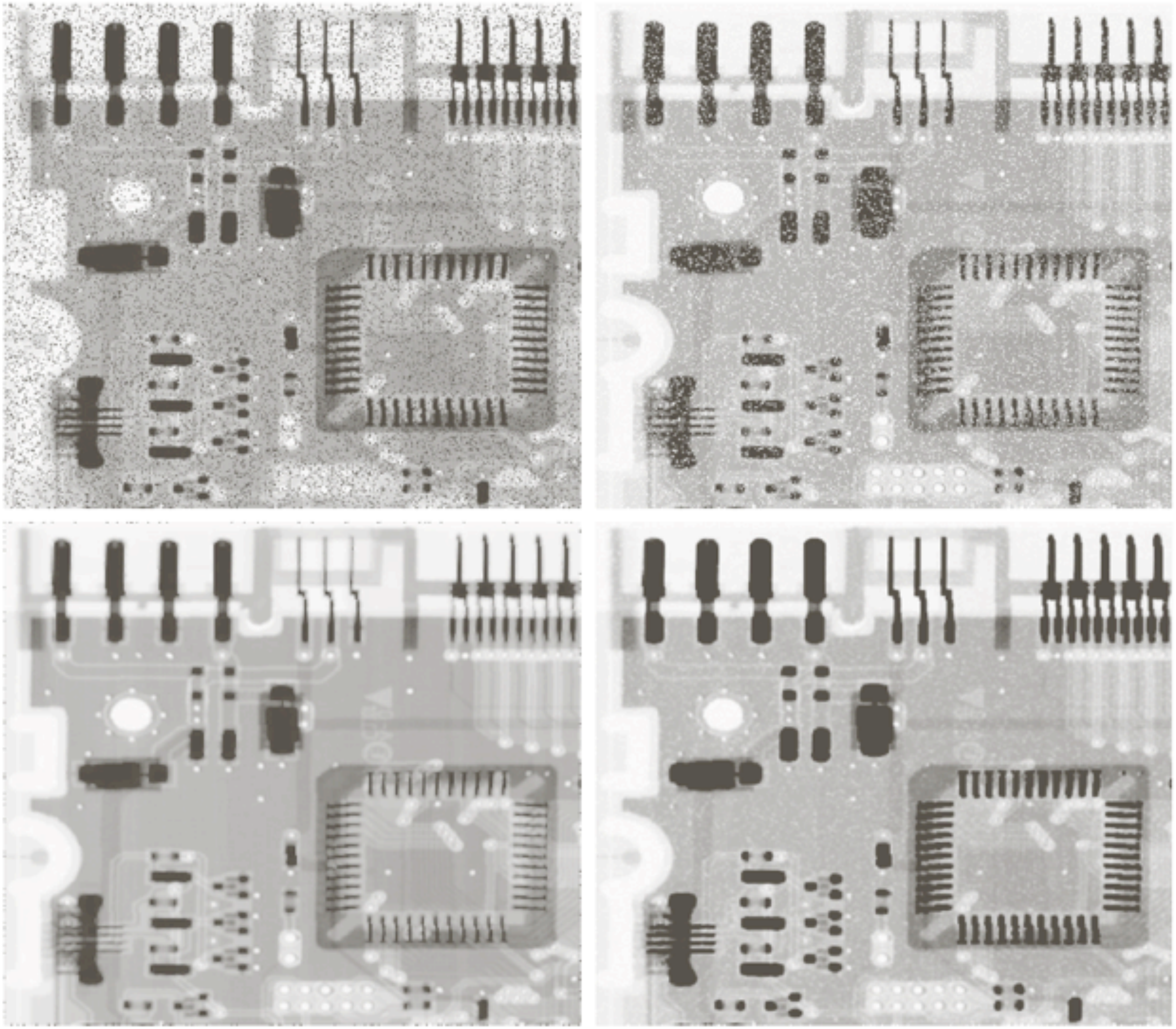
$$\hat{f}(x,y) = \frac{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q+1}}{\sum_{(s,t)\in S_{xy}} g(s,t)^{Q}}$$

Works well for salt and pepper noise:
Q>0 eliminates pepper noise,
Q<0 eliminates salt noise.
Q=-1 harmonic mean
Q=0 arithmetic mean



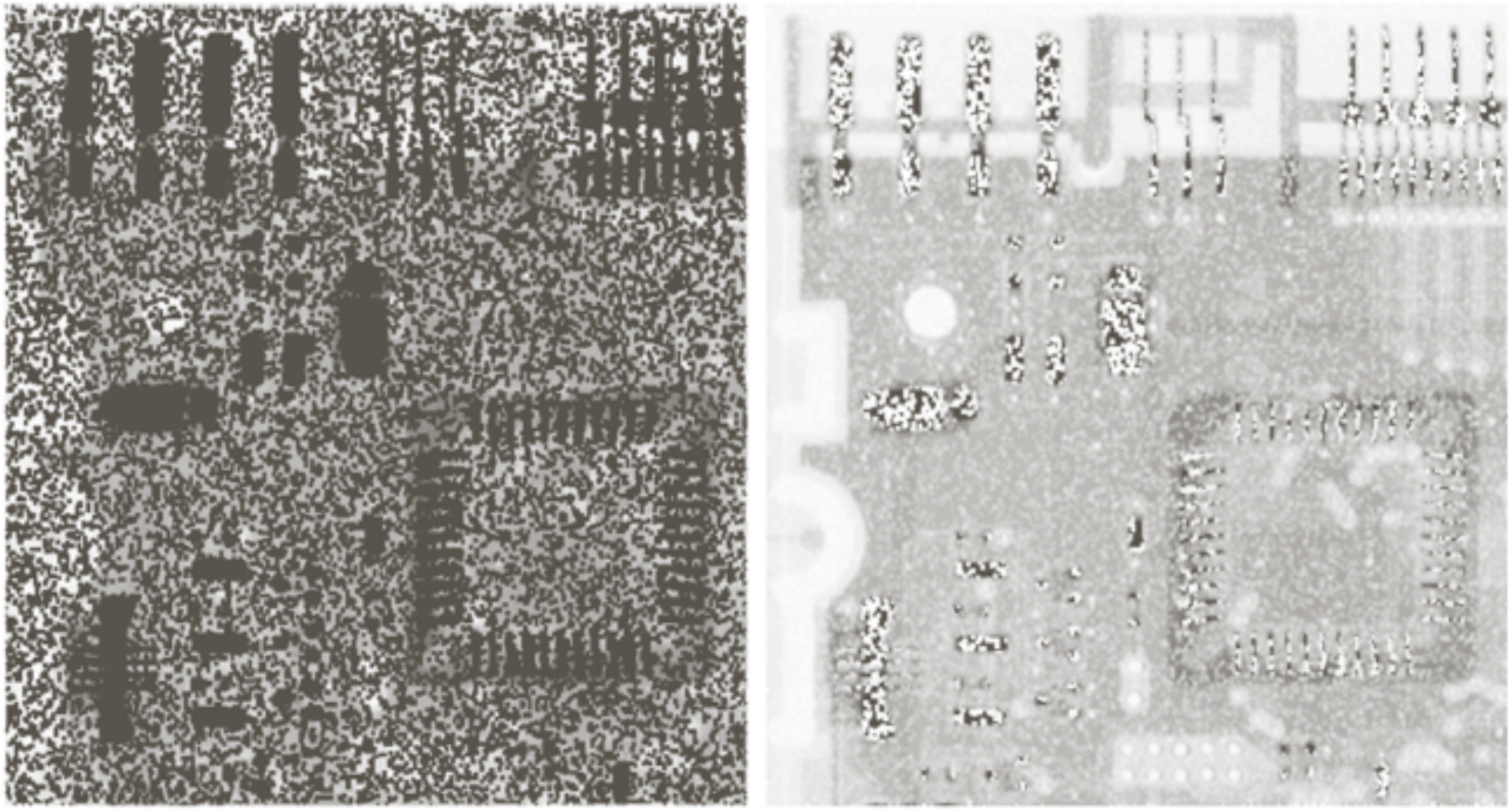
a b
c d

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.)



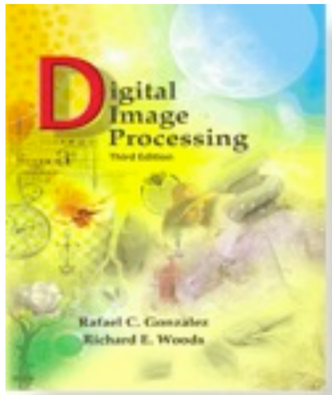
a b
c d

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 $Q = -1.5$.



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 $Q = 1.5$.



In general:

	Gaussian noise	Uniform noise	Salt and Pepper
Arithmetic mean	+	+	-
Geometric mean	++	++	-
Contra-harmonic	-	-	++ must know the sign

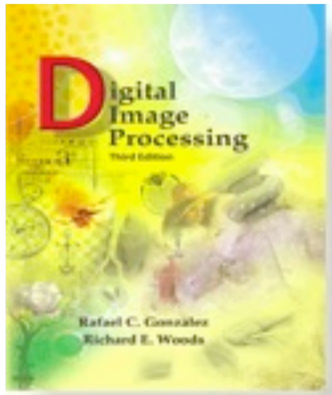
Order-statistic methods:

-median filter (50th percentile)
works well for S&P noise

-max, min (100th, 0th percentile),

max makes the image lighter, min makes the image darker
work well for pepper resp. salt noise

-midpoint filter: 1/2 (max + min)
works best for uniform/gaussian noise



-Alpha trimmed mean filter:

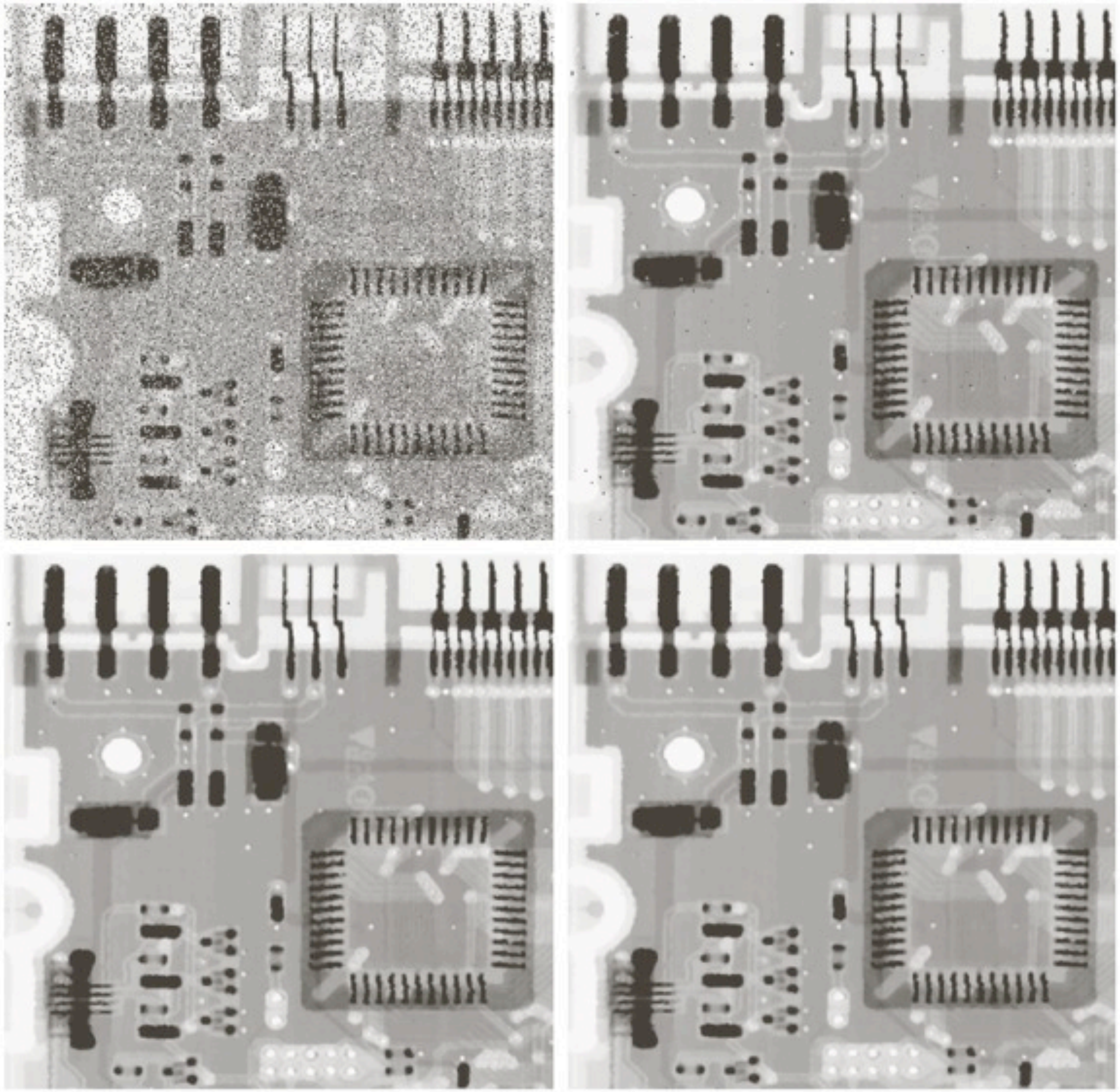
Remove *d/2* lowest values and *d/2* highest intensity values in S_{xy}, Thereafter apply the mean filter

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

represent the remaining pixels

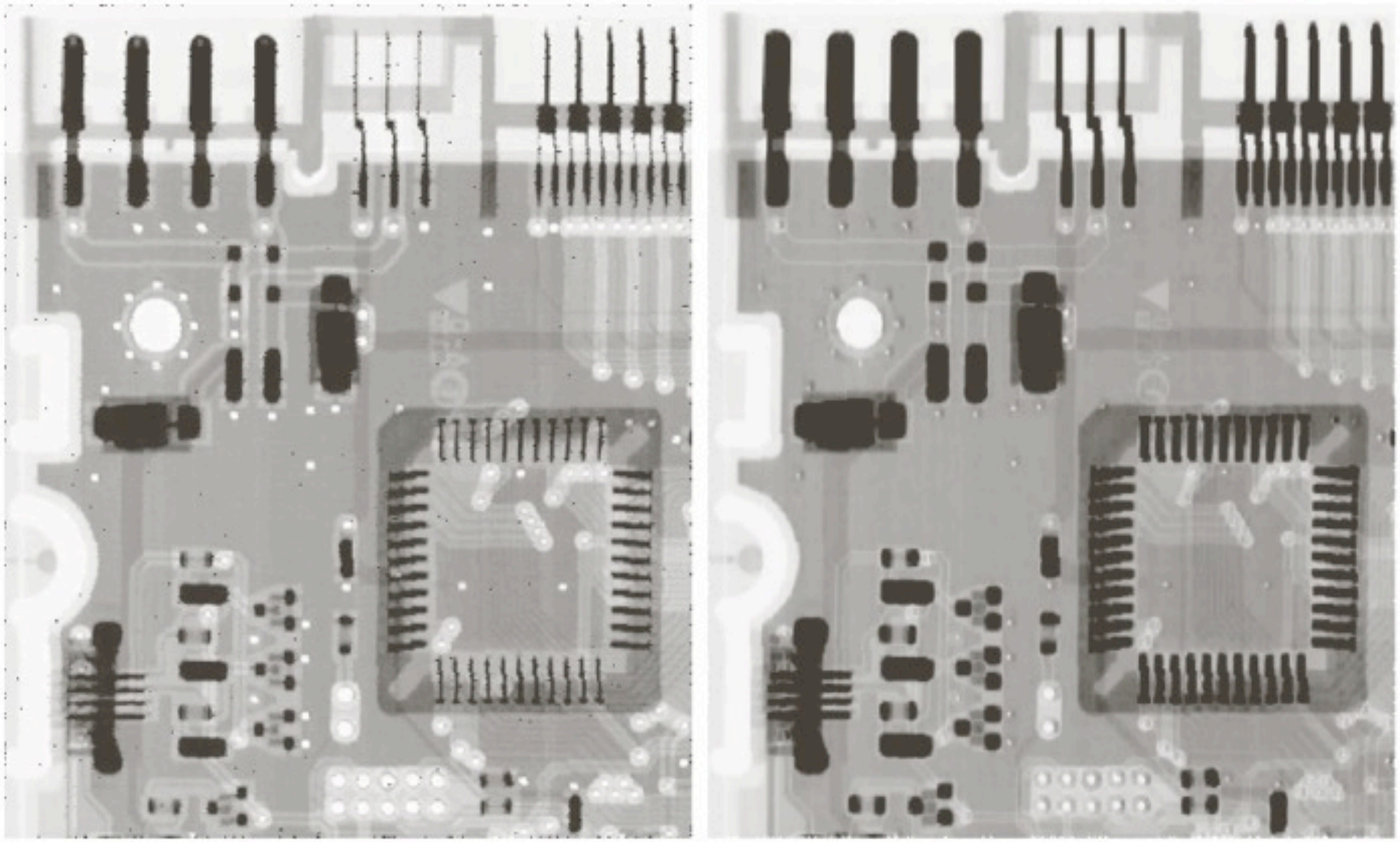
d=0: arithmetic mean filter
d=mn-1: median filter

Being a combination of arithmetic and median, it is expected to work well for mixed forms of noise



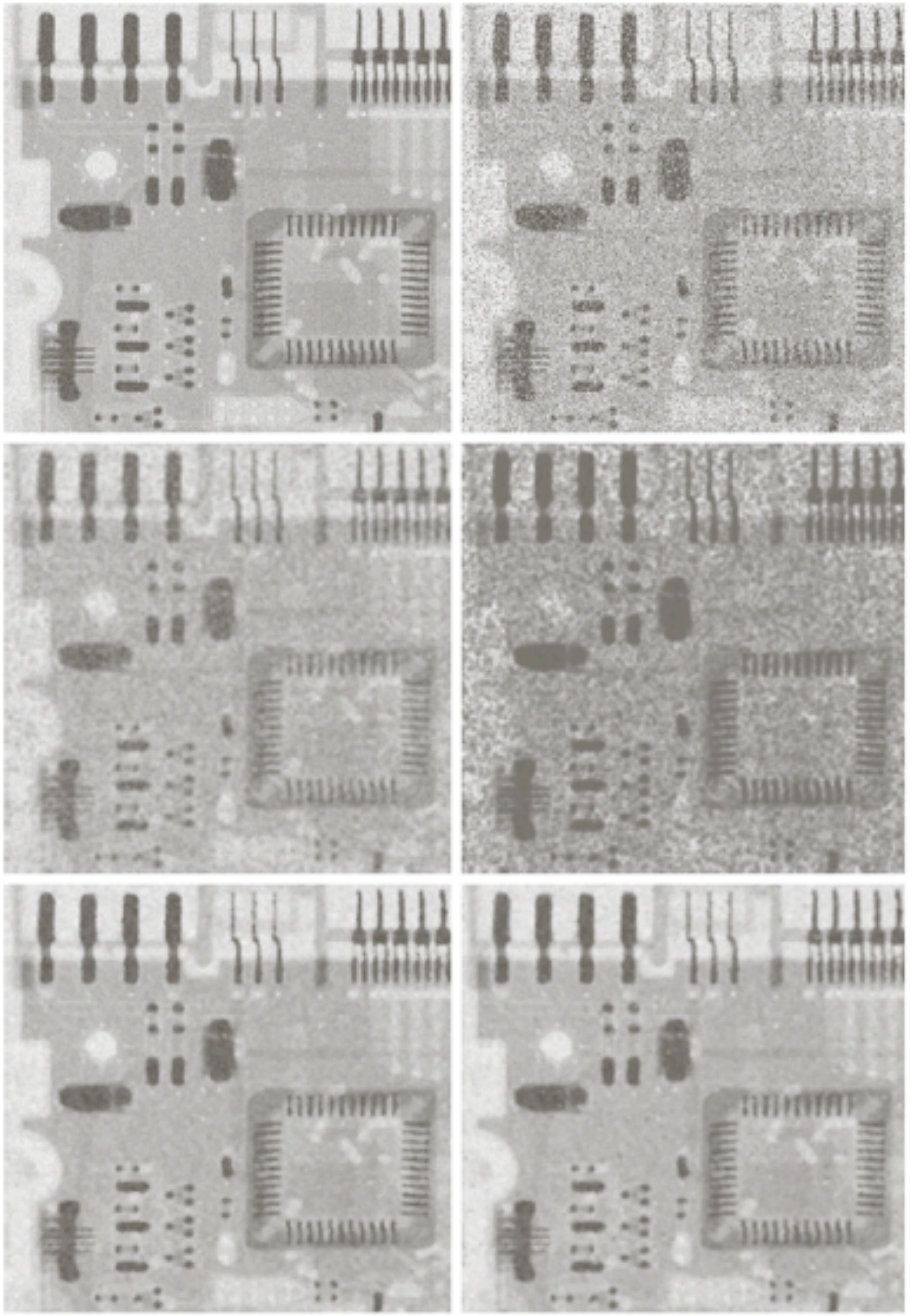
a b
c d

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



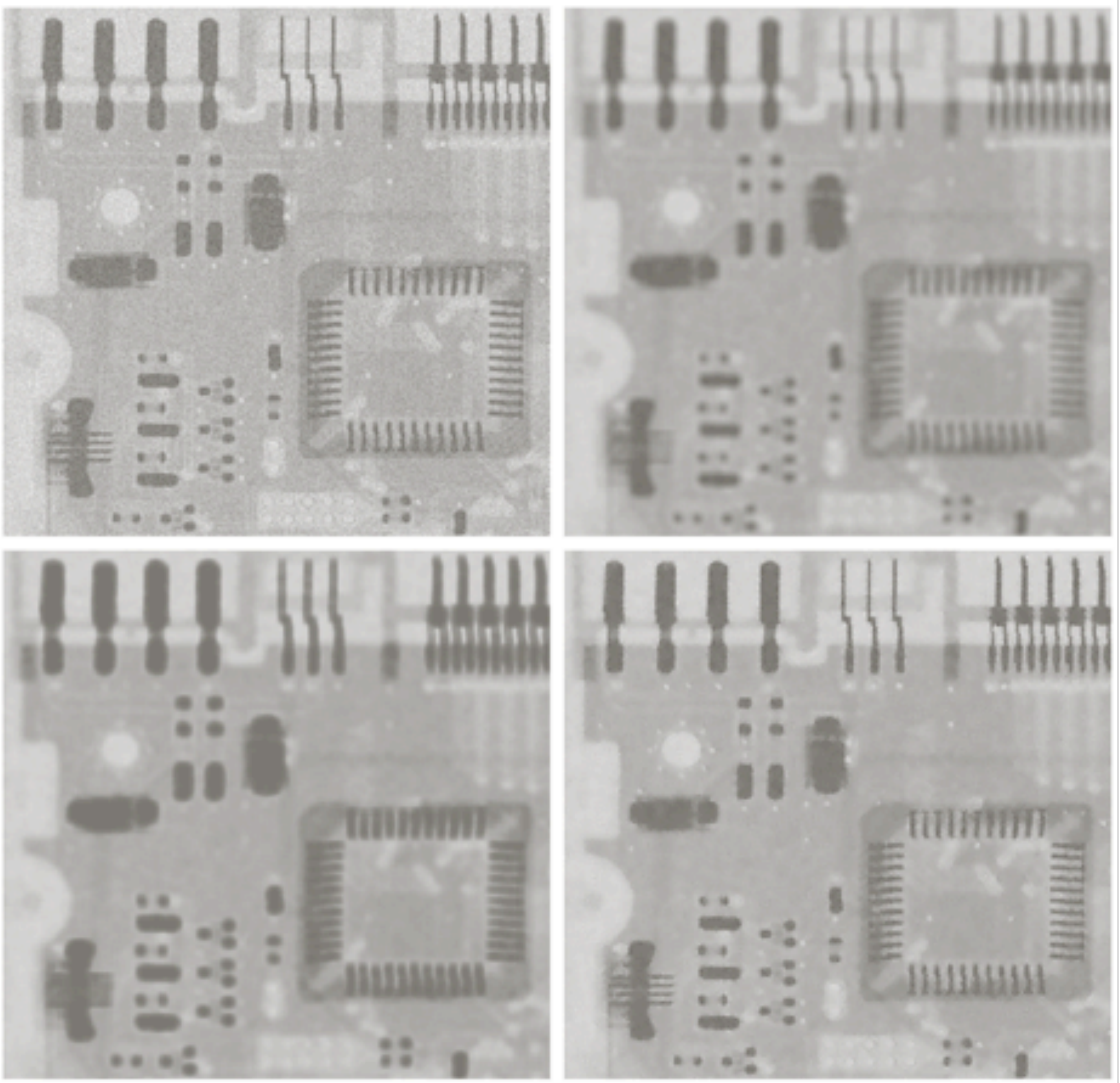
a b

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.



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 (b) filtered with a 5×5; (c) arithmetic mean filter; (d) geometric mean filter; (e) median filter; and (f) alpha-trimmed mean filter with $d = 5$.



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.

Color processing

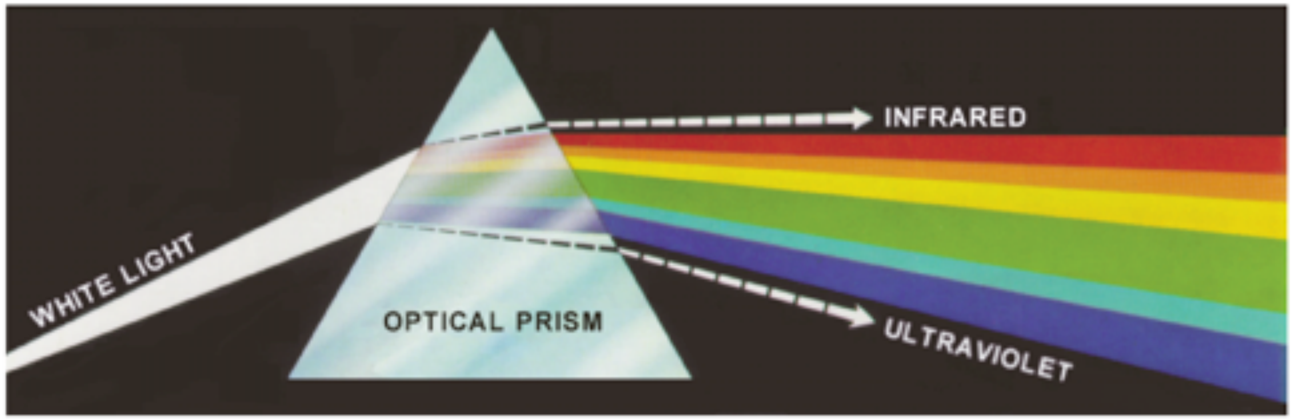


FIGURE 6.1 Color spectrum seen by passing white light through a prism. (Courtesy of the General Electric Co., Lamp Business Division.)

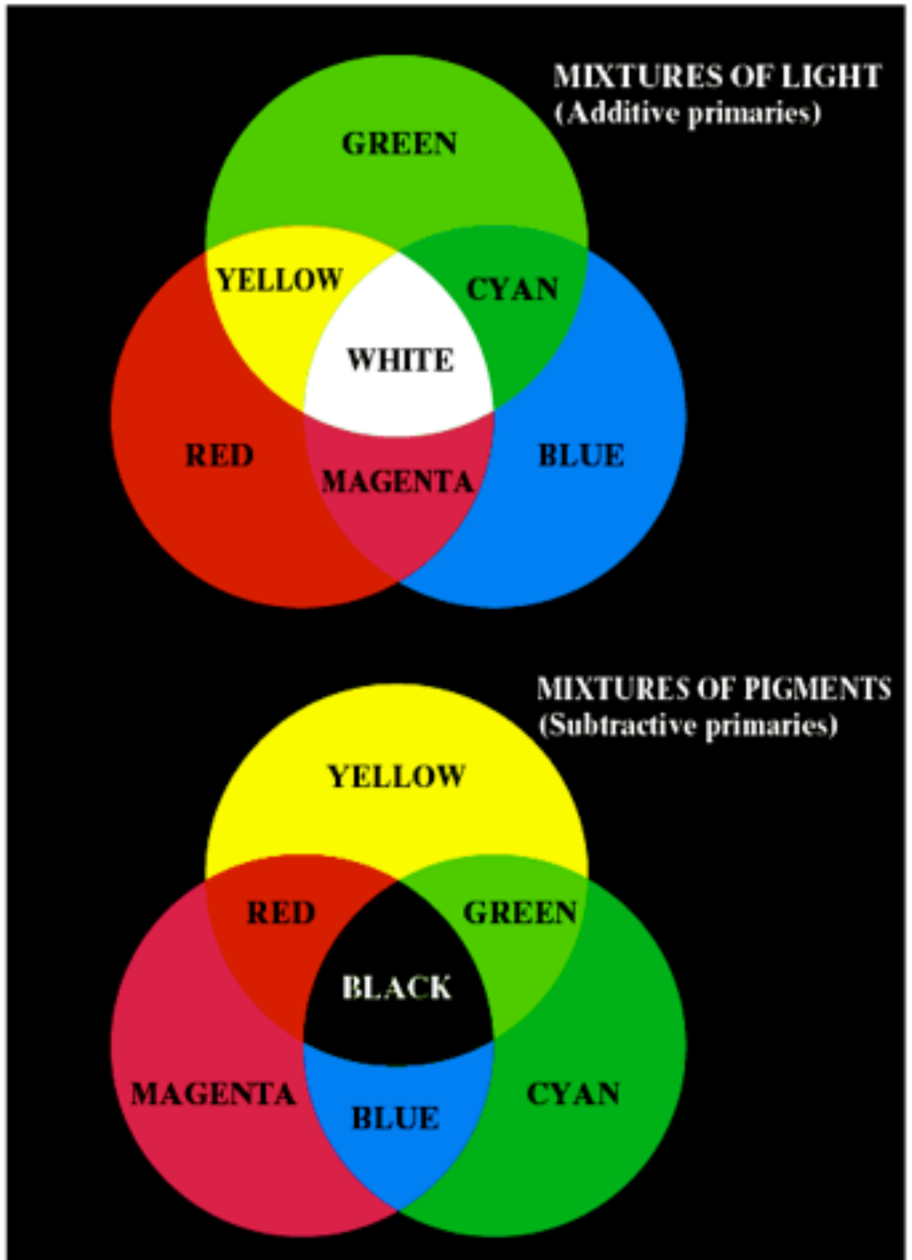
Colors can be obtained by
- adding
- subtracting

primary colors.

(primary colors are those that can be combined to obtain all the other colors)

Experiments on humans: high sensibility to Red (65%), Green (33%), Blue (2%)

Painters used yellow, red and blue as primary colors (pigments)

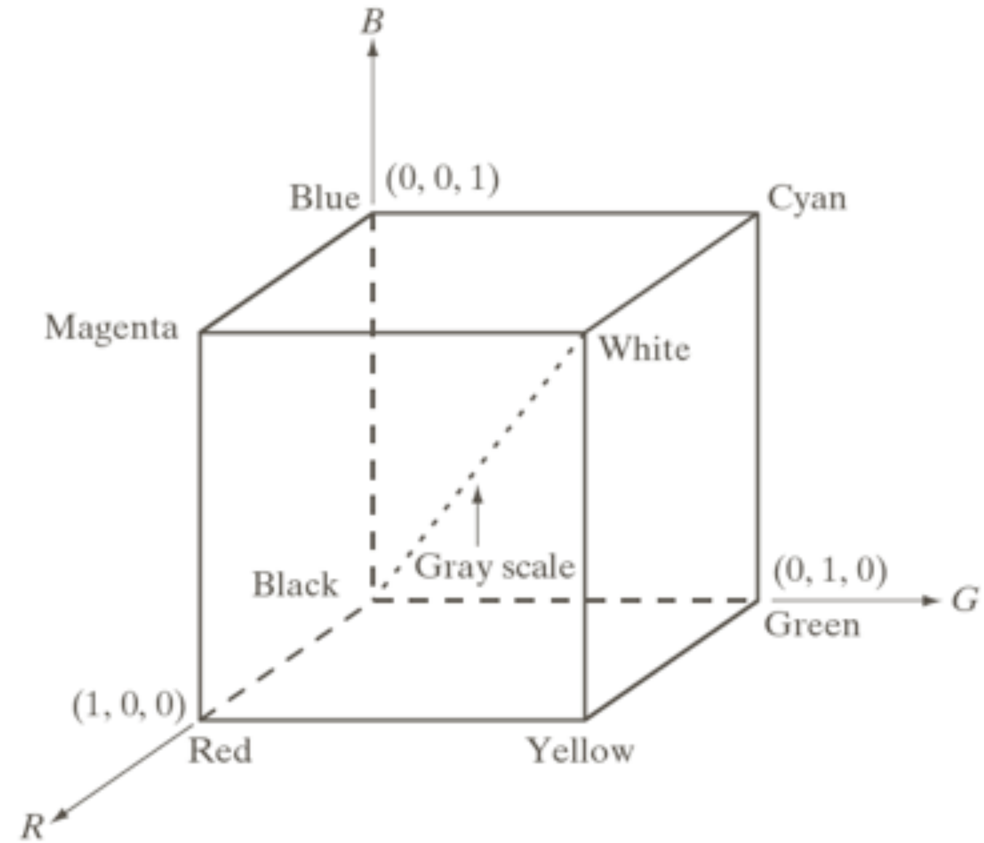


PRIMARY AND SECONDARY COLORS OF LIGHT AND PIGMENT

In images/computer graphics the RGB primaries are used.

An arbitrary color is obtained as an additive combination of
red = (1, 0, 0)
green = (0, 1, 0)
blue = (0, 0, 1)

and will be of the form (r,g,b), where the values range between 0 and 1.



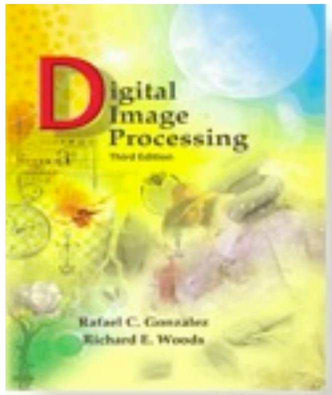
All the representable colors are in the “color cube”.

In images/computer graphics the RGB primaries are used.

An arbitrary color is obtained as an additive combination of
red = (1, 0, 0)
green = (0, 1, 0)
blue = (0, 0, 1)

and will be of the form (r,g,b), where the values range between 0 and 1.

All the representable colors are in the “color cube”.



A RGB image consists of three components (R,G,B).

Assume that each component is represented using 8-bits (a value between 0 and 255=2^8-1).
In this system (full color representation), there are (2^8)^3= 16.777.216 possible colors.

Not all devices can represent faithfully all these colors. It is assumed that most devices can represent faithfully 256 colors (*safe RGB colors)*.

These are obtained as combinations of the values
0, 51, 102, 153, 204, 255
this gives 6^3 = 216 possible color values. In addition, we have the *gray colors*.

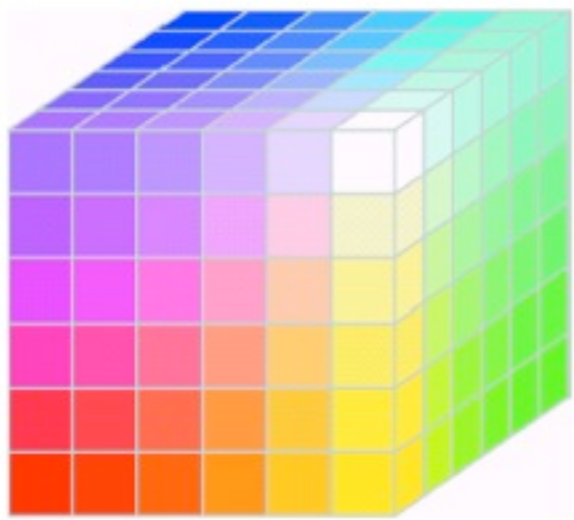
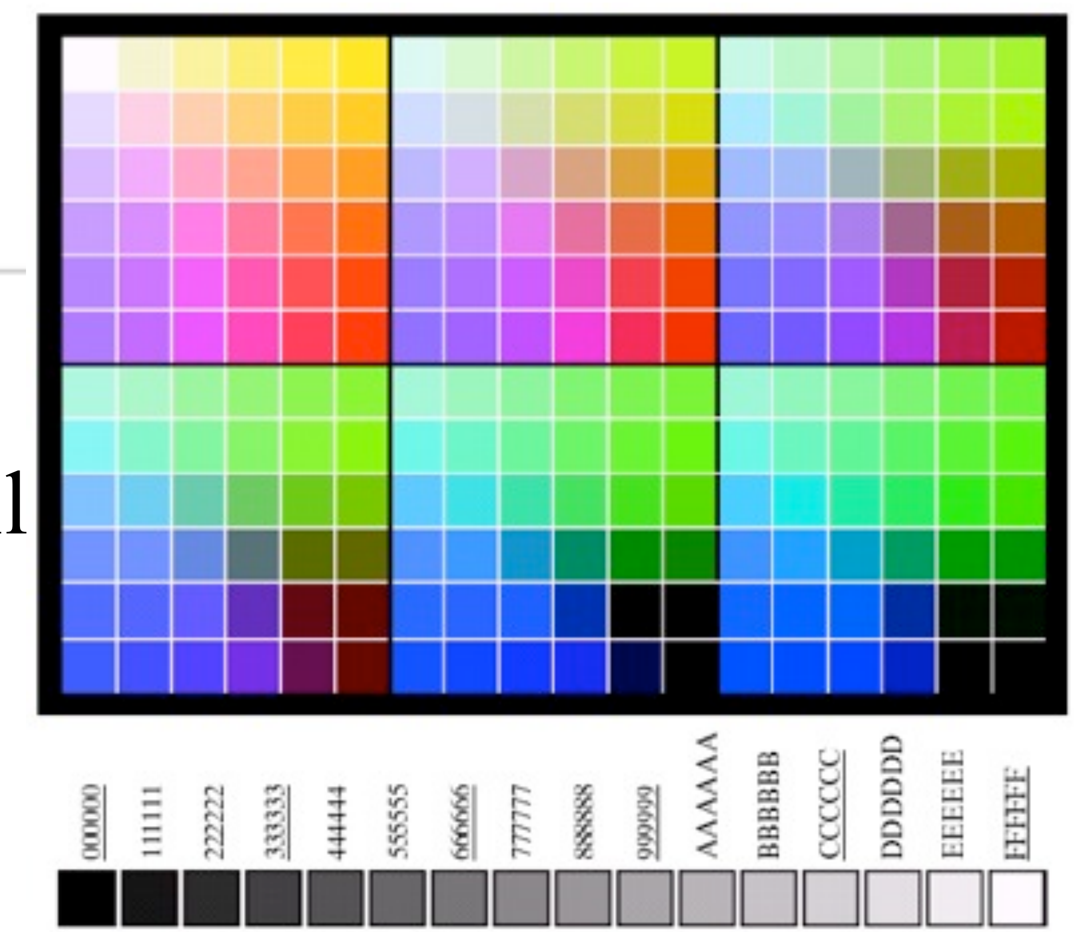


FIGURE 6.11 The RGB safe-color cube.

Number System	Color Equivalents					
Hex	00	33	66	99	CC	FF
Decimal	0	51	102	153	204	255

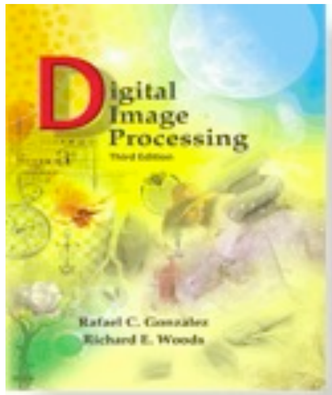
TABLE 6.1 Valid values of each RGB component in a safe color.



a
b

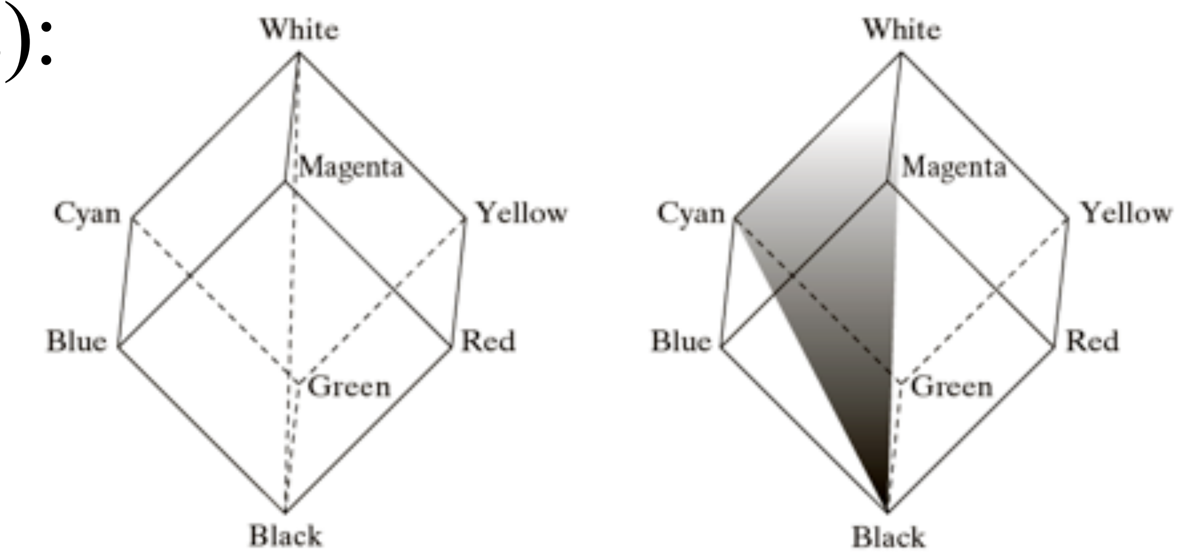
FIGURE 6.10 (a) The 216 safe RGB colors. (b) All the grays in the 256-color RGB system (grays that are part of the safe color group are shown underlined).

We have a cube with faces of size 36. Differently from the full color system, only the faces of the cube are representable, plus some gray tones.



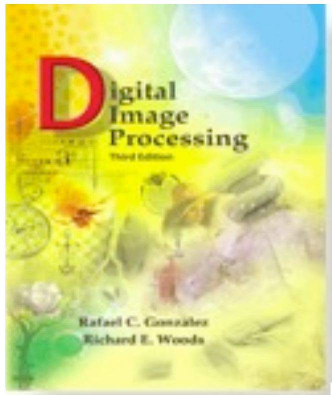
CMY system (primary for pigments):

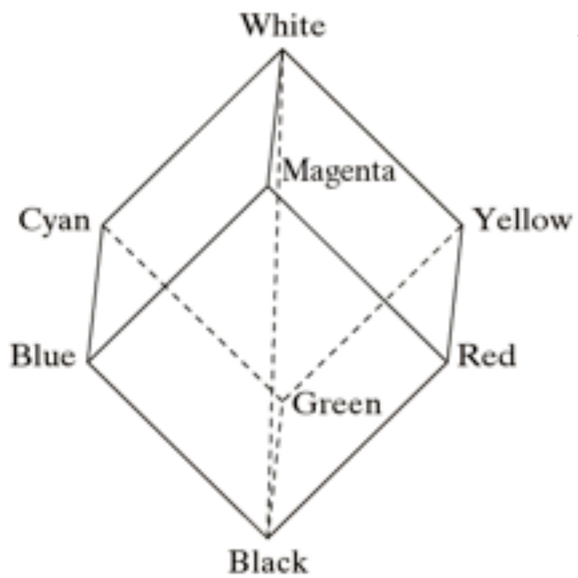
(subtractive system)

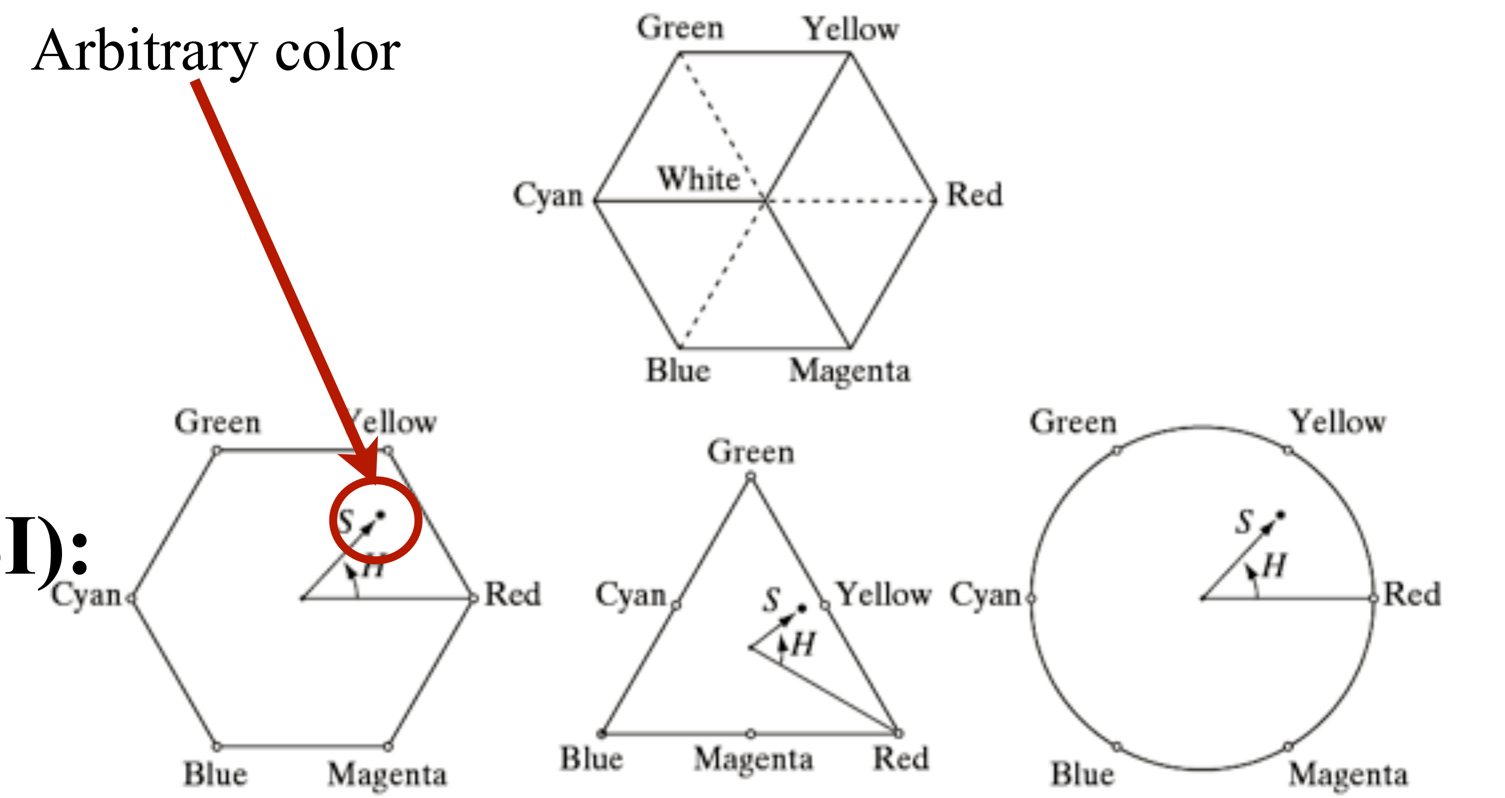


C		1-R
M	=	1-G
Y		1-B

The black (obtained in the CMY system), can sometimes be of bad quality in print. For this reason, it is often added as a primary color (CMYK system).







a
b c d

FIGURE 6.13 Hue and saturation in the HSI color model. The dot is an arbitrary color point. The angle from the red axis gives the hue, and the length of the vector is the saturation. The intensity of all colors in any of these planes is given by the position of the plane on the vertical intensity axis.

Hue Saturation Intensity (HSI):

Take the color cube (primary colors on the corners) and rotate it so that seen from above the white is in the center.

Hue: angle between the line joining to the red and to the color point
Saturation: length of the vector between the center and the color point.
Intensity: elevation

This is equivalent to taking cutting planes perpendicular to the gray scale.

(R,G,B)-> (H,S,I):

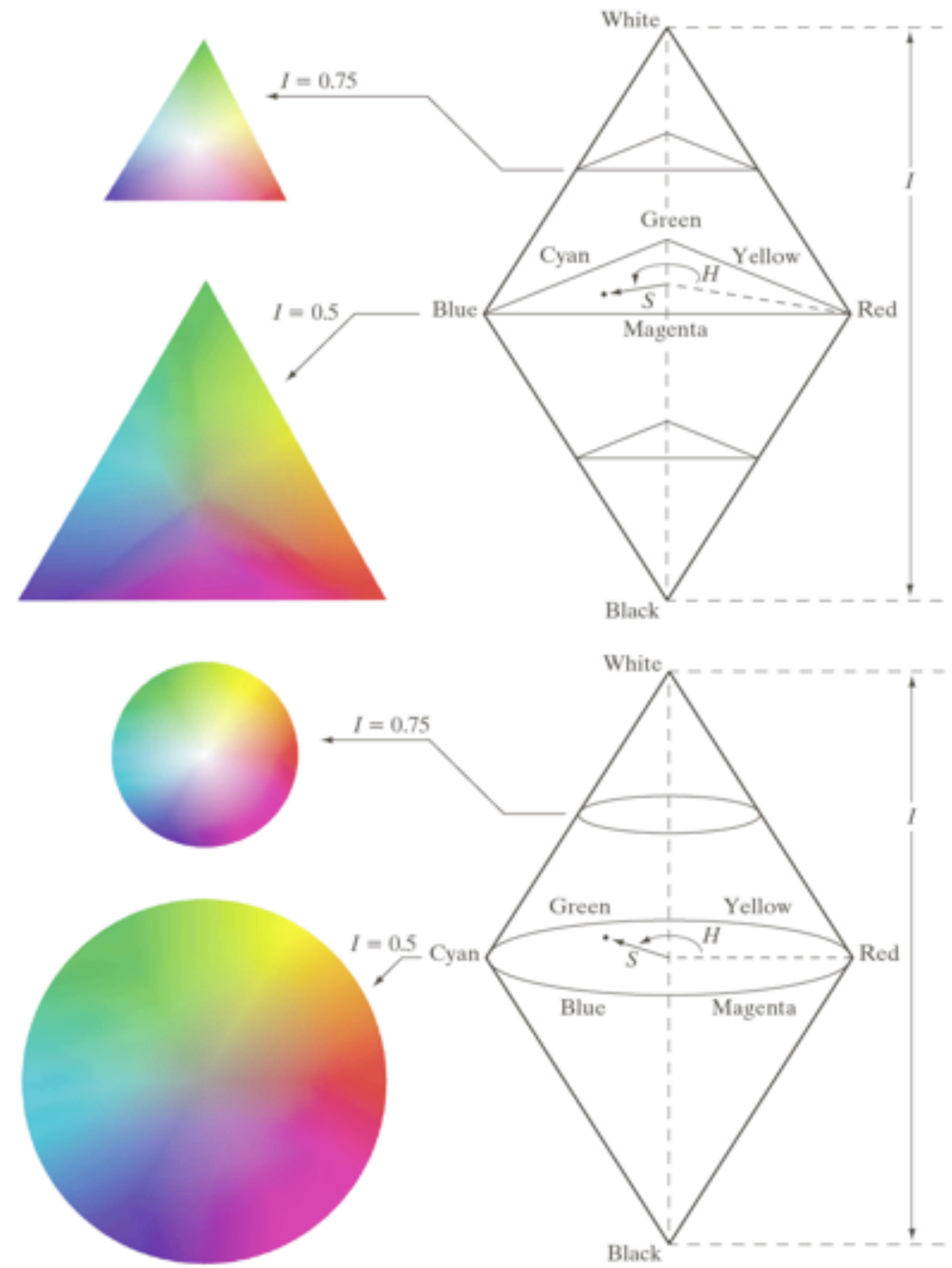
$$H = \begin{cases} \theta & \text{if } B \leq G \\ 360 - \theta & \text{otherwise} \end{cases}$$

where

$$\theta = \cos^{-1}\left(\frac{\frac{1}{2}[(R-G)+(R-B)]}{[(R-G)^2_{(}R-B)(G-B)]^{1/2}}\right)$$

$$S = 1 - \frac{3}{R+G+B}\min\{R, G, B\},$$

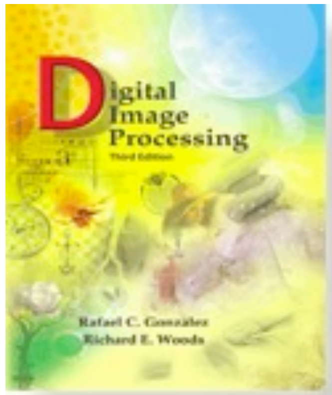
$$I = \frac{1}{3}(R+G+B)$$



HSI -> RGB: depends on sector, RG, GB, BR

RG (0≤H<120):

$$B = I(1 - S) \qquad R = I(1 + \frac{S \cos H}{\cos(60 - H)}, \qquad G = 3I - (R + B)$$



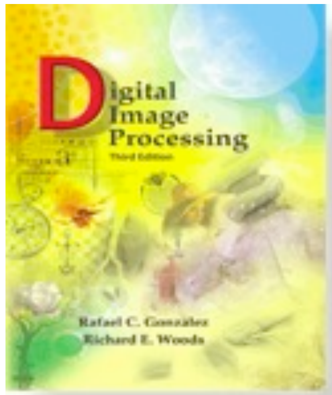
HSI -> RGB: depends on sector, RG, GB, BR

RG (0≤H<120):

$$B = I(1 - S) \qquad R = I\left(1 + \frac{S\cos H}{\cos(60 - H)}, \qquad G = 3I - (R + B)\right.$$

GB (120≤H<240):

$$H = H - 120, \qquad R = I(1 - S), \qquad G = I\left(1 + \frac{S\cos H}{\cos(60 - H)}\right), \qquad B = 3I - (R + G)$$



HSI -> RGB: depends on sector, RG, GB, BR

RG (0≤H<120):

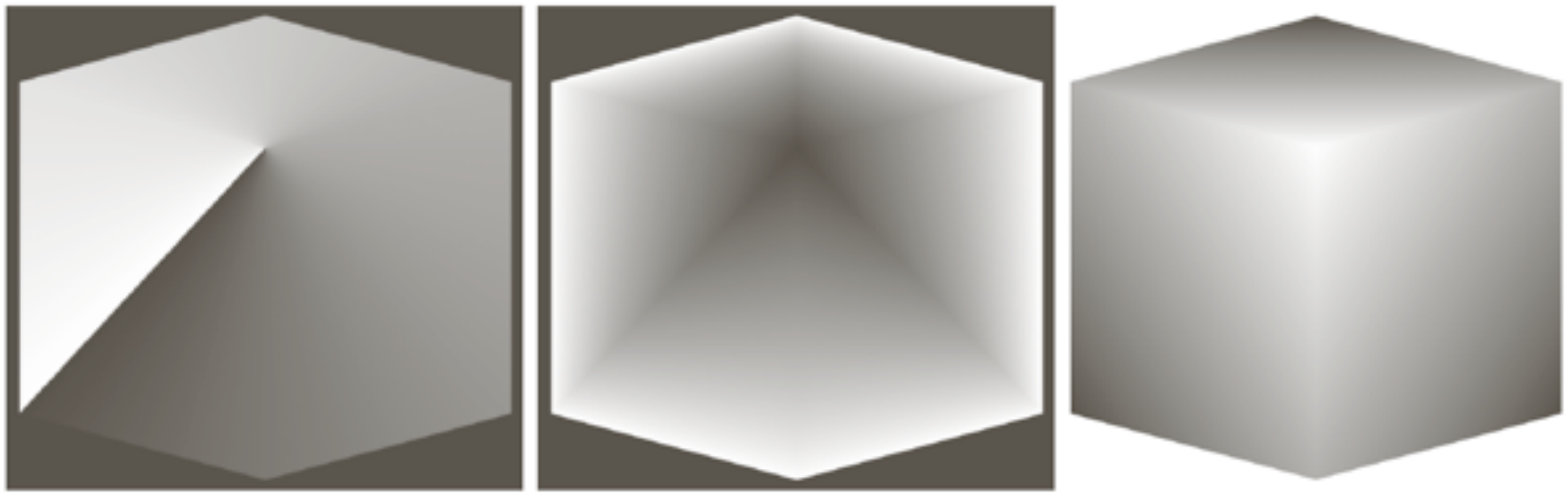
$$B = I(1 - S) \qquad R = I\left(1 + \frac{S\cos H}{\cos(60 - H)}\right), \qquad G = 3I - (R + B)$$

GB (120≤H<240):

$$H = H - 120, \qquad R = I(1 - S), \qquad G = I\left(1 + \frac{S\cos H}{\cos(60 - H)}\right), \qquad B = 3I - (R + G)$$

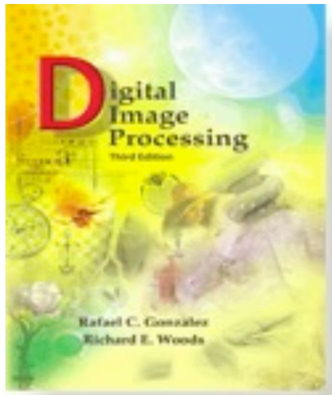
BR (240≤H≤360):

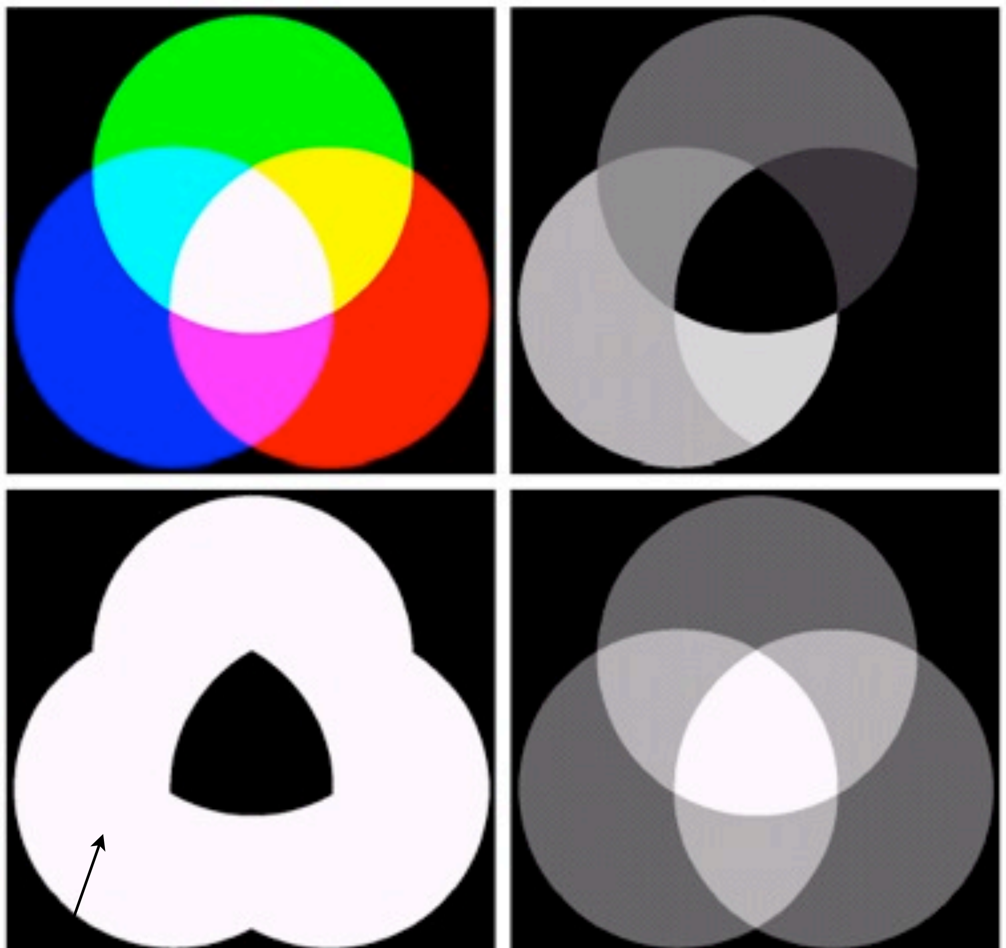
$$H = H - 240, \qquad G = I(1 - S), \qquad B = I\left(1 + \frac{S\cos H}{\cos(60 - H)}\right), \qquad R = 3I - (G + B)$$



a b c

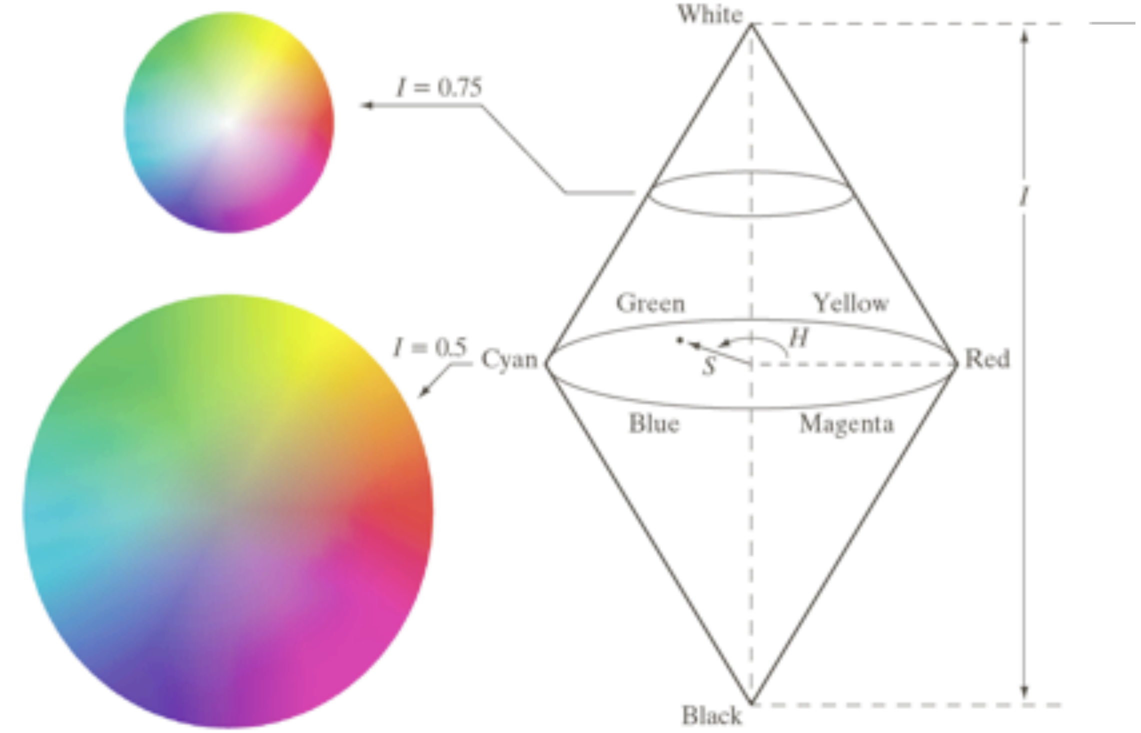
FIGURE 6.15 HSI components of the image in Fig. 6.8. (a) Hue, (b) saturation, and (c) intensity images.

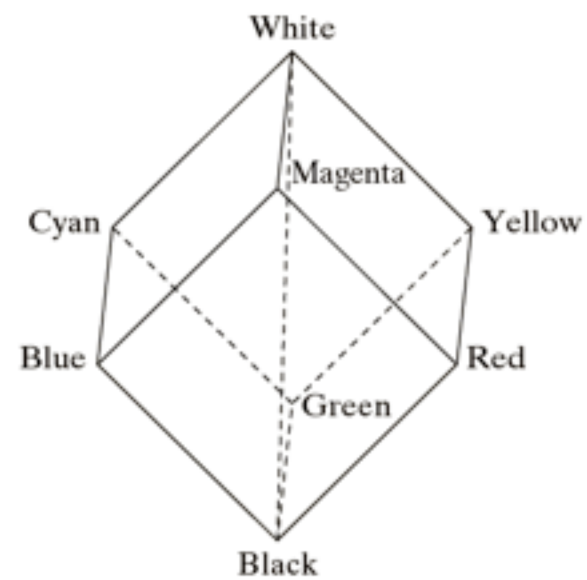




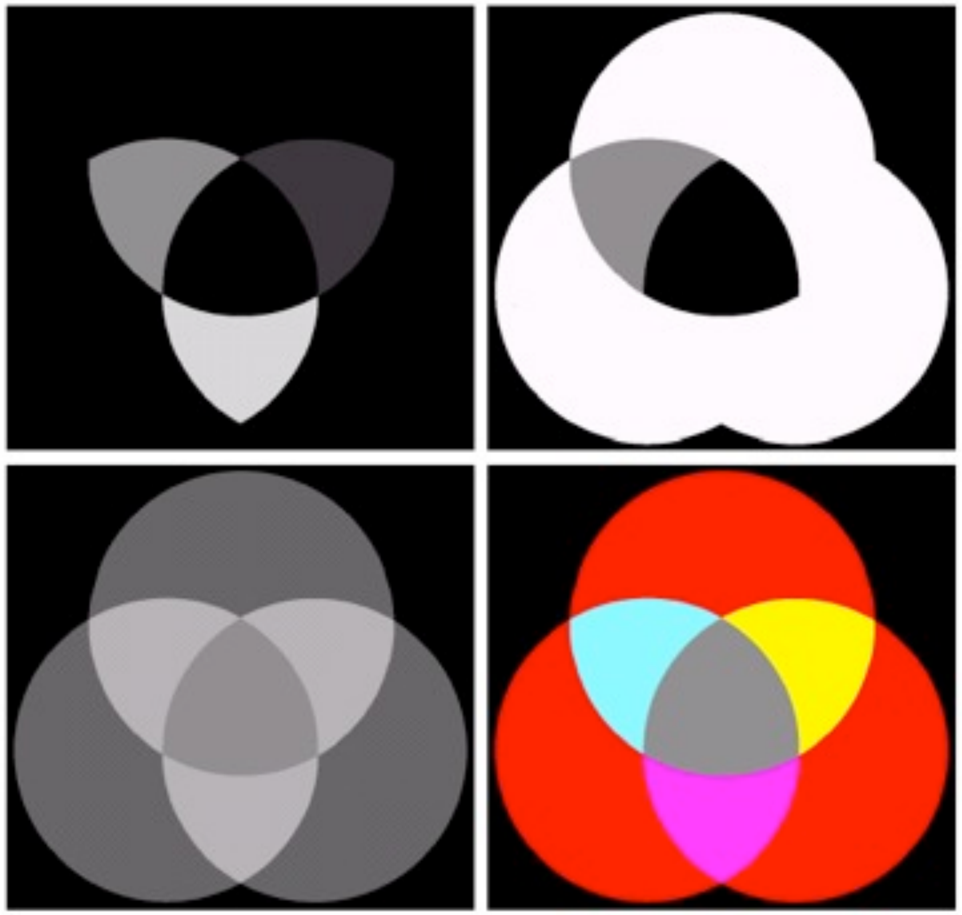
a b
c d

FIGURE 6.16 (a) RGB image and the components of its corresponding HSI image: (b) hue, (c) saturation, and (d) intensity.





These colors are on the outer border of the cone, hence have maximal saturation



a b
c d

FIGURE 6.17 (a)–(c) Modified HSI component images. (d) Resulting RGB image. (See Fig. 6.16 for the original HSI images.)