Review of SGN 12007– Basic Course in Image and Video Processing

Fall 2017

Course Outline

Chapter 1: Introduction to Digital Image Processing
Chapter 2: Digital Image Fundamentals
Chapter 3: Image Enhancement in the Spatial Domain
Chapter 4: Image Enhancement in the Frequency Domain
Chapter 5: Image Restoration
Chapter 6: Color Image Processing
Chapter 7: Basics of Digital Video

Chapter 1: Introduction to Digital Image Processing

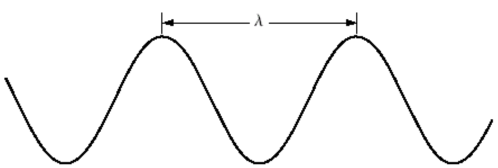
- Digital images
- The electromagnetic spectrum
- Digital images processing systems
- How are pictures made?
- Goals of image processing

Chapter 1: Introduction
Radiation-based images

Images based on radiation from ElectroMagnetic spectrum are most familiar, e.g. X-ray images and visible spectrum images.

EM waves can be thought of as propagating sinusoidal waves of varying wavelengths or as a stream of massless particles, each traveling in a wavelike pattern and moving at the speed of light.

FIGURE 2.11 Graphical representation of one wavelength.



Each massless particle contains a certain amount (or bundle) of energy. Each bundle of energy is called a **photon**.

If spectral bands are grouped according to energy per photon, we obtain the **spectrum** below.

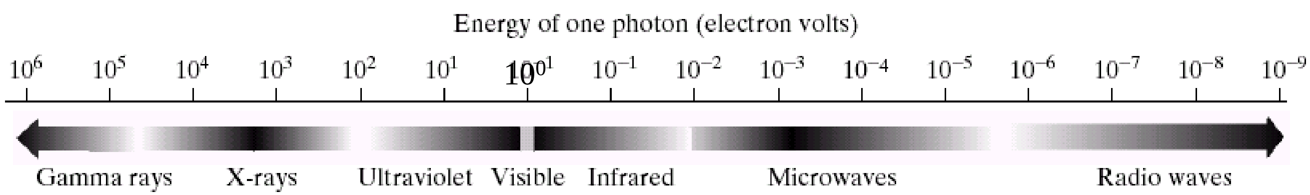
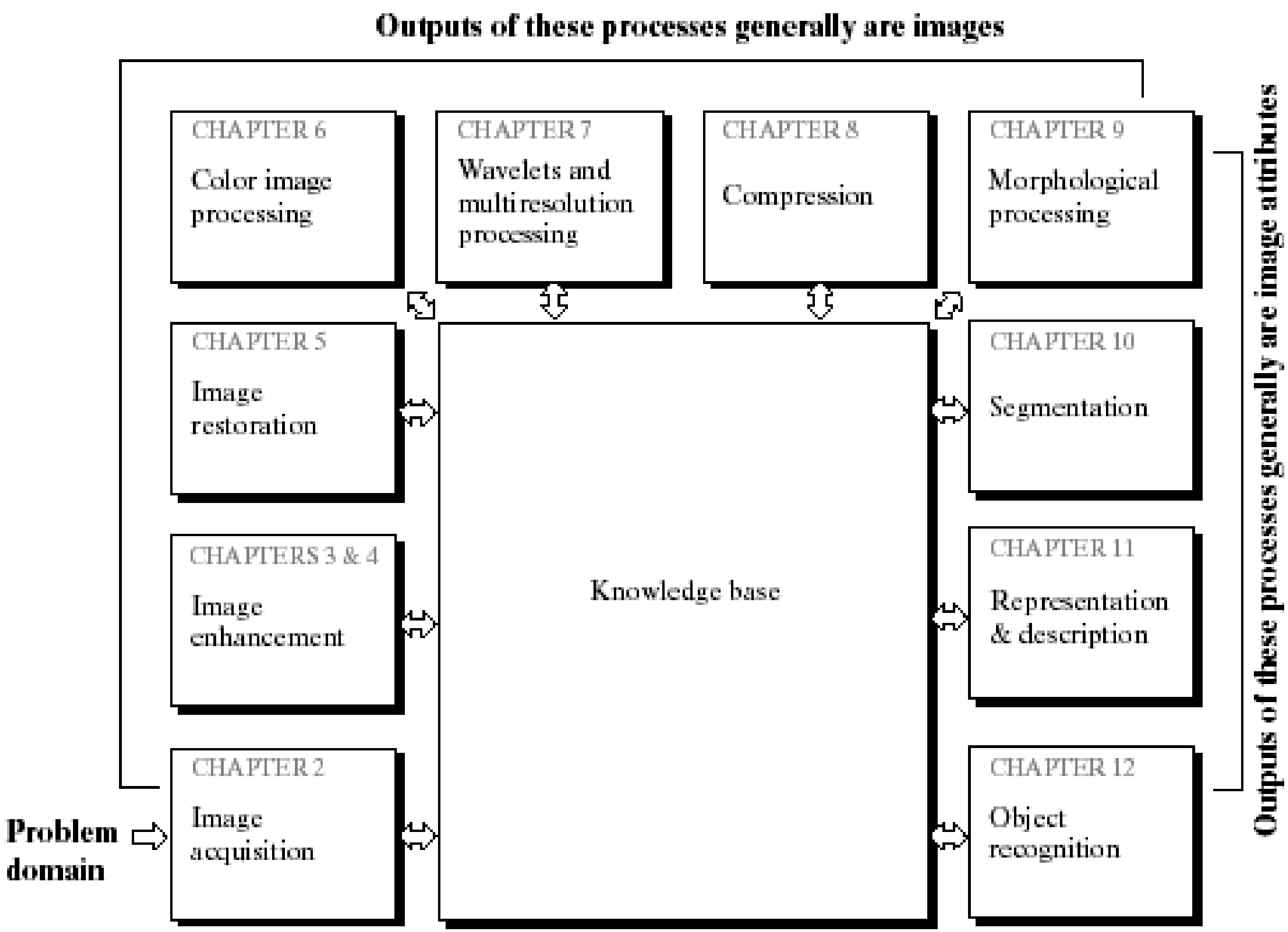


FIGURE 1.5 The electromagnetic spectrum arranged according to energy per photon.

Chapter 1: Introduction

FIGURE 1.23 Fundamental steps in digital image processing.



Chapter 2: Digital Image Fundamentals

Outline

Elements of Visual Perception

Light and the Electromagnetic Spectrum

Image Sensing and Acquisition

Sampling and Quantization

Chapter 2: Digital Image Fundamentals

- Structure of the human eye
- Rod and cones and scotopic and photopic vision
- Subjective brightness and brightness adaptation
- brightness discrimination and Weber ratio
- Mach band
- Simultaneous contrast and other visual illusions
- Light and the electromagnetic spectrum
- Image Sensing and Acquisition
- How to transform illumination energy into digital images?
- Image acquisition
- Image formation
- Image sampling and quantization
- Contouring effects
- Moiré patterns

Chapter 2: Digital Image Fundamentals

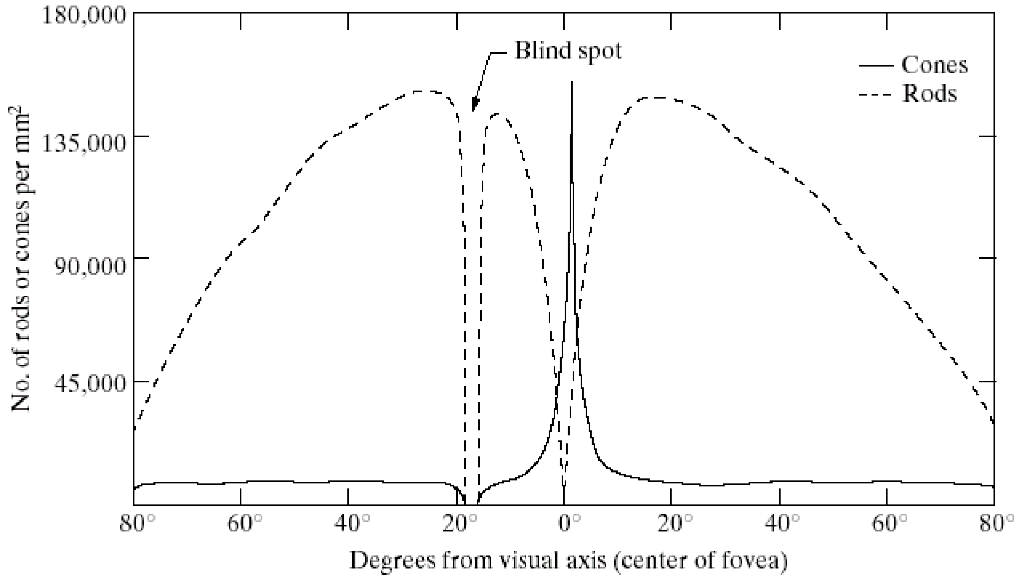


FIGURE 2.2 Distribution of rods and cones in the retina.

The distribution of rods and cones is radially symmetric wrt the fovea (central portion of the retina), except at the blind spot which includes no receptors.

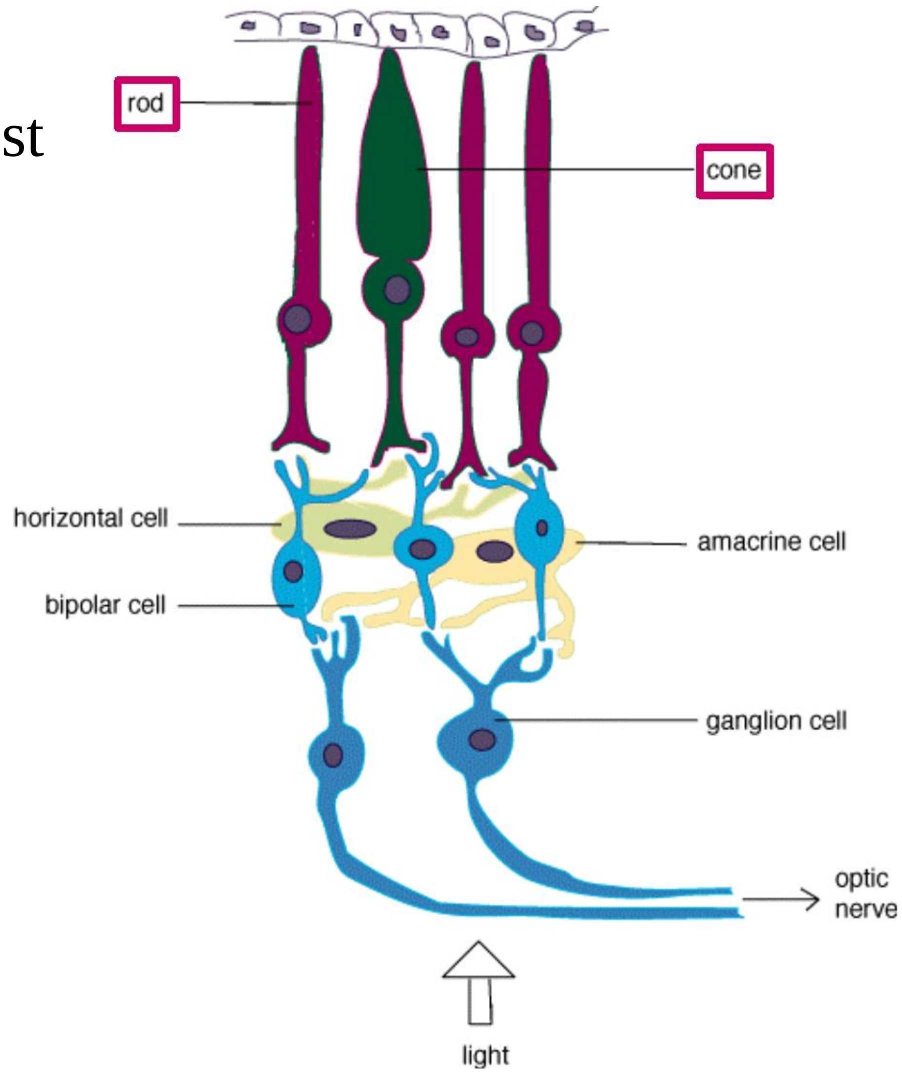
Cones are responsible for photopic (color or bright-light) vision; while **rods** are for **scotopic** (dim-light) vision.

Retina is circular with 1.5 mm in diameter with 150 000 cones/mm^2 , easily achievable with medium resolution CCD imaging chip of size 5mm x 5mm!

Structure of the Retina

Light receptors in the retina consist of two types: *rods* and *cones*.

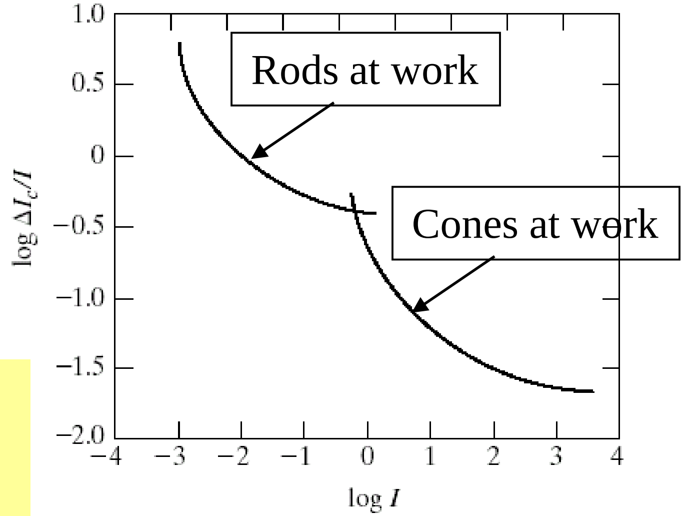
Rods are long slender receptors, 75~150 million, and cones are shorter and thicker, 6~7 million.



Chapter 2: Digital Image Fundamentals

A small Weber ratio indicates "good" brightness where a small percentage change in illumination is discriminable. On the other hand, a large Weber ratio represents "poor" brightness indicating that a large percentage change in intensity is needed.

The curve shows that brightness discrimination is poor (large Weber ratio) at low level of illumination, and it improves significantly (Weber ratio decreases) as background illumination increases.



The two branches illustrate the fact that at low levels of illumination, vision is carried out by the rods, whereas at high levels (showing better discrimination), cones are at work.

Chapter 2: Digital Image Fundamentals

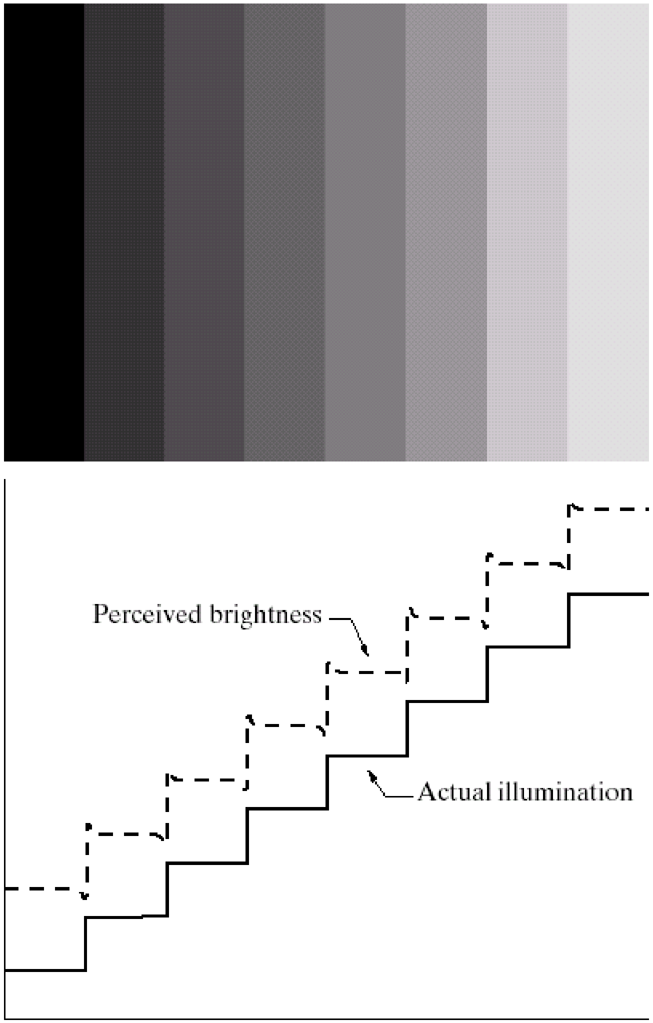
Perceived brightness is NOT a simple function of intensity.

Example 1: Mach bands

The reflected light intensity from each strip is uniform over its width and differs from its neighbors by a constant amount; nevertheless, the virtual appearance is that transitions at each bar appear brighter on the right side and darker on the left side.

The Mach band* effect can be used to estimate the impulse response of the visual system.

*Mach 1906.



a
b

FIGURE 2.7 (a) An example showing that perceived brightness is not a simple function of intensity. The relative vertical positions between the two profiles in (b) have no special significance; they were chosen for clarity.

Chapter 2: Digital Image Fundamentals

Definition:

Light is an electromagnetic radiation which, by simulation, arouses a sensation on the visual receptors making sight possible.

Sir Isaac Newton (1666) discovered that when a beam of sunlight is passed through a glass prism, the emerging beam of light is not white but consists instead of a continuous spectrum of colors ranging from violet to red. This is called the visible region of the spectrum, see next figure.

Chapter 2: Digital Image Fundamentals

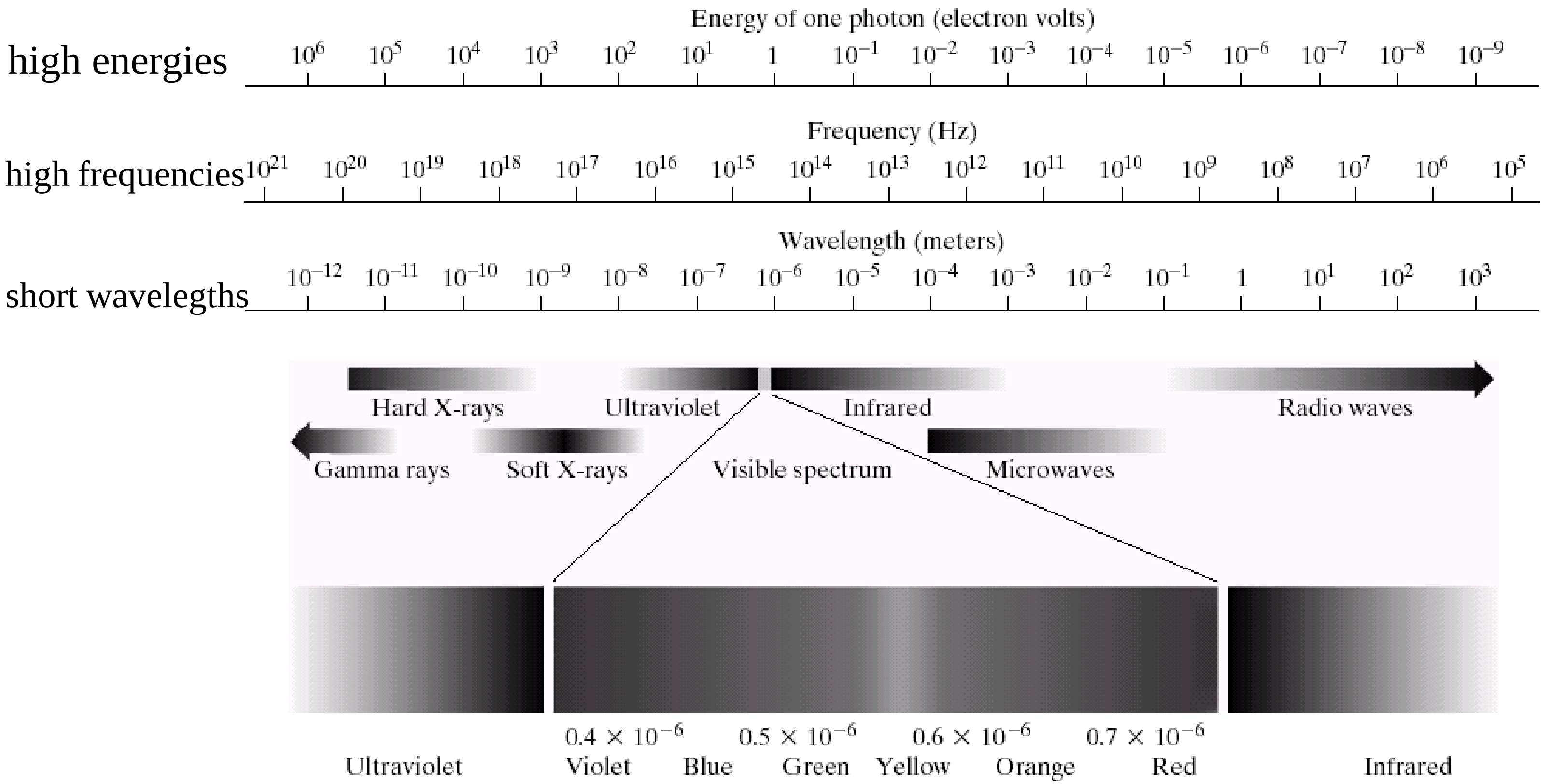


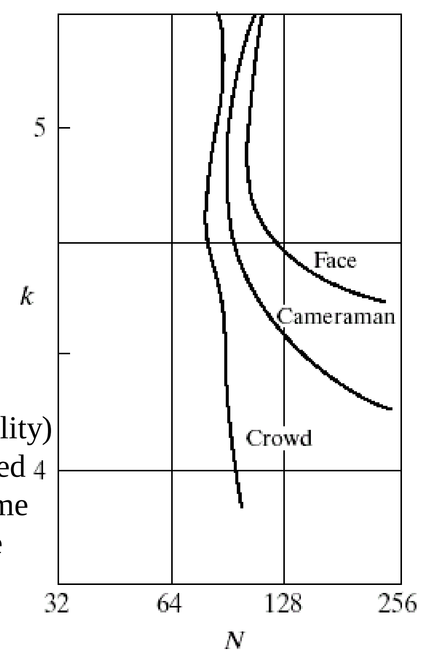
FIGURE 2.10 The electromagnetic spectrum. The visible spectrum is shown zoomed to facilitate explanation, but note that the visible spectrum is a rather narrow portion of the EM spectrum.

Chapter 2: Digital Image Fundamentals

Isopreference [Huang 1965] curves are plotted in the Nk-plane, where each point represents an image having values of N and k equal to the coordinates of that point.

Points lying on an isopreference curve correspond to images of equal subjective quality.

FIGURE 2.23 Representative isopreference curves for the three types of images in Fig. 2.22.



Comments:

1. Isopreference curves tend to shift right and upward (i.e. better image quality)
2. In images with a large amount of details, only a few gray levels are needed
3. In the other two image categories, the perceived quality remained the same in some intervals in which N was increased but k actually decreased! (more contrast in the image is perhaps preferred by some people!)

A Simple Image Formation Model

Consider the monochrome case, e.g., black and white images

Represent the spectral intensity distribution of the image by a continuous function $f(x,y)$, *i.e.*, for fixed value of (x,y), $f(x,y)$ is proportional to the grey level of the image at that point.

Of course,

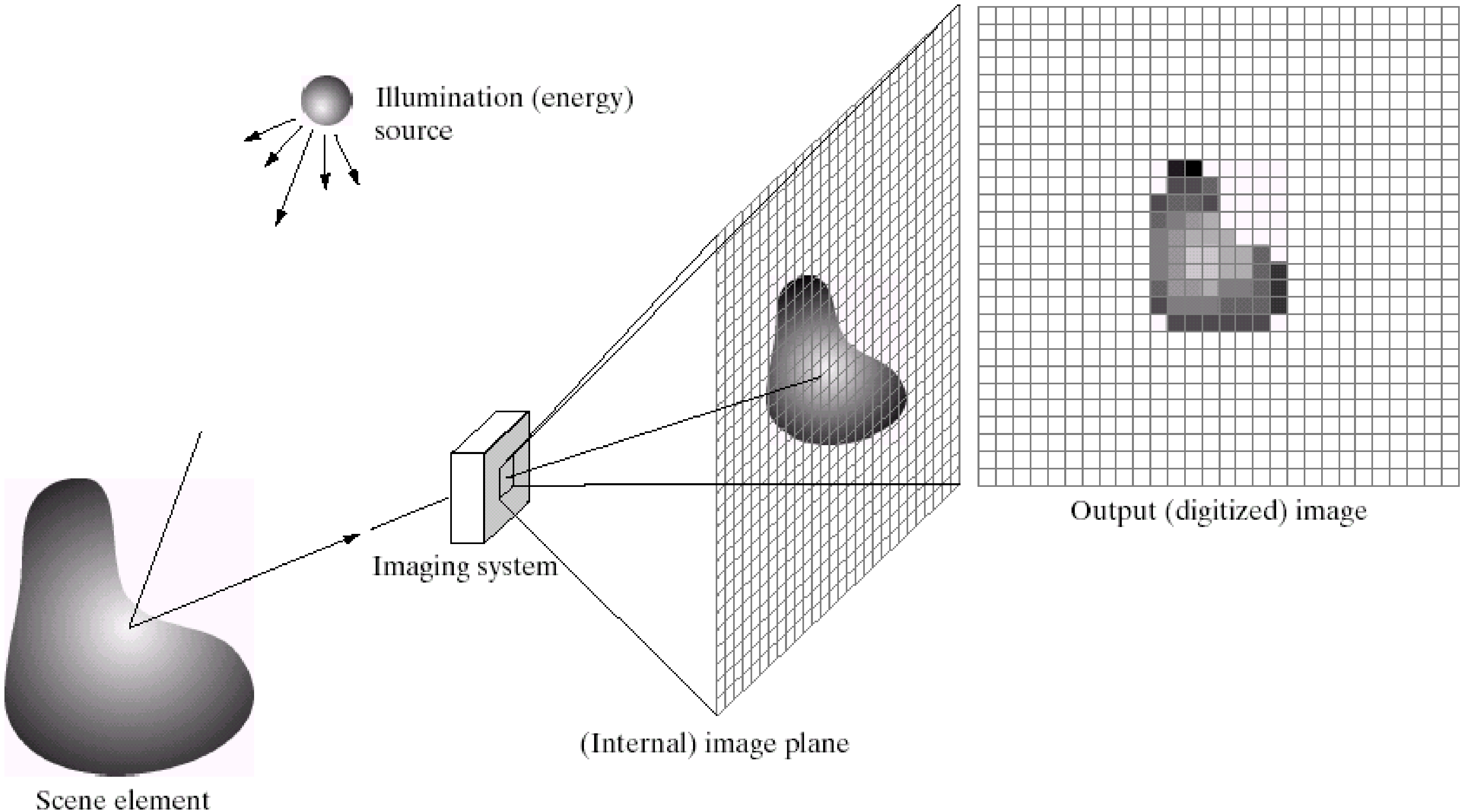
$$\textbf{(black) } 0 \le f(x,y) \le f_{max} \textbf{ (white)}$$

Why such limits?

Lower bound is because light intensity is a real positive quantity (recall that intensity f is proportional to $|E|^2$, where E is the electric field).

Upper bound is due to the fact that in all practical imaging systems, the physical system imposes some restrictions on the maximum intensity level of an image, e.g., film saturation and cathode ray tube phosphor heating.

Intermediate values between 0 and f_{max} are called shades of gray varying from black to white.



a b c d e

FIGURE 2.15 An example of the digital image acquisition process. (a) Energy ("illumination") source. (b) An element of a scene. (c) Imaging system. (d) Projection of the scene onto the image plane. (e) Digitized image.

Digital Image

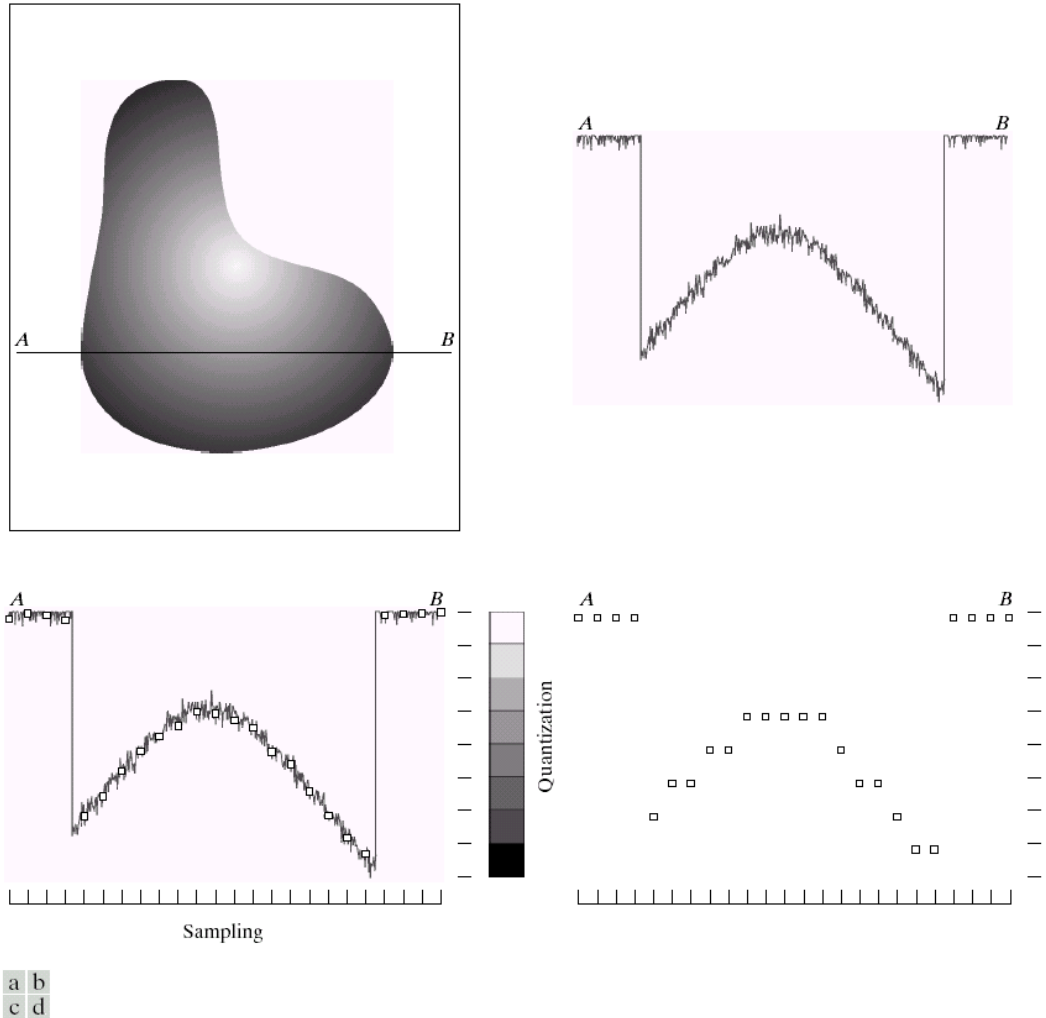
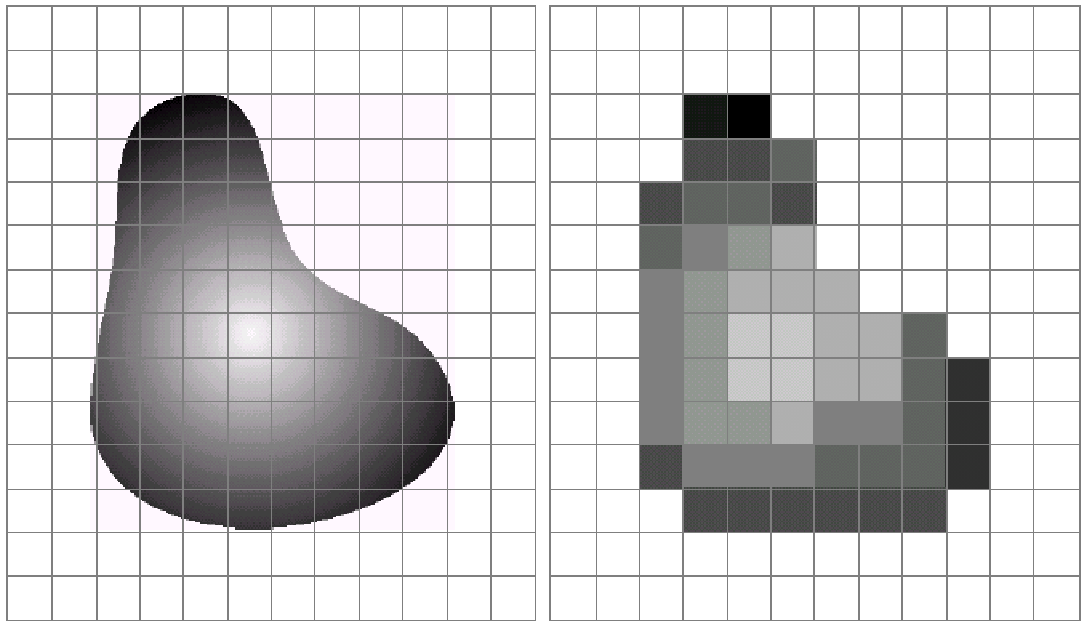


FIGURE 2.16 Generating a digital image. (a) Continuous image. (b) A scan line from A to B in the continuous image, used to illustrate the concepts of sampling and quantization. (c) Sampling and quantization. (d) Digital scan line.

Sampling and Quantization



a b

FIGURE 2.17 (a) Continuos image projected onto a sensor array. (b) Result of image sampling and quantization.

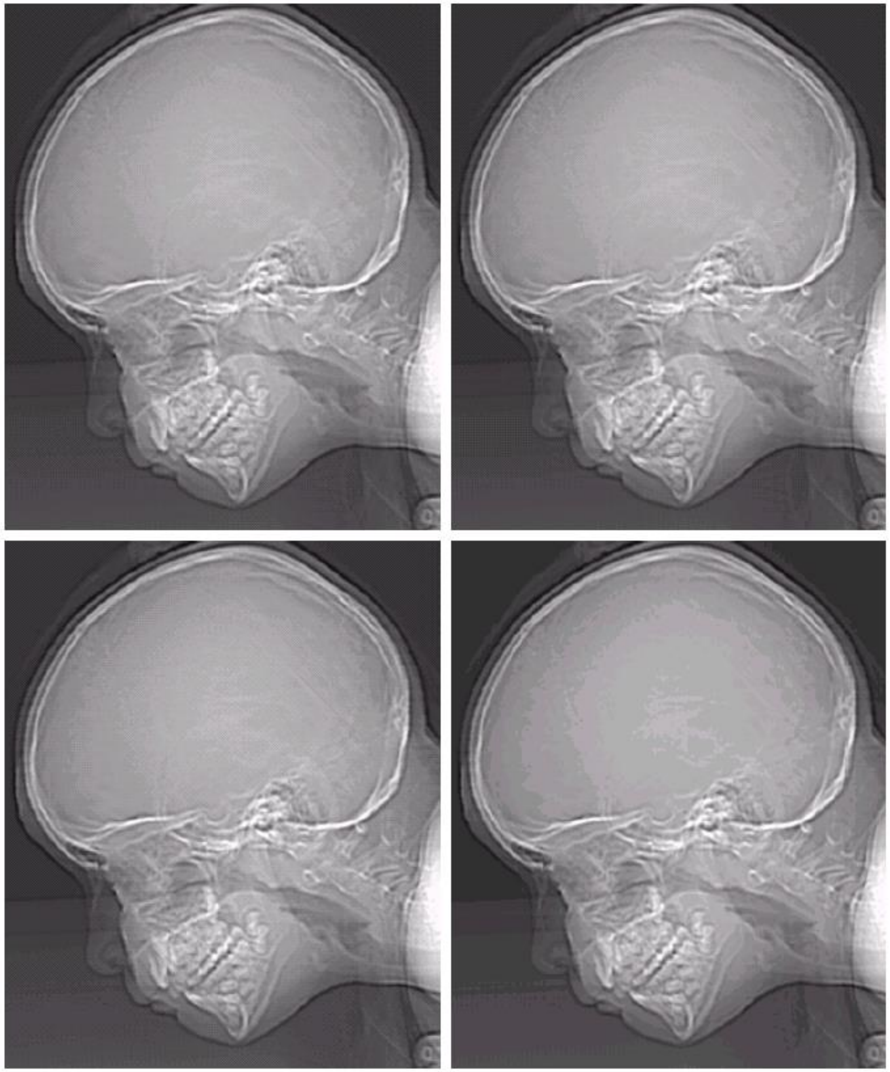
Contouring Effect

- If the number of quantization levels is not sufficient, contouring can be seen in the image.
- Contouring starts to become visible at 6 bits/pixel.
- Quantization should attempt to keep the quantization contours below the visible level.

To reduce this effect:

- Contrast Quantization,
- Dithering.

64 levels



a b
c d

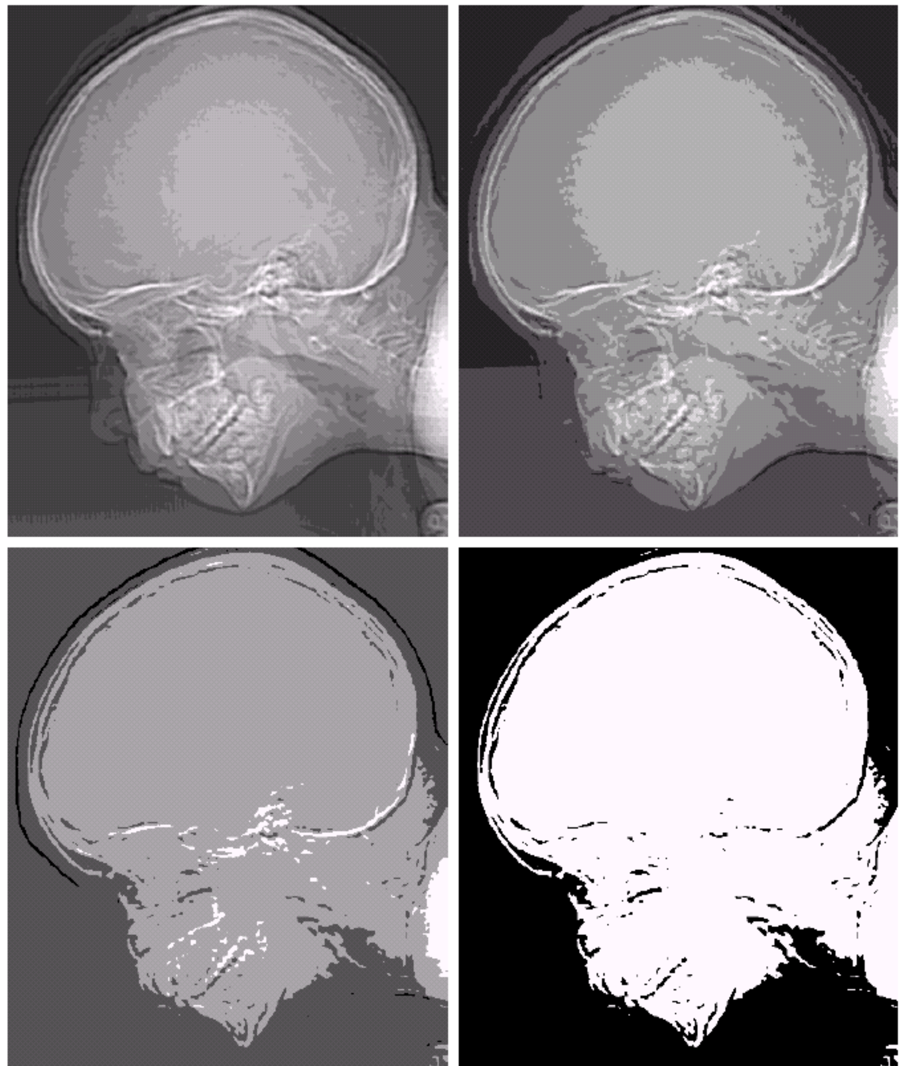
FIGURE 2.21 (a) 452 × 374, 256-level image. (b)–(d) Image displayed in 128, 64, and 32 gray levels, while keeping the spatial resolution constant.

in this 32-level image, note the appearance of very fine ridge-like structures in the areas of smooth gray levels, e.g. skull.

Due to insufficient number of gray levels, this artifact is more visible below and it is called false contouring.

e	f
g	h

FIGURE 2.21 *(Continued)* (e)–(h) Image displayed in 16, 8, 4, and 2 gray levels. (Original courtesy of Dr. David R. Pickens, Department of Radiology & Radiological Sciences, Vanderbilt University Medical Center.)



16	8
4	2

Chapter 3: Image Enhancement in the Spatial Domain

- Contrast stretching
- Grey level transformations
- Image negatives
- Log transformations
- Power-law transformations
- Piece-wise linear transformations
- Histogram equalisation
- Enhancement using logica and arithmetic operations
- Enhancement using spatial averaging
- Spatial filtering: linear and nonlinear filtering
- First and second order derivatives (Laplacian)
- Laplacian with high-boost filtering
- Combining spatial enhancement methods

Image Enhancement in the Spatial Domain

- Goal: Image enhancement seeks
  - to improve the visual appearance of an image, or
  - convert it to a form suited for analysis by a human or a machine.
- Image enhancement does not, however,
  - seek to restore the image, nor
  - increase its information contents
- Peculiarity:
  - actually, there is some evidence which suggests that a distorted image can be more pleasing than a perfect image!

Image Enhancement in the Spatial Domain

Major Problem in Image Enhancement:

the lack of a general standard of **image quality** makes it very difficult to evaluate the performance of different IE schemes.

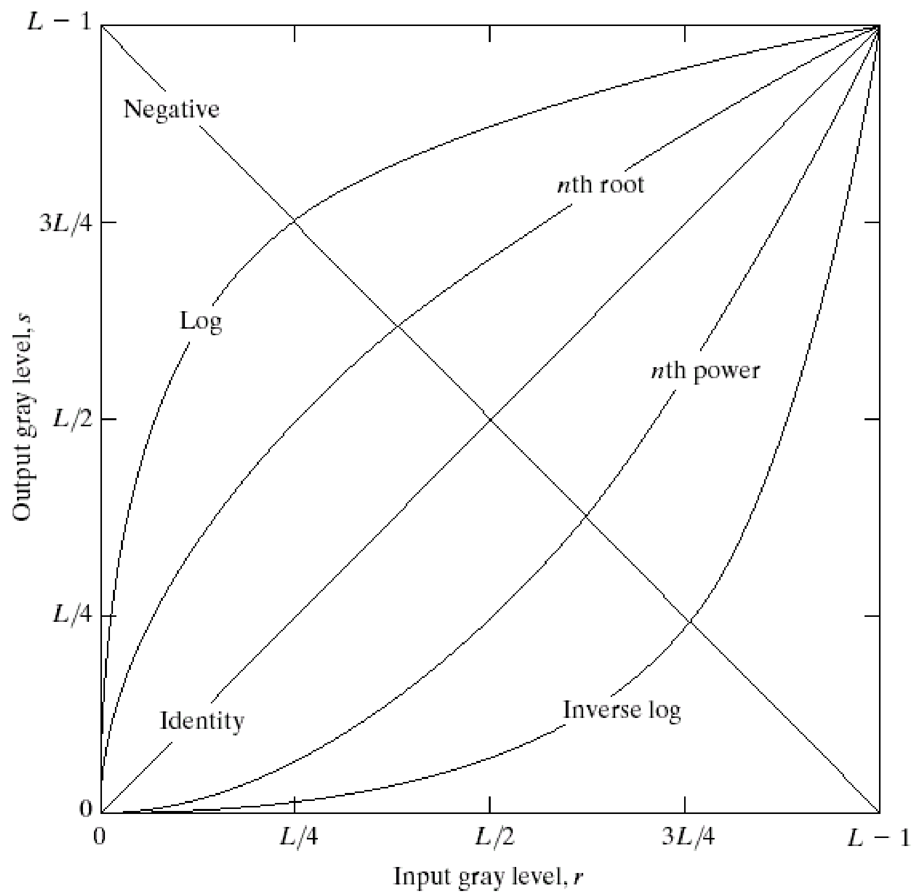
Thus, Image Enhancement algorithms are mostly application-dependent, subjective and often ad-hoc.

Therefore, mostly subjective criteria are used in evaluating image enhancement algorithms.

Image Enhancement in the Spatial Domain

Basic Grey Level Transformations

FIGURE 3.3 Some basic gray-level transformation functions used for image enhancement.



$$s = T[r]$$

Image Enhancement in the Spatial Domain

Power-Law transformations

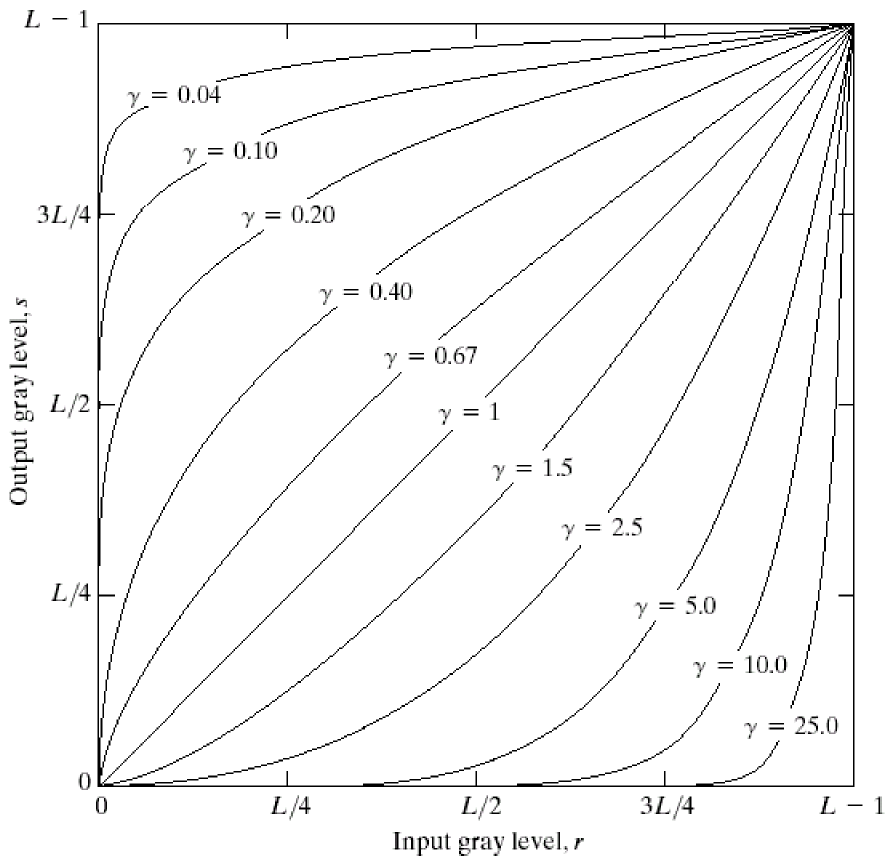


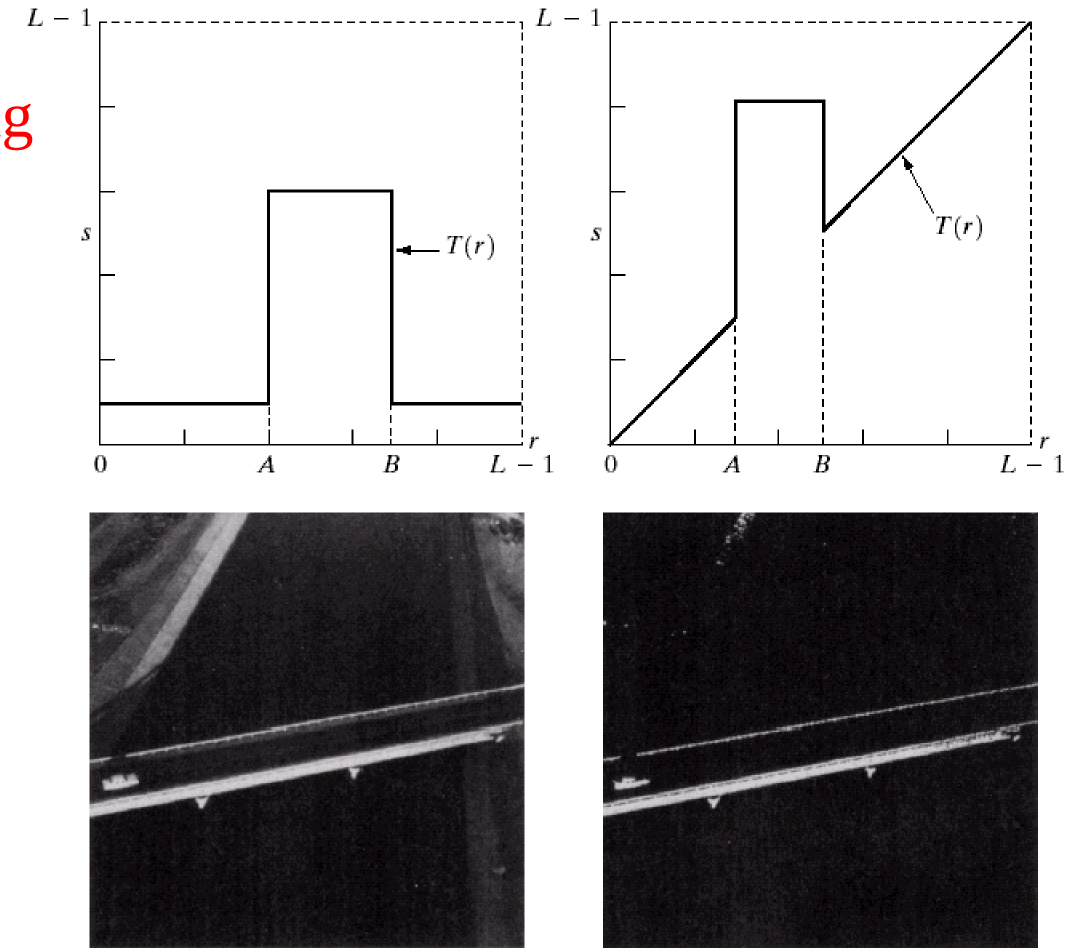
FIGURE 3.6 Plots of the equation $s = cr^{\gamma}$ for various values of γ ($c = 1$ in all cases).

$s = cr^{\gamma}$ (c and γ are positive constants)

Image Enhancement in the Spatial Domain

Piecewise-Linear Transformations

2. Level slicing



a b
c d

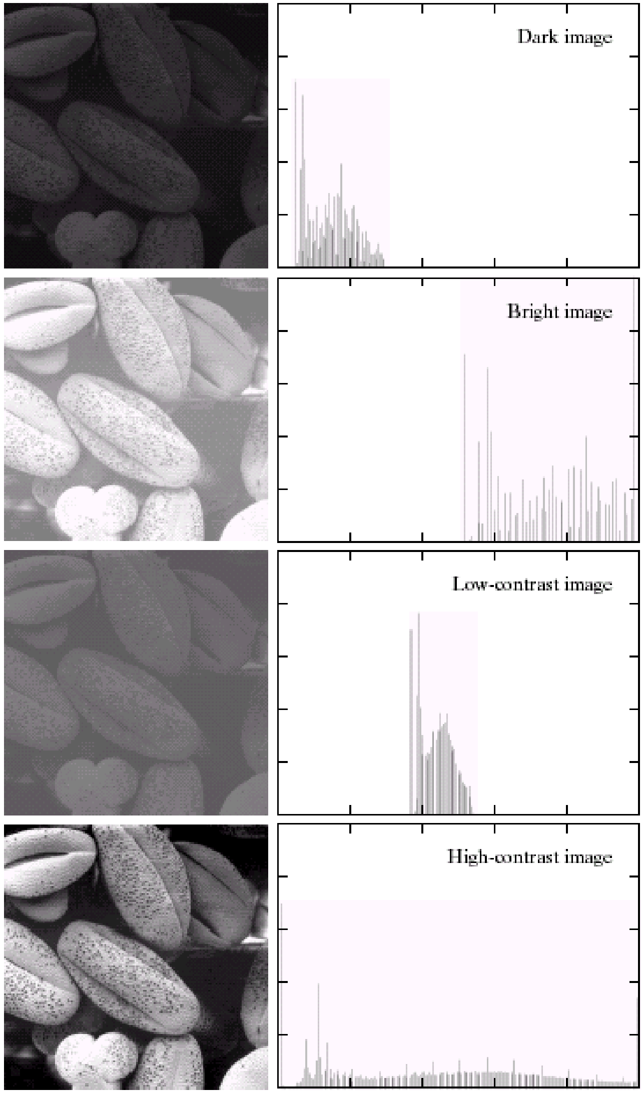
FIGURE 3.11 (a) This transformation highlights range [A, B] of gray levels and reduces all others to a constant level. (b) This transformation highlights range [A, B] but preserves all other levels. (c) An image. (d) Result of using the transformation in (a).

an input image

result after applying transformation in (a).

Applications: enhancing features, e.g. masses of water in satellite imagery and enhancing flaws in X-ray images.

Image Enhancement in the Spatial Domain
Histogram Processing



Basic image types:

dark

light

low-contrast

high-contrast

a b

FIGURE 3.15 Four basic image types: dark, light, low contrast, high contrast, and their corresponding histograms. (Original image courtesy of Dr. Roger Heady, Research School of Biological Sciences, Australian National University, Canberra, Australia.)

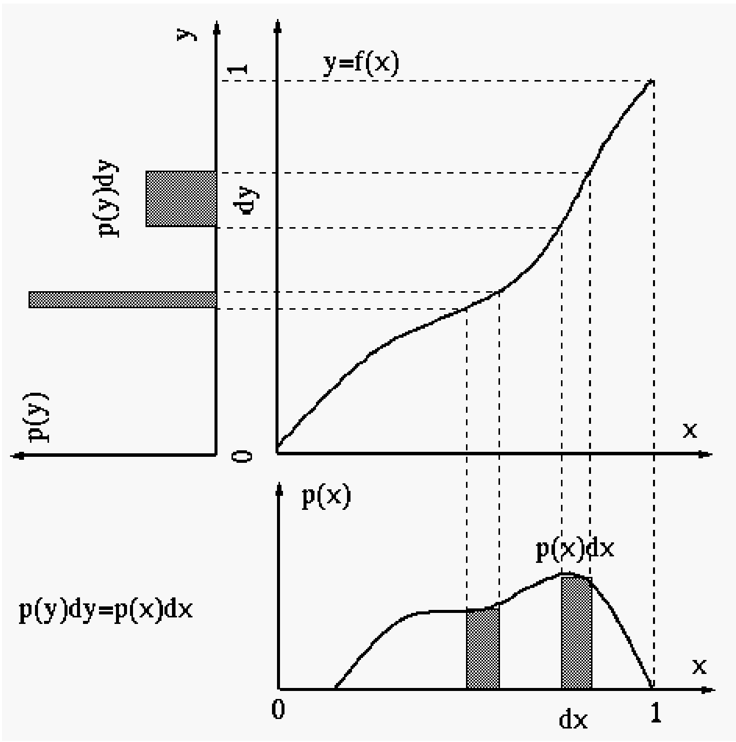
Image Enhancement in the Spatial Domain

Histogram Equalization

- To transfer the gray levels so that the histogram of the resulting image is equalized to be a constant:

$$h[i] = \text{constant}, \quad \text{for all } i$$

- The purposes:
  - to equally use all available gray levels;
  - for further histogram specification.



- This figure shows that for any given mapping function $y = f(x)$ between the input and output images, the following holds:

$$p(y)dy = p(x)dx$$

- i.e., the number of pixels mapped from *x* to *y* is unchanged.

Image Enhancement in the Spatial Domain

Histogram Equalization

- To equalize the histogram of the output image, we let *p(y)* be a constant *c*.
  $c dy = p(x)dx$ or $c\frac{dy}{dx} = p(x)$ or $c\frac{df(x)}{dx} = p(x)$
- If the gray levels are assumed to be in the ranges between 0 and 1 $(0 \le x \le 1, 0 \le y \le 1)$, $\int_0^1 c\, dy = 1 \ \rightarrow c = 1$
- $\int_0^x \frac{df(u)}{du} du = \int_0^x p(u)du$
- $\int_0^x df(u) = \int_0^x p(u)du$
- $f(x) - f(0) = \int_0^x p(u)du = F(x) - F(0)$
- Finally, we have the following mapping function for histogram equalization:

  $$y = f(x) = \int_0^x p(u)du = F(x),$$

  where *F(x)* is the cumulative probability distribution of the input image, which monotonically increases.

(Notation is confusing. *F(x)* is the integral of *p(x),* not *f(x)*)

Image Enhancement in the Spatial Domain

Histogram Equalization

Intuitively, histogram equalization is realized by the following:

- If *p(x)* is high, *y=f(x)* has a steep slope, *dy* will be wide, causing *p(y)* to be low to keep *p(y)dy* = *p(x)dx*;
- If *p(x)* is low, *y=f(x)* has a shallow slope, *dy* will be narrow, causing *p(y)* to be high to keep p*(y)dy* = *p(x)dx*

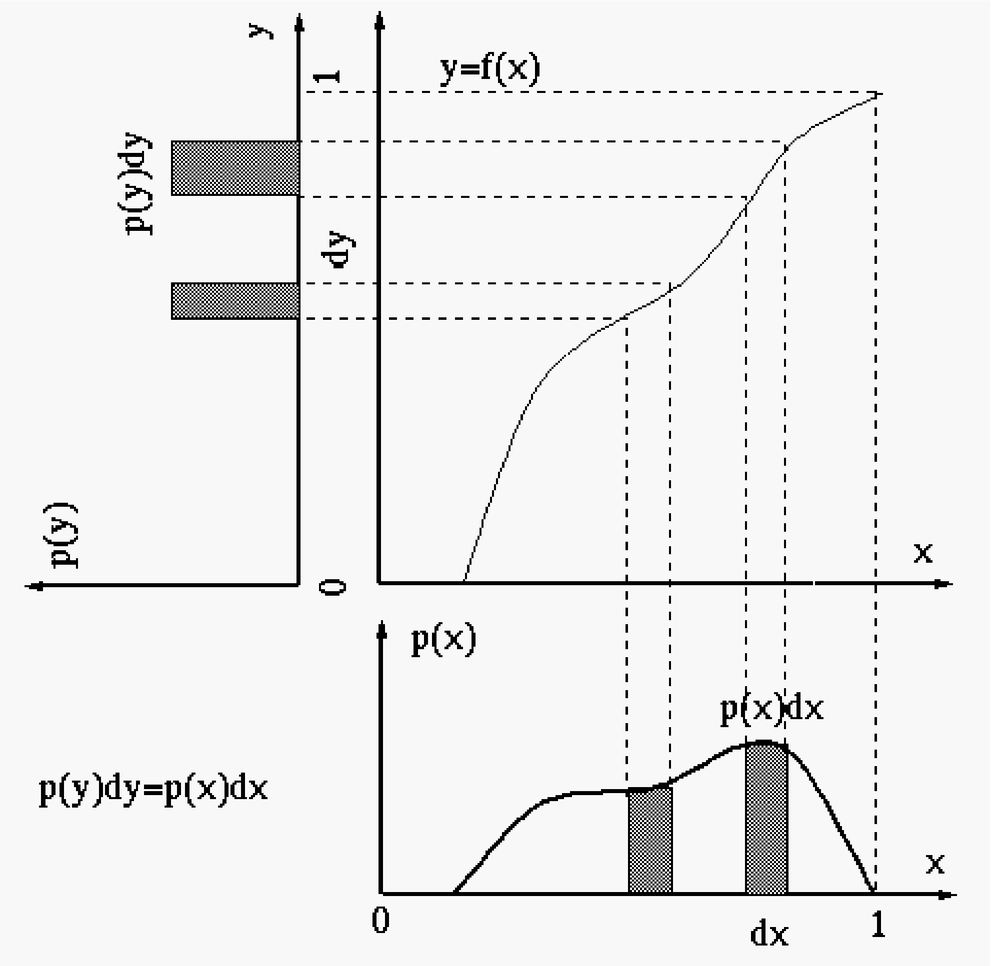
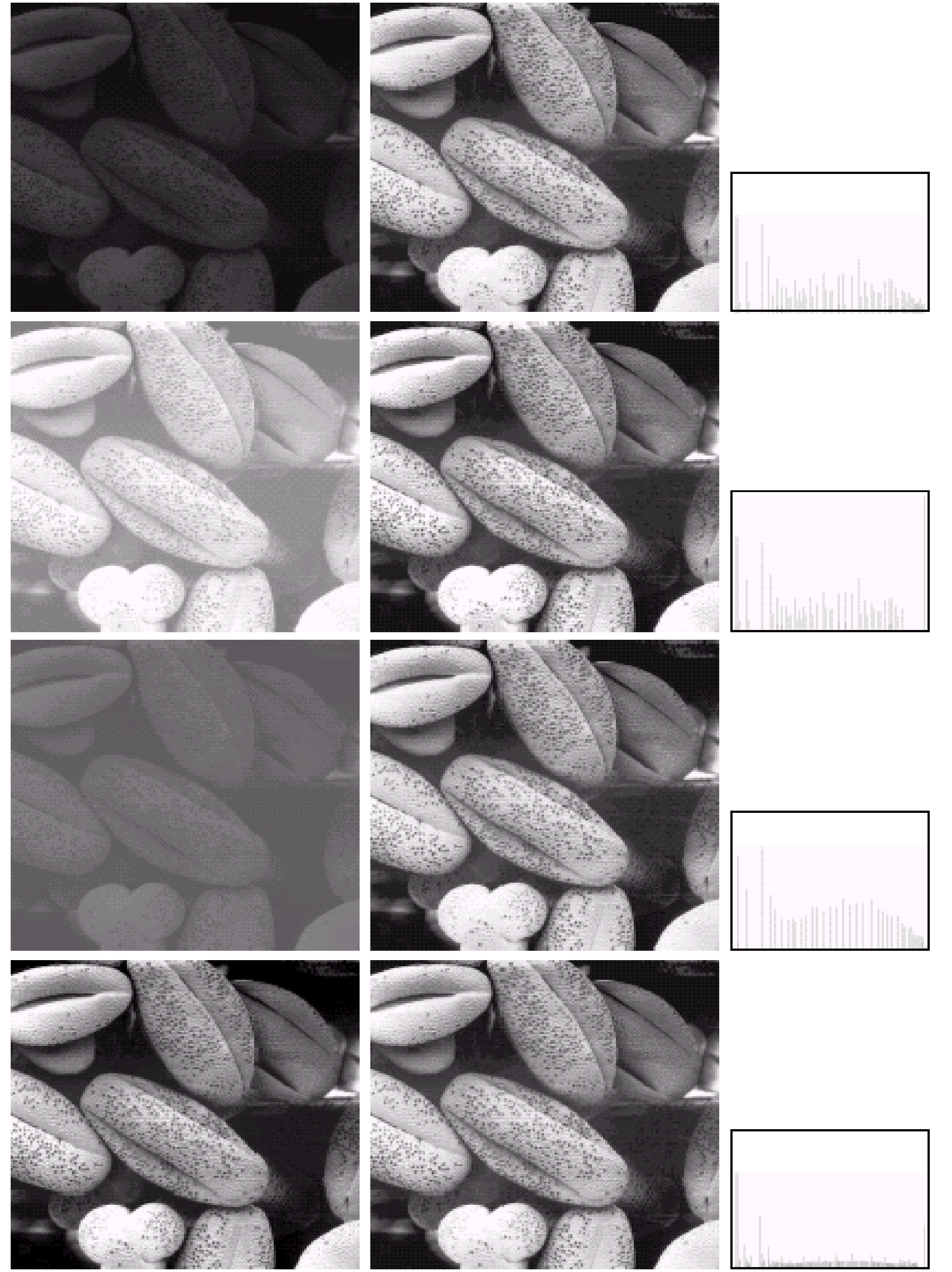


Image Enhancement in the Spatial Domain

Histogram Equalization



a b c

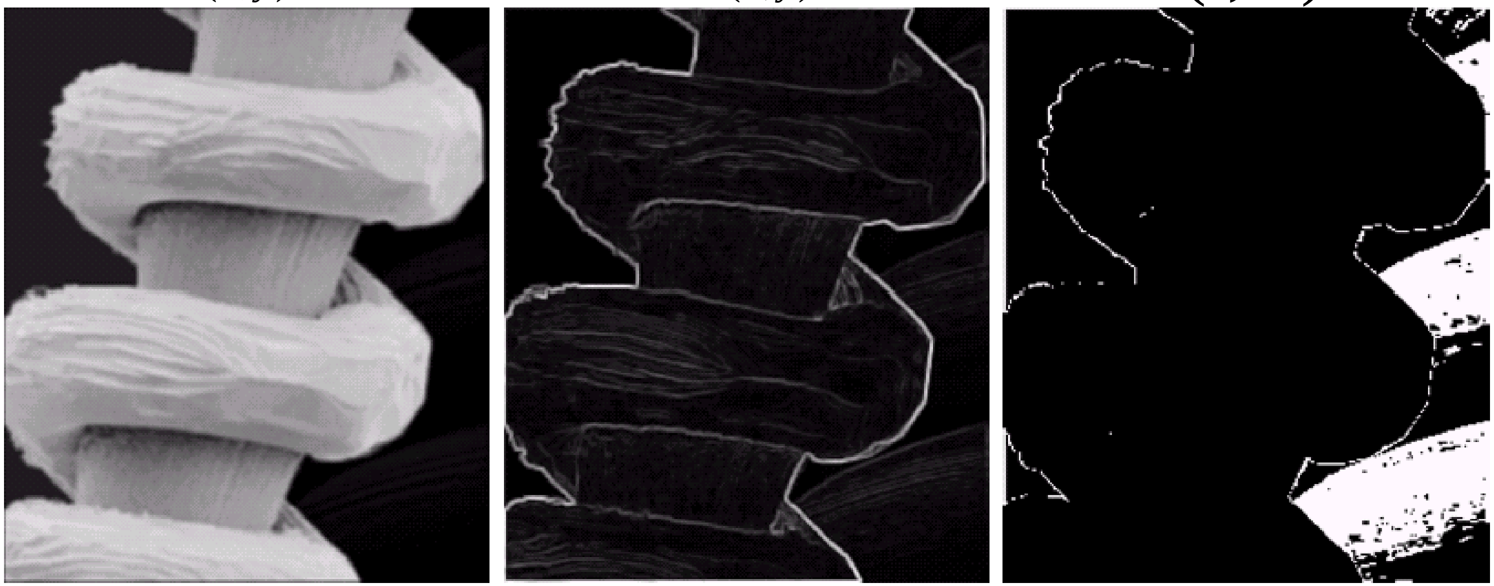
FIGURE 3.17 (a) Images from Fig. 3.15. (b) Results of histogram equalization. (c) Corresponding histograms.

Image Enhancement in the Spatial Domain

FIGURE 3.24 SEM image of a tungsten filament and support, magnified approximately 130×. (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene).

Image Enhancement in the Spatial Domain

$m_{S(x,y)}$ $\sigma_{S(x,y)}$ $(1, E)$



a b c

FIGURE 3.25 (a) Image formed from all local means obtained from Fig. 3.24 using Eq. (3.3-21). (b) Image formed from all local standard deviations obtained from Fig. 3.24 using Eq. (3.3-22). (c) Image formed from all multiplication constants used to produce the enhanced image shown in Fig. 3.26.

$$g(x,y) = \begin{cases} E.f(x,y) & \text{if} \quad m_{S(x,y)} < k_0 m_G \quad \text{and} \quad k_1\sigma_G < \sigma_{S(x,y)} < k_2\sigma_G \\ f(x,y) & \text{otherwise} \end{cases}$$

Image Enhancement in the Spatial Domain

FIGURE 3.26 Enhanced SEM image. Compare with Fig. 3.24. Note in particular the enhanced area on the right side of the image.

$$g(x,y) = \begin{cases} E.f(x,y) & \text{if} \quad m_{S(x,y)} < k_0 m_G \quad \text{and} \quad k_1\sigma_G < \sigma_{S(x,y)} < k_2\sigma_G \\ f(x,y) & \text{otherwise} \end{cases}$$

Image Enhancement in the Spatial Domain

Enhancement using averaging operations

original

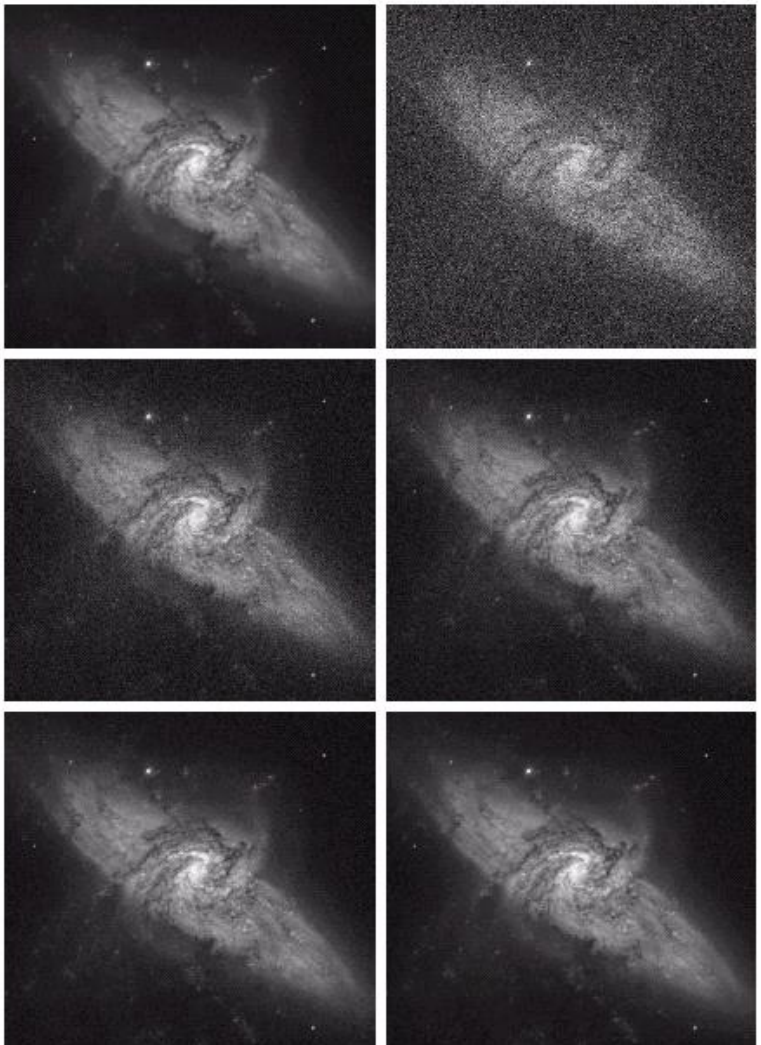
noisy image, N(0,64^2)

result of aver 8 noisy images

16 noisy images

64 noisy images

128 noisy images



a b
c d
e f

FIGURE 3.30 (a) Image of Galaxy Pair NGC 3314. (b) Image corrupted by additive Gaussian noise with zero mean and a standard deviation of 64 gray levels. (c)–(f) Results of averaging $K = 8, 16, 64,$ and 128 noisy images. (Original image courtesy of NASA.)

Image Enhancement in the Spatial Domain

Enhancement using spacial averaging operations

Consider a noisy image:

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

where the second term is noise which is uncorrelated with the input and has zero mean. Then, averaging K different noisy images:

$$\bar{g}(x,y) = \frac{1}{K}\sum_{i=1}^{K} g_i(x,y)$$

produces an output image with

$$E[\bar{g}(x,y)] = f(x,y) \quad and \quad \sigma_{\bar{g}}^2 = \frac{1}{K}\sigma_{\eta}^2$$

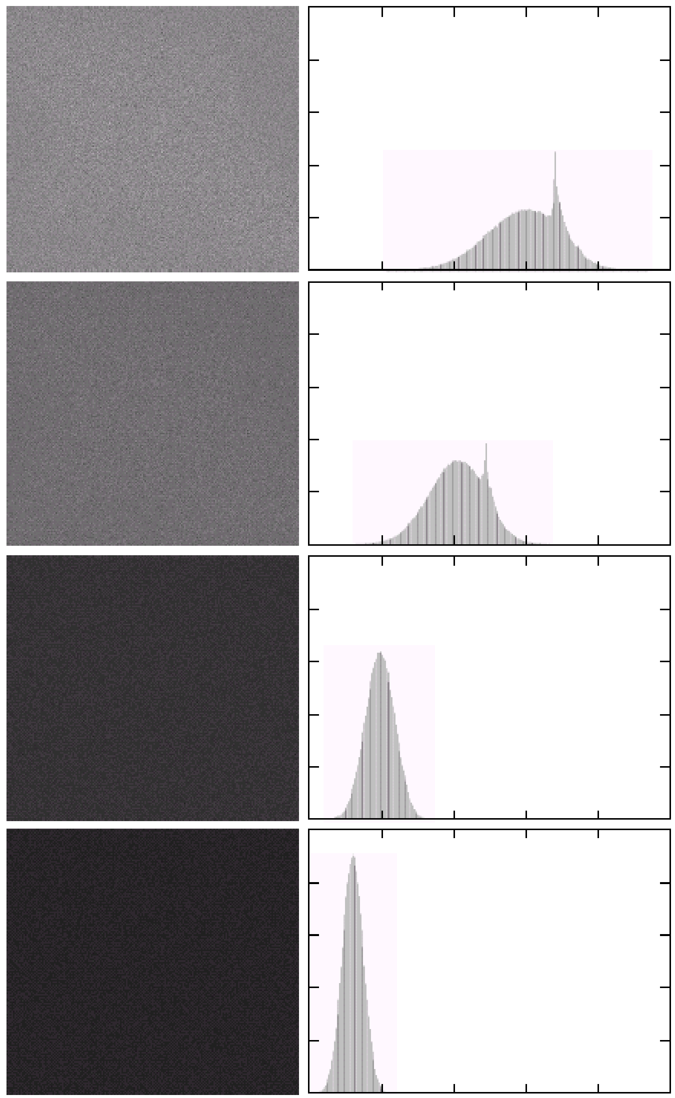
Image Enhancement in the Spatial Domain

K=8

K=16

K=64

K=128



a b

FIGURE 3.31 (a) From top to bottom: Difference images between Fig. 3.30(a) and the four images in Figs. 3.30(c) through (f), respectively. (b) Corresponding histograms.

notice how the noise variance is decreasing with increasing K.

Image Enhancement in the Spatial Domain
Linear Filtering

FIGURE 3.33 Another representation of a general 3×3 spatial filter mask.

w_1	w_2	w_3
w_4	w_5	w_6
w_7	w_8	w_9

$$g(x,y) = \frac{\sum_{m=-M}^{M} \sum_{n=-N}^{N} w(m,n) f(s+m, y+n)}{\sum_{m=-M}^{M} \sum_{n=-N}^{N} w(m,n)}$$

where *g(x,y)* is the output image and *f(x,y)* is the input image. In the mask above, *M=N=1*.

Image Enhancement in the Spatial Domain
Spacial averaging operations

Consider again a noisy image:

$$g(x,y) = f(x,y) + \eta_{in}(x,y)$$

where the second term is noise which is uncorrelated with the input and has zero mean. Let's apply a local averaging filter (all weights are equal) with size K=(2M+1)x(2N+1):

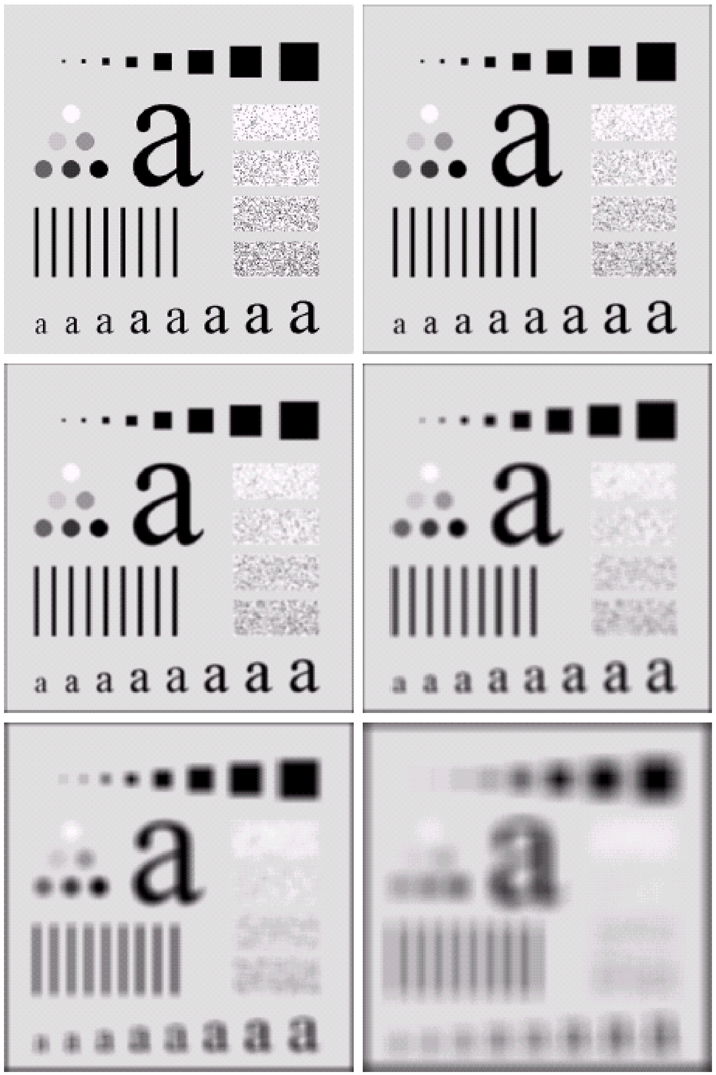
$$\bar{g}(x,y) = \frac{1}{K}\sum\sum_{(x,y)\in W} g(x,y) = \left[\frac{1}{K}\sum\sum_{(x,y)\in W} f(x,y)\right] + \eta_{out}(x,y)$$

produces an output image with

$$\sigma^2_{out} = \frac{1}{K}\sigma^2_{in}$$

Therefore, if the input is constant over W, the SNR has improved by a factor of K!!

Image Enhancement in the Spatial Domain



a b
c d
e f

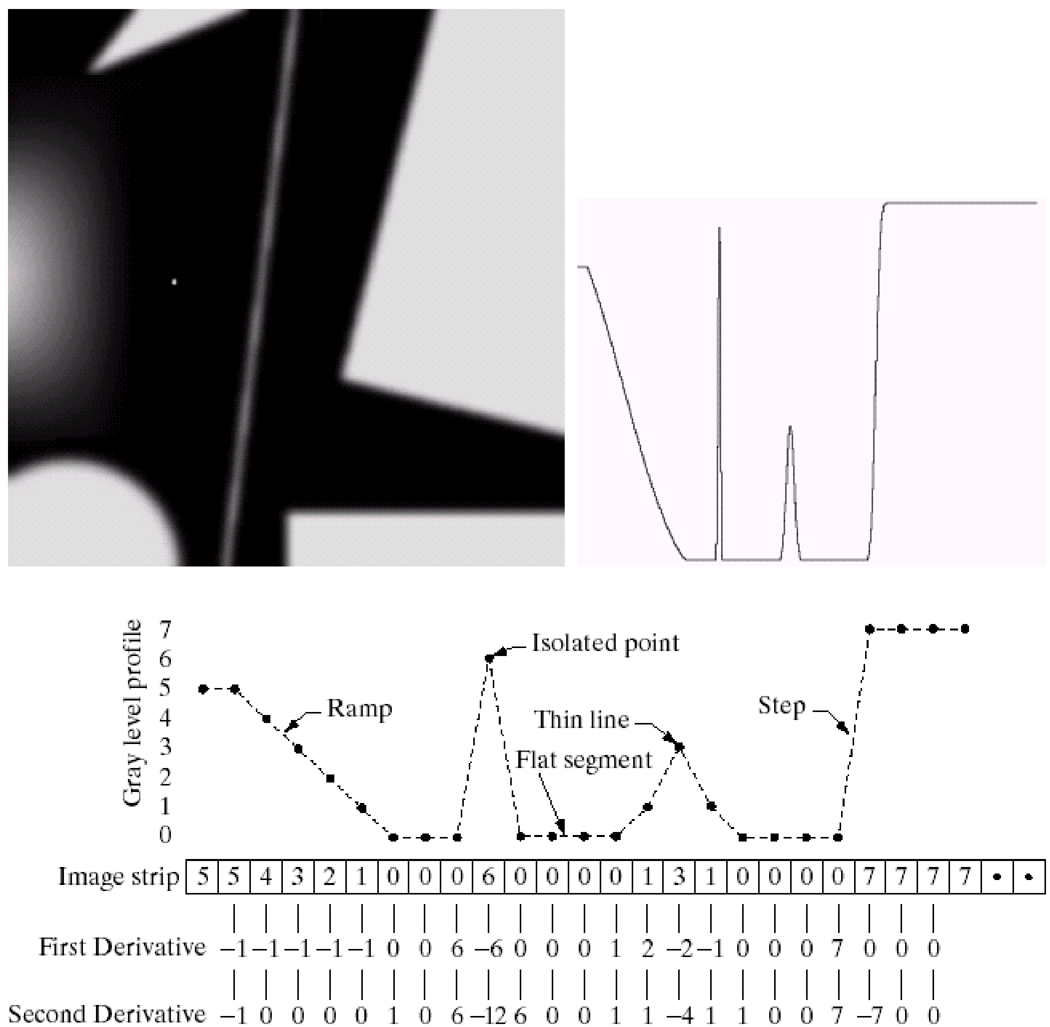
FIGURE 3.35 (a) Original image, of size 500 × 500 pixels. (b)–(f) Results of smoothing with square averaging filter masks of sizes n = 3, 5, 9, 15, and 35, respectively. The black squares at the top are of sizes 3, 5, 9, 15, 25, 35, 45, and 55 pixels, respectively; their borders are 25 pixels apart. The letters at the bottom range in size from 10 to 24 points, in increments of 2 points; the large letter at the top is 60 points. The vertical bars are 5 pixels wide and 100 pixels high; their separation is 20 pixels. The diameter of the circles is 25 pixels, and their borders are 15 pixels apart; their gray levels range from 0% to 100% black in increments of 20%. The background of the image is 10% black. The noisy rectangles are of size 50 × 120 pixels.

Image Enhancement in the Spatial Domain

Sharpening Spacial Filters

a b
c

FIGURE 3.38
(a) A simple image. (b) 1-D horizontal gray-level profile along the center of the image and including the isolated noise point.
(c) Simplified profile (the points are joined by dashed lines to simplify interpretation).



First Order Derivative
Second Order Derivatives

Image Enhancement in the Spatial Domain

Sharpening Spacial Filters: Laplacian

- **Isotropic** 2nd order derivative (Laplacian)

$$\nabla^2 f = \frac{\partial^2 f}{\partial^2 x^2} + \frac{\partial^2 f}{\partial^2 y^2}$$

- In digital form:

$$\frac{\partial^2 f}{\partial^2 x^2} = f(x+1, y) + f(x-1, y) - 2f(x, y)$$

- in the x-direction and in the y-direction:

$$\frac{\partial^2 f}{\partial^2 y^2} = f(x, y+1) + f(x, y-1) - 2f(x, y)$$

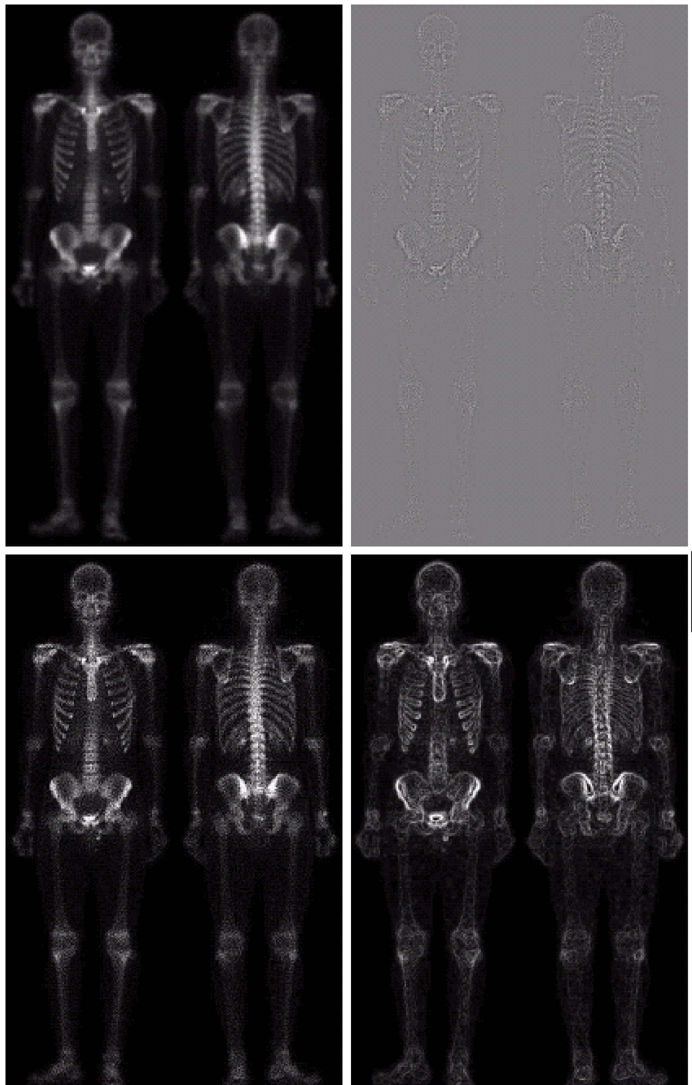
- 2-D Laplacian:

$$\nabla^2 f = \frac{\partial^2 f}{\partial^2 x^2} + \frac{\partial^2 f}{\partial^2 y^2}$$

Image Enhancement in the Spatial Domain

Combining spatial enhancement methods

original



Laplacian of original

Sobel of orig

original+ Laplacian

a b
c d

FIGURE 3.46 (a) Image of whole body bone scan. (b) Laplacian of (a). (c) Sharpened image obtained by adding (a) and (b). (d) Sobel of (a).

- Single technique may not produce desirable results
- Must thus devise a strategy for the given application at hand

This application:
nuclear whole body scan
want to detect deseases, e.g. bone infection and tumors

Strategy:

- use Laplacian to highlight details,
- gradient to enhance edges,
- grey-level trans. to increase dynamic range

Chapter 4: Enhancement in the frequency domain

- Linear filters (notch, lowpass, highpass, bandpass, bandreject filters)
- Gaussian and Butterworth filters
- Laplacian and high-boost filtering
- Homomorphic filtering

Chapter 4
Image Enhancement in the Frequency Domain

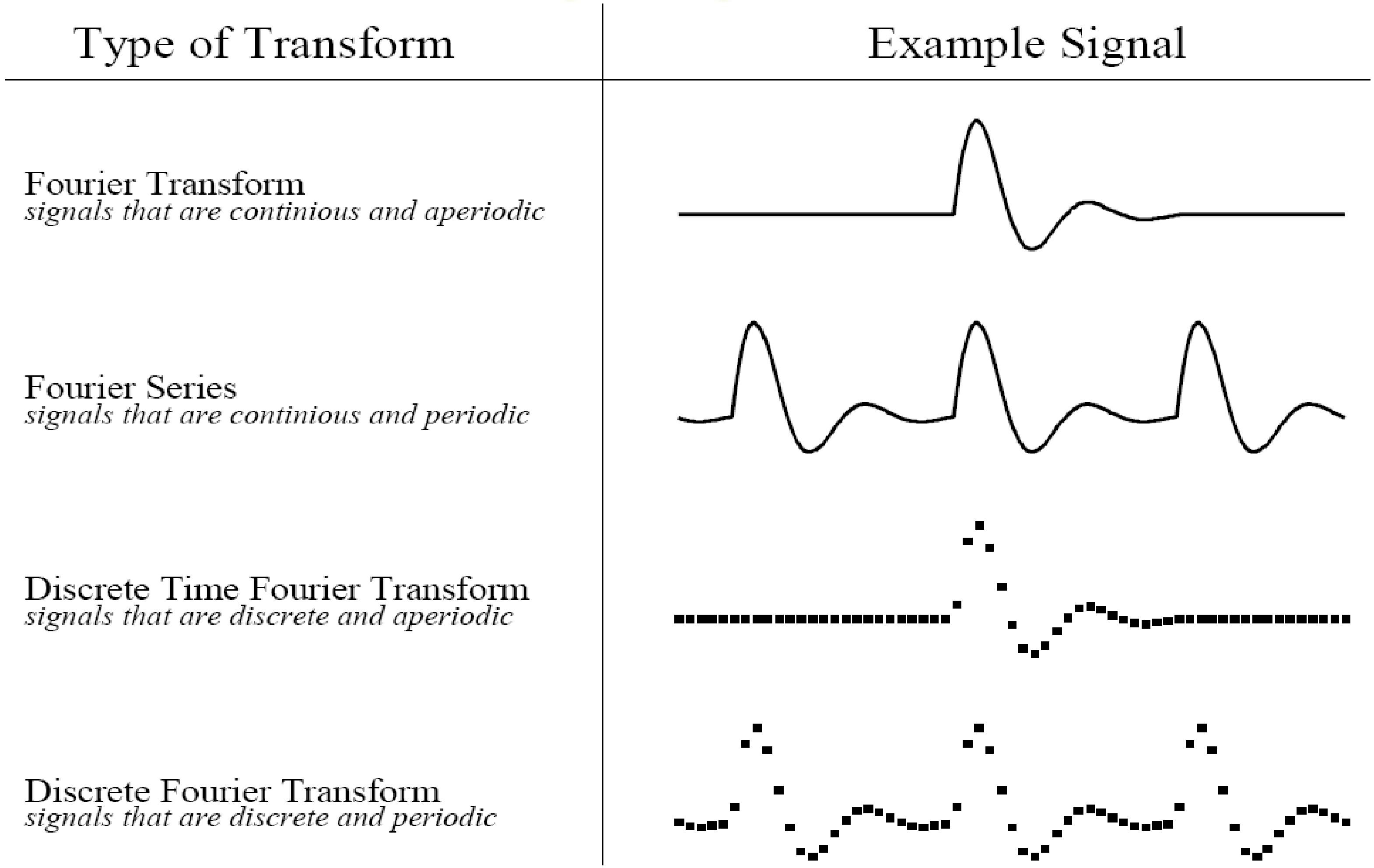
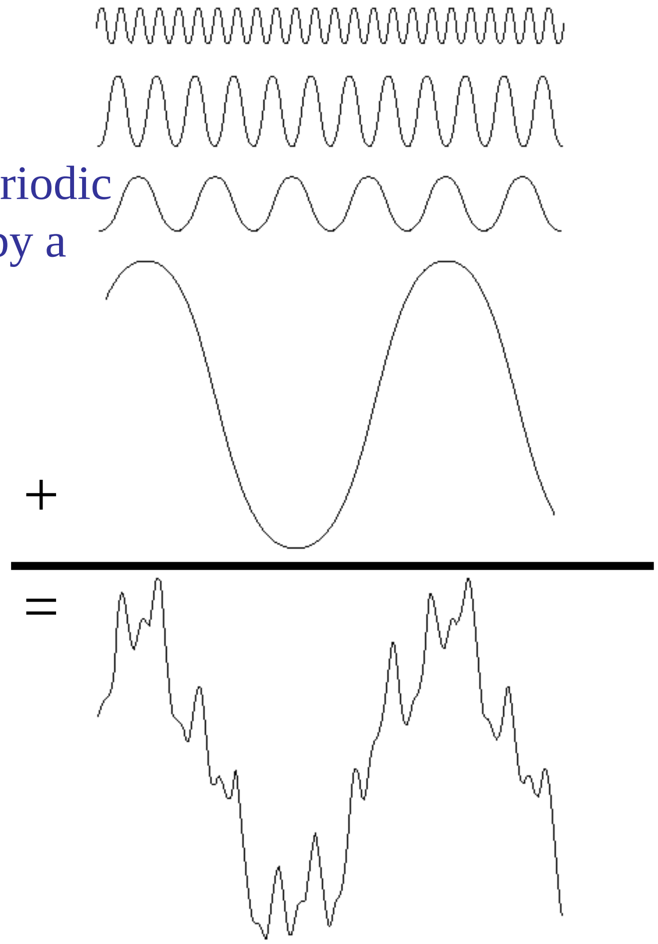


FIGURE 8-2
Illustration of the four Fourier transforms. A signal may be continuous or discrete, and it may be periodic or aperiodic. Together these define four possible combinations, each having its own version of the Fourier transform. The names are not well organized; simply memorize them.

Chapter 4
Image Enhancement in the Frequency Domain

Fourier, 1807

Fourier series states that a periodic function can be represented by a weighted sum of sinusoids



Periodic and non-periodic functions can be represented by an integral of weighted sinusoids

FIGURE 4.1 The function at the bottom is the sum of the four functions above it. Fourier's idea in 1807 that periodic functions could be represented as a weighted sum of sines and cosines was met with skepticism.

Chapter 4
Image Enhancement in the Frequency Domain

4.2.1 The One-Dimensional Fourier Transform and its Inverse

The Fourier transform, $F(u)$, of a single variable, continuous function, $f(x)$, is defined by the equation

$$F(u) = \int_{-\infty}^{\infty} f(x)e^{-j2\pi ux}\,dx \tag{4.2-1}$$

where $j = \sqrt{-1}$. Conversely, given $F(u)$, we can obtain $f(x)$ by means of the *inverse* Fourier transform

$$f(x) = \int_{-\infty}^{\infty} F(u)e^{j2\pi ux}\,du. \tag{4.2-2}$$

These two equations comprise the *Fourier transform pair*. They indicate the important fact mentioned in the previous section that a function can be recovered from its transform. These equations are easily extended to two variables, u and v:

Chapter 4
Image Enhancement in the Frequency Domain

From the Continuous Fourier to the Discrete-time Fourier Transform

The frequency domain representation of continuous signals is given by

$$X(\omega) = \int_{-\infty}^{\infty} x(t)e^{-j\omega t}dt$$

If we consider a sampled signal $x_s(t)$, that is

$$x_s(t) = \sum_{n=-\infty}^{\infty} x(nT)\delta(t - nT)$$

then its F.T. is

$$X_s(\omega) = \int_{-\infty}^{\infty} \sum_{n=-\infty}^{\infty} x(nT)\delta(t - nT)e^{-j\omega t}dt$$

$$X_s(\omega) = \sum_{n=-\infty}^{\infty} x(nT)e^{-j\omega nT}$$

Chapter 4
Image Enhancement in the Frequency Domain

Notes on the DFT

The DFT transform is an exact one-to-one transform

The DFT can only approximate the continuous Fourier Transform

The DFT components correspond to N frequencies that are fs/N apart

The DFT of a real-valued signal gives symmetric frequency components

A fast algorithm, the FFT, is available for implementing the DFT

The FFT has several applications in spectral analysis, speech analysis-synthesis, fast convolution, etc

Chapter 4
Image Enhancement in the Frequency Domain

Frequency resolution of the DFT

The frequency resolution of the N-point DFT is

$$f_r = \frac{f_s}{N}$$

- The DFT can resolve exactly only the frequencies falling exactly at: k fs/N. There is spectral leakage for components falling between the DFT bins
- Typically we use an FFT that is as large as we can afford
- Zero-padding is often use to provide more resolution in the frequency components
- Zero padding is often combined with tapered windows

Chapter 4
Image Enhancement in the Frequency Domain

Extension of the one-dimensional discrete Fourier transform and its inverse to two dimensions is straightforward. The discrete Fourier transform of a function (image) $f(x, y)$ of size $M \times N$ is given by the equation

$$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)}. \qquad (4.2\text{-}16)$$

As in the 1-D case, this expression must be computed for values of $u = 0, 1, 2, \dots, M - 1$, and also for $v = 0, 1, 2, \dots, N - 1$. Similarly, given $F(u, v)$, we obtain $f(x, y)$ via the *inverse* Fourier transform, given by the expression

$$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M + vy/N)} \qquad (4.2\text{-}17)$$

for $x = 0, 1, 2, \dots, M - 1$ and $y = 0, 1, 2, \dots, N - 1$. Equations (4.2-16) and (4.2-17) comprise the *two-dimensional, discrete Fourier transform (DFT) pair.*

Chapter 4
Image Enhancement in the Frequency Domain

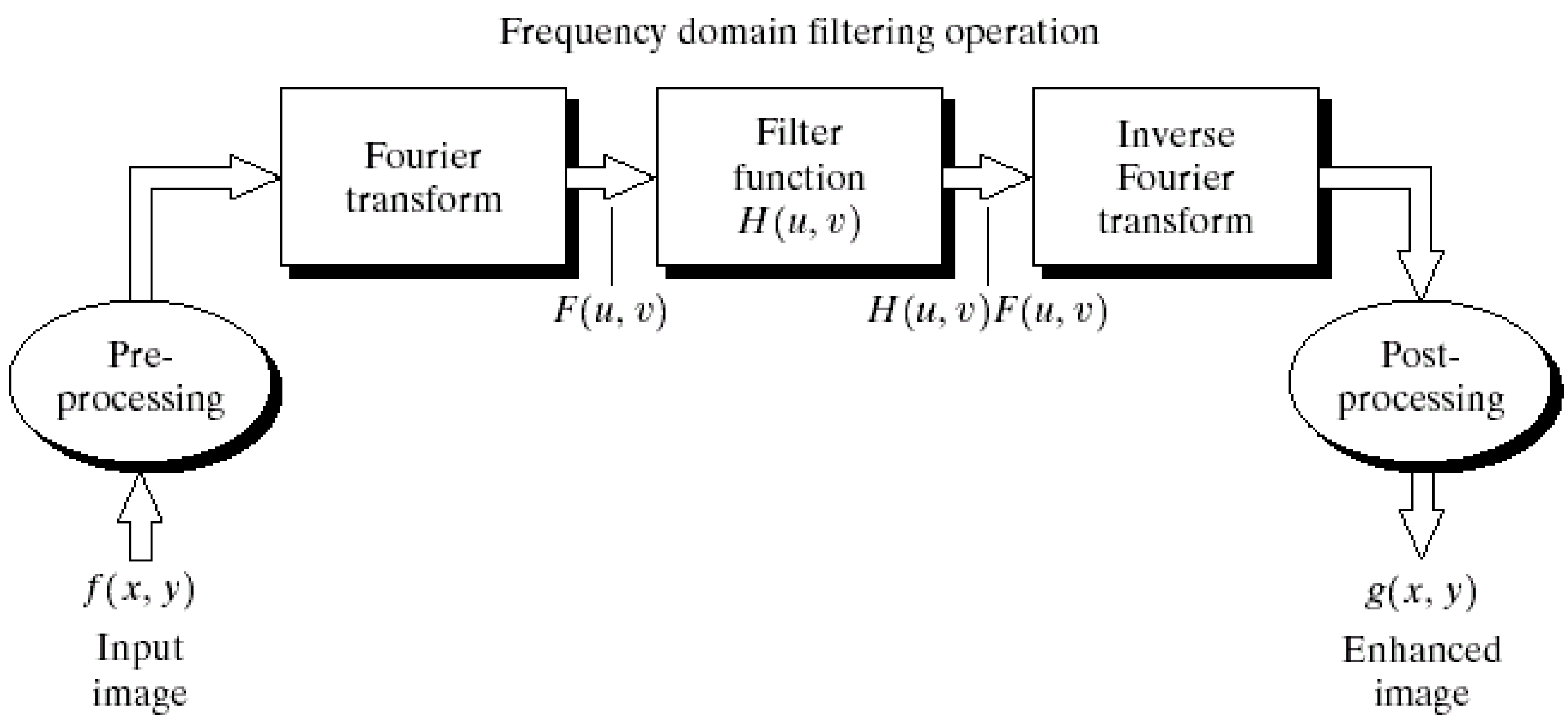


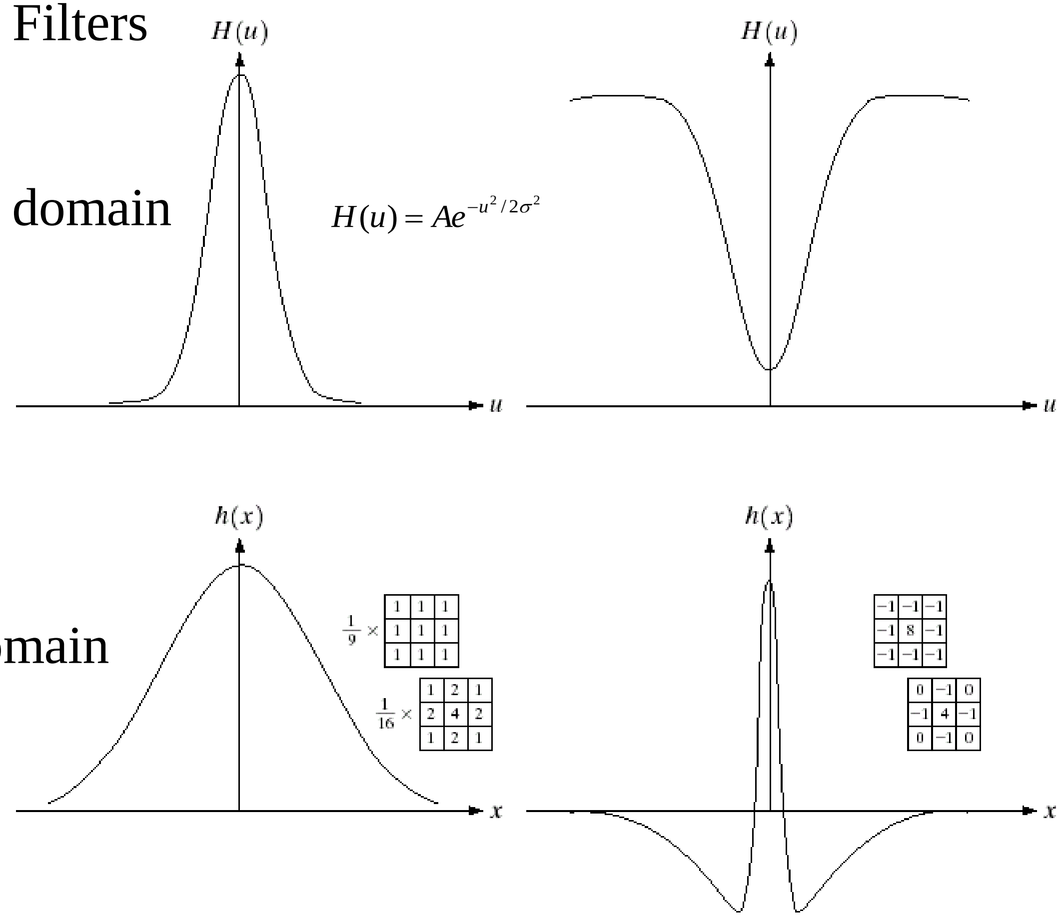
FIGURE 4.5 Basic steps for filtering in the frequency domain.

Chapter 4
Image Enhancement in the Frequency Domain

3. Gaussian Filters

frequency domain

spatial domain



$$H(u) = Ae^{-u^2/2\sigma^2}$$

Low-pass high-pass

a b
c d

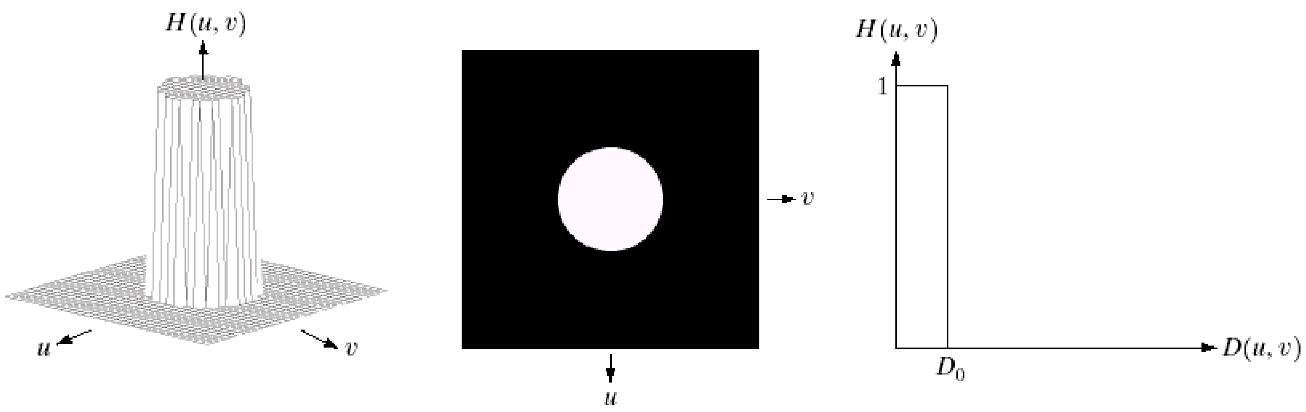
FIGURE 4.9 (a) Gaussian frequency domain lowpass filter. (b) Gaussian frequency domain highpass filter. (c) Corresponding lowpass spatial filter. (d) Corresponding highpass spatial filter. The masks shown are used in Chapter 3 for lowpass and highpass filtering.

Chapter 4
Image Enhancement in the Frequency Domain

4. Ideal low-pass filter

$$H(u,v)=\begin{cases}1 & \text{if } D(u,v)\le D_0\\ 0 & \text{if } D(u,v)\succ D_0\end{cases}$$

D_0 is the cutoff frequency and D(u,v) is the distance between (u,v) and the frequency origin.



a b c

FIGURE 4.10 (a) Perspective plot of an ideal lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

Chapter 4
Image Enhancement in the Frequency Domain

H(u,v) of Ideal Low-Pass Filter (ILPF) with radius 5

input image containing 5 bright impulses

diagonal scan line through the filtered image center

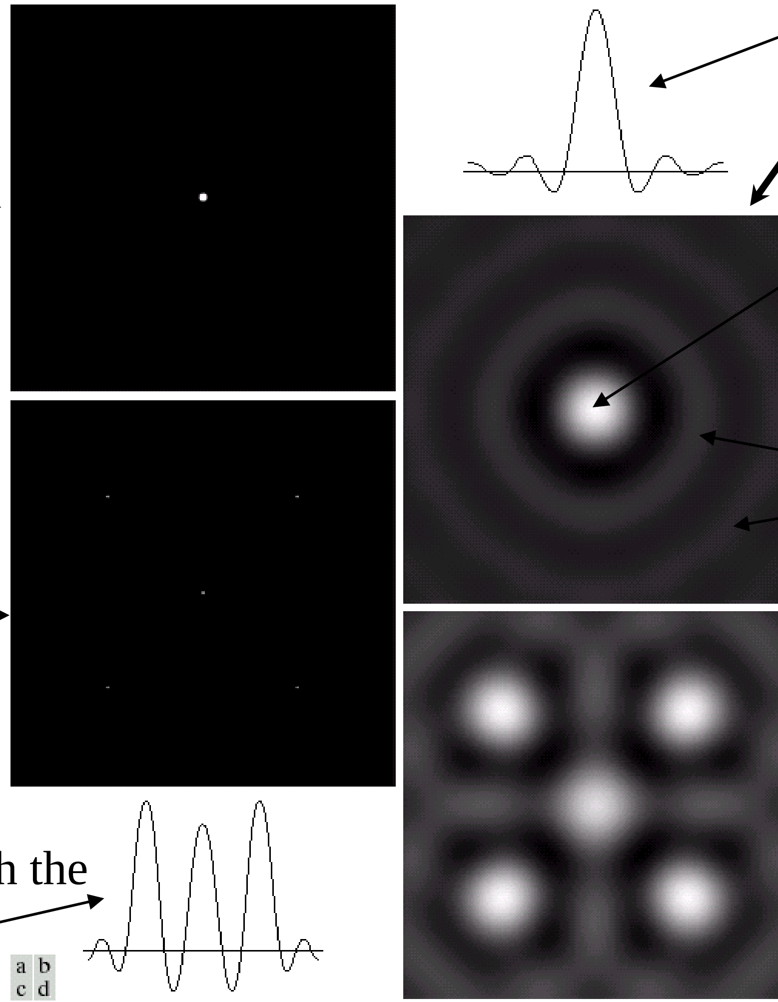


FIGURE 4.13 (a) A frequency-domain ILPF of radius 5. (b) Corresponding spatial filter (note the ringing). (c) Five impulses in the spatial domain, simulating the values of five pixels. (d) Convolution of (b) and (c) in the spatial domain.

a greylevel profile of a horizontal scan line through the center

h(x,y) is the corresponding spatial filter

the center component is responsible for blurring

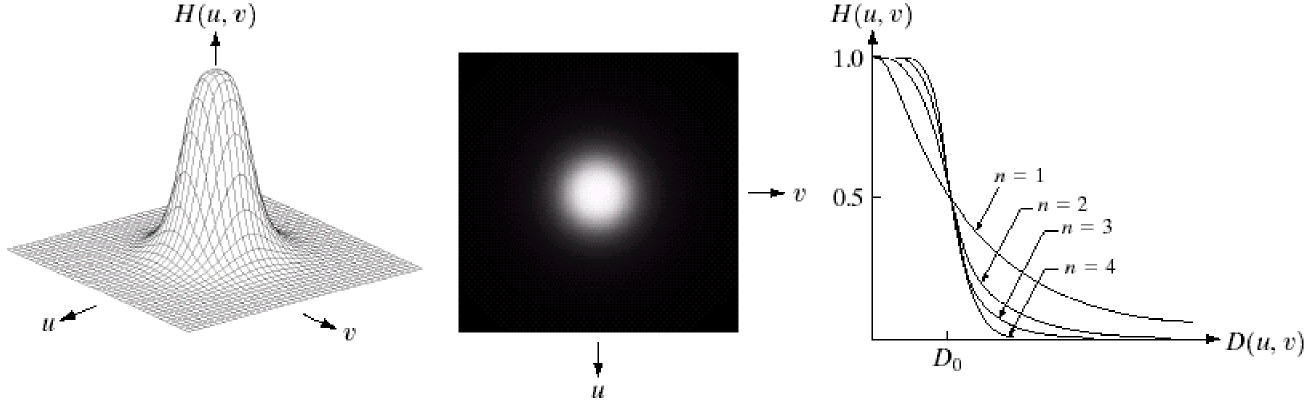
the concentric components are responbile for ringing

result of convolution of input with h(x,y)

notice blurring and ringing!

Chapter 4
Image Enhancement in the Frequency Domain

how to achieve blurring with little or no ringing? BLPF is one technique



a b c

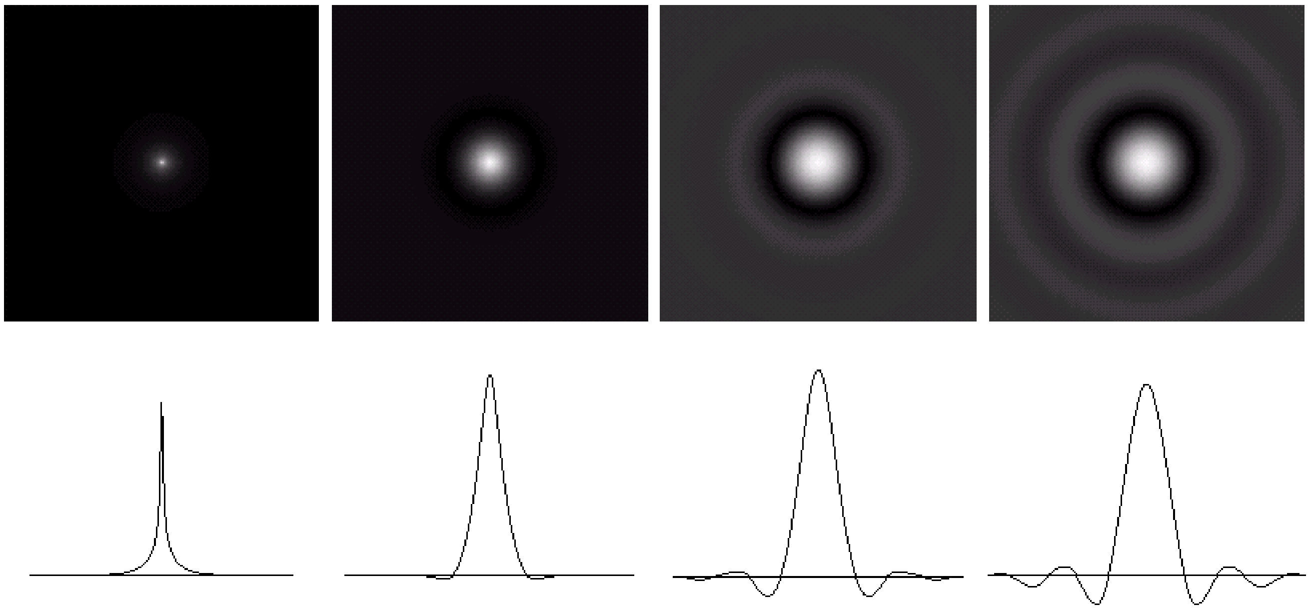
FIGURE 4.14 (a) Perspective plot of a Butterworth lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.

Transfer function of a BLPF of order n and cut-off frequency at distance D_0 (at which H(u,v) is at ½ its max value) from the origin:

$$H(u,v) = \frac{1}{1+[D(u,v)/D_0]^{2n}} \quad \text{where} \quad D(u,v) = [(u-M/2)^2 + (v-N/2)^2]^{1/2}$$

D(u,v) is just the distance from point *(u,v)* to the center of the FT

Chapter 4
Image Enhancement in the Frequency Domain



a b c d

FIGURE 4.16 (a)–(d) Spatial representation of BLPFs of order 1, 2, 5, and 20, and corresponding gray-level profiles through the center of the filters (all filters have a cutoff frequency of 5). Note that ringing increases as a function of filter order.

no ringing for n=1, imperceptible ringing for n=2, ringing increases for higher orders (getting closer to Ideal LPF).

Chapter 4
Image Enhancement in the Frequency Domain

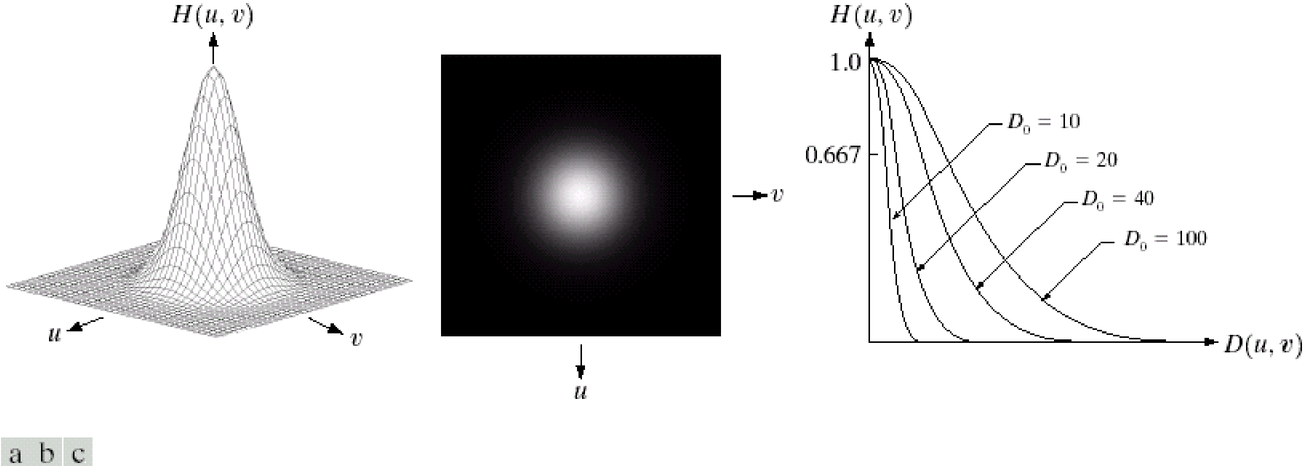


FIGURE 4.17 (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of D_0.

The 2-D Gaussian low-pass filter (GLPF) has this form:

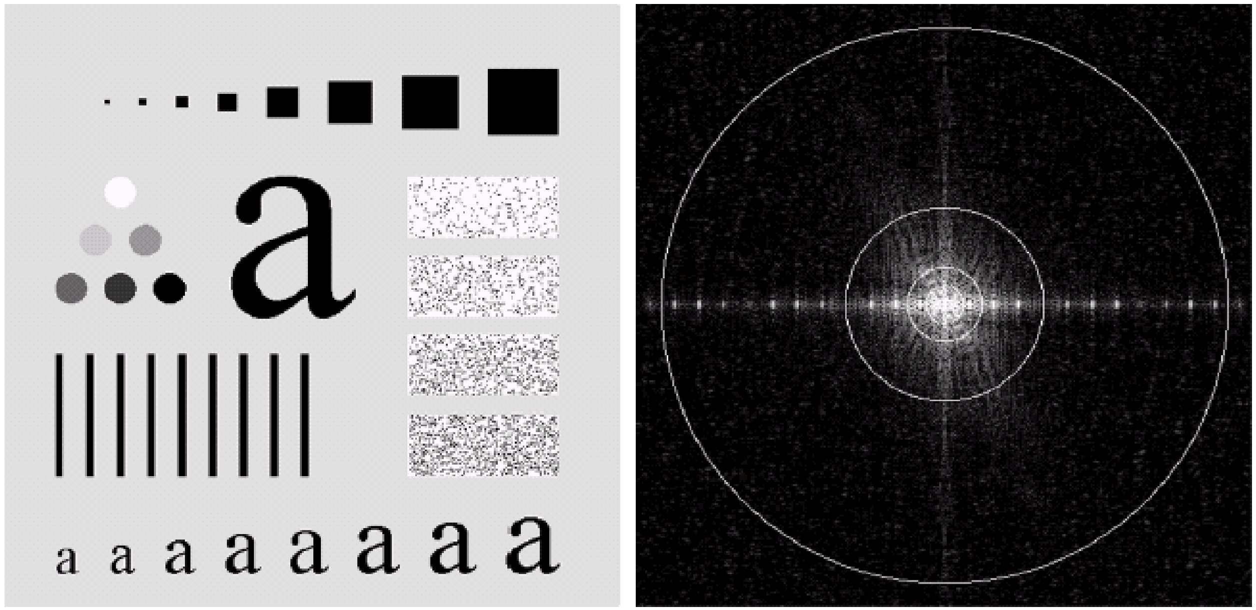
$$H(u,v) = e^{-D^2(u,v)/2\sigma^2} \qquad D_0 = \sigma$$

σ is a measure of the spread of the Gaussian curve

recall that the inverse FT of the GLPF is also Gaussian, i.e. it has no ringing!

at the cutoff frequency D_0, H(u,v) decreases to 2/3 of its max value.

Chapter 4
Image Enhancement in the Frequency Domain



a b

FIGURE 4.11 (a) An image of size 500 × 500 pixels and (b) its Fourier spectrum. The superimposed circles have radii values of 5, 15, 30, 80, and 230, which enclose 92.0, 94.6, 96.4, 98.0, and 99.5% of the image power, respectively.

- note the concentration of image energy inside the inner circle.
- what happens if we low-pass filter it with cut-off freq. at the position of these circles? (see next slide)

Chapter 4
Image Enhancement in the Frequency Domain

Results of GLPFs

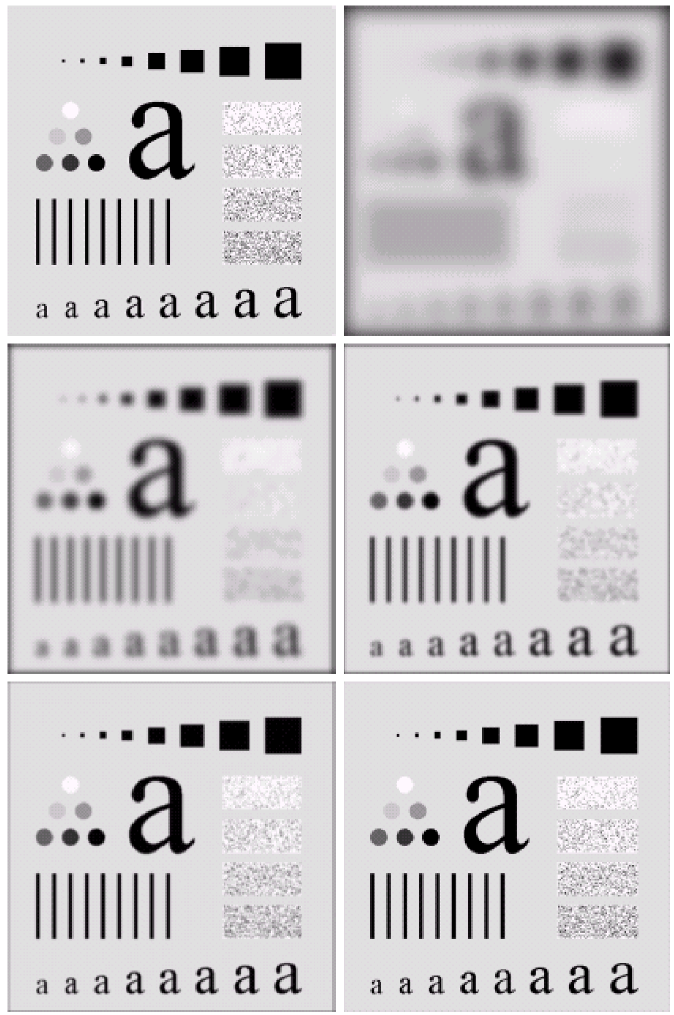


FIGURE 4.18 (a) Original image. (b)–(f) Results of filtering with Gaussian lowpass filters with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Figs. 4.12 and 4.15.

a b
c d
e f

Remarks:

1. Note the smooth transition in blurring achieved as a function of increasing cutoff frequency.

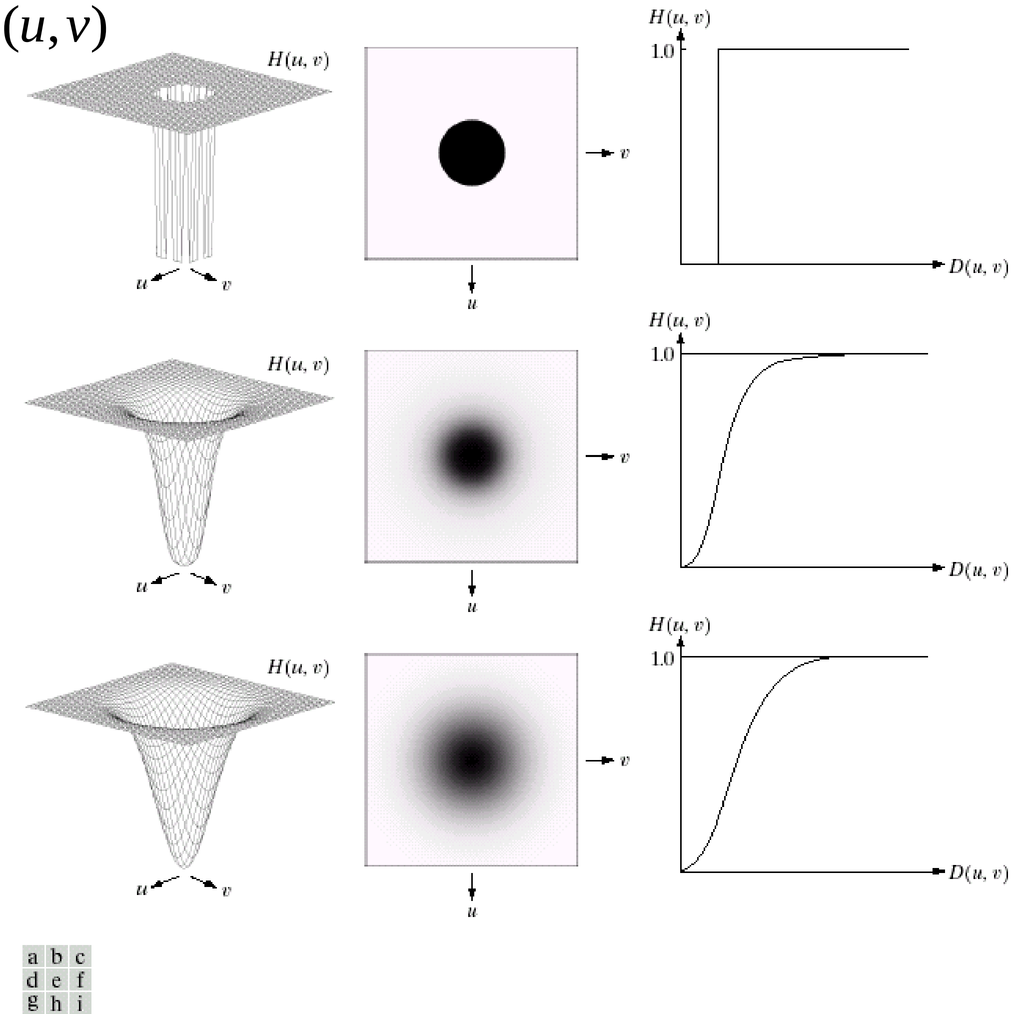
2. Less smoothing than BLPFs since the latter have tighter control over the transitions bet low and high frequencies.

The price paid for tighter control by using BLP is possible ringing.

3. No ringing!

Chapter 4
Image Enhancement in the Frequency Domain
Sharpening Frequency Domain Filters

$$H_{hp}(u,v) = 1 - H_{lp}(u,v)$$



ideal high-pass filter

Butterworth high-pass

Gaussian high-pass

FIGURE 4.22 Top row: Perspective plot, image representation, and cross section of a typical ideal highpass filter. Middle and bottom rows: The same sequence for typical Butterworth and Gaussian highpass filters.

Chapter 4
Image Enhancement in the Frequency Domain

Laplacian in the frequency domain

one can show that:

$$FT\left[\frac{d^n f(x)}{dx^n}\right] = (ju)^n F(u)$$

From this, it follows that:

$$FT\left[\frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}\right] = FT\left[\nabla^2 f(x,y)\right] = -(u^2+v^2)F(u,v)$$

Therefore, the Laplacian can be implemented in frequency by:

$$H(u,v) = -(u^2+v^2)$$

Recall that F(u,v) is centered if $F(u,v) = FT\left[(-1)^{x+y} f(x,y)\right]$

and thus the center of the filter must be shifted, i.e.

$$H(u,v) = -\left[(u-M/2)^2 + (v-N/2)^2\right]$$

Chapter 4
Image Enhancement in the Frequency Domain: **Homomorphic Filtering**

Recall that the image is formed through the multiplicative illumination-reflectance process:

$$f(x,y) = i(x,y)\, r(x,y)$$

where *i(x,y)* is the illumination and *r(x,y)* is the reflectance component

Question: how can we operate on the frequency components of illumination and reflectance?

Recall that: $$FT[f(x,y)] \neq FT[i(x,y)]\, FT[r(x,y)]$$

Let's make this transformation:

$$z(x,y) = \ln(f(x,y)) = \ln(i(x,y)) + \ln(r(x,y))$$

Then

$$FT[z(x,y)] = FT[\ln(f(x,y))] = FT[\ln(i(x,y))] + FT[\ln(r(x,y))] \quad or$$

$$Z(u,v) = F_i(u,v) + F_r(u,v)$$

Z(u,v) can then be filtered by a *H(u,v)*, i.e.

$$S(u,v) = H(u,v)Z(u,v) = H(u,v)F_i(u,v) + H(u,v)F_r(u,v)$$

Chapter 4 Image Enhancement in the Frequency Domain: **Homomorphic Filtering**

$$\begin{aligned} s(x, y) &= \Im^{-1}\{S(u, v)\} \\ &= \Im^{-1}\{H(u, v)F_i(u, v)\} + \Im^{-1}\{H(u, v)F_r(u, v)\}. \end{aligned} \tag{4.5-6}$$

By letting

$$i'(x, y) = \Im^{-1}\{H(u, v)F_i(u, v)\} \tag{4.5-7}$$

and

$$r'(x, y) = \Im^{-1}\{H(u, v)F_r(u, v)\}, \tag{4.5-8}$$

Eq. (4.5-6) can be expressed in the form

$$s(x, y) = i'(x, y) + r'(x, y). \tag{4.5-9}$$

Finally, as $z(x, y)$ was formed by taking the logarithm of the original image $f(x, y)$, the inverse (exponential) operation yields the desired enhanced image, denoted by $g(x, y)$; that is,

$$\begin{aligned} g(x, y) &= e^{s(x, y)} \\ &= e^{i'(x, y)} \cdot e^{r'(x, y)} \\ &= i_0(x, y)r_0(x, y) \end{aligned} \tag{4.5-10}$$

where

$$r_0(x, y) = e^{r'(x, y)} \qquad i_0(x, y) = e^{i'(x, y)} \tag{4.5-11}$$

Chapter 4
Image Enhancement in the Frequency Domain: Homomorphic filtering

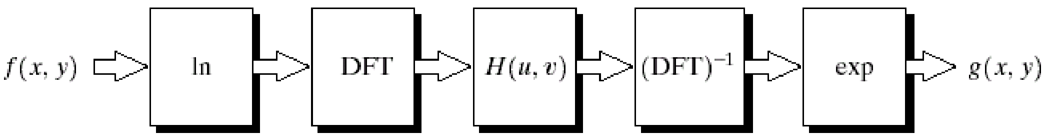


FIGURE 4.31 Homomorphic filtering approach for image enhancement.

if the gain of H(u,v) is set such as

$$\gamma_L \prec 1 \quad \text{and} \quad \gamma_H \succ 1$$

then H(u,v) tends to decrease the contribution of low-freq (illum) and amplify high freq (refl)

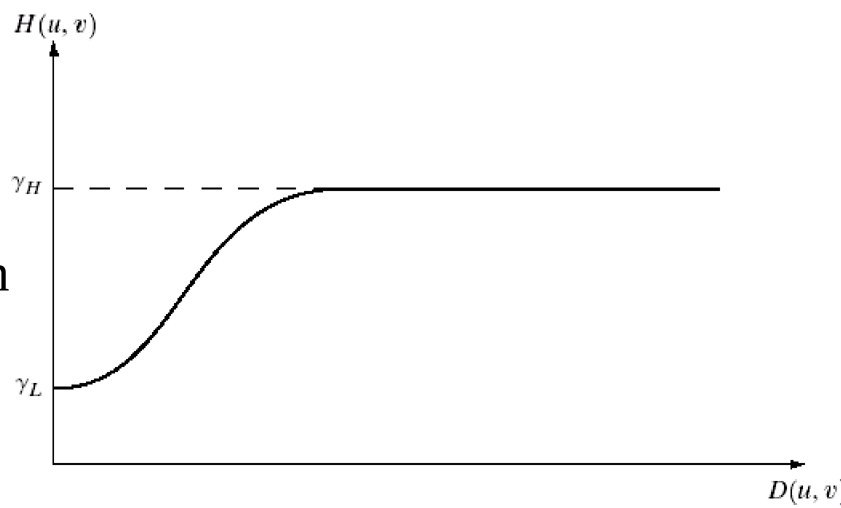


FIGURE 4.32 Cross section of a circularly symmetric filter function. $D(u, v)$ is the distance from the origin of the centered transform.

Net result: simultaneous dynamic range compression and contrast enhancement

Chapter 5: Image Filtering and Restoration

- Image degradation and restoration model
- Common noise densities (Gaussian, uniform, exponential, salt&pepper, periodic)
- Noise parameter estimation
- Restoration in the presence of noise (arithmetic, geometric, harmonic, contraharmonic, median, min, max, mid-point, alpha-trimmed mean and simple adaptive filters)
- Linear position invariant degradations
- Inverse filtering
- Wiener filtering

Chapter 5
Image Restoration

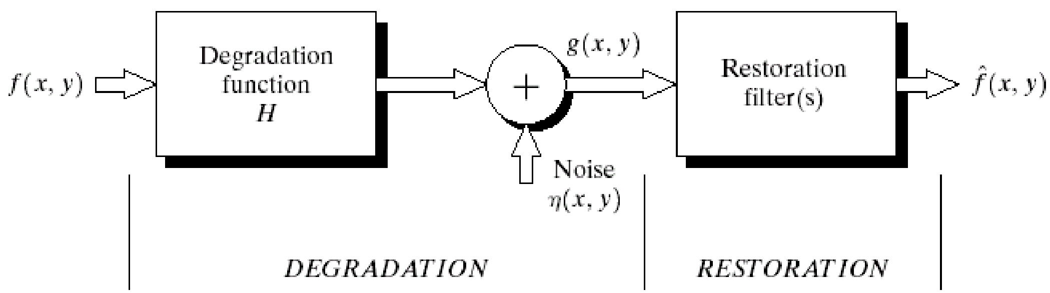


FIGURE 5.1 A model of the image degradation/restoration process.

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

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

Goal of restoration:
Find the restoration filter such that

$$\hat{f}(x,y) \textit{ is as close to } f(x,y) \textit{ as possible}$$

Mean Filters

1. Arithmetic mean filter

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

- which can be implemented with a convolution mask in which all coefficients have equal values of *1/mn*. S_{xy} represents the set of coordinates in a rectangular subimage window of size *m* x *n*, centered at (*x,y*).
- Effects: smoothes local variations in an image and noise is reduced as a result of blurring.

Order Statistics Filters

Based on ranking the input samples in a local window and selecting one of them as the output.

1. Median filter

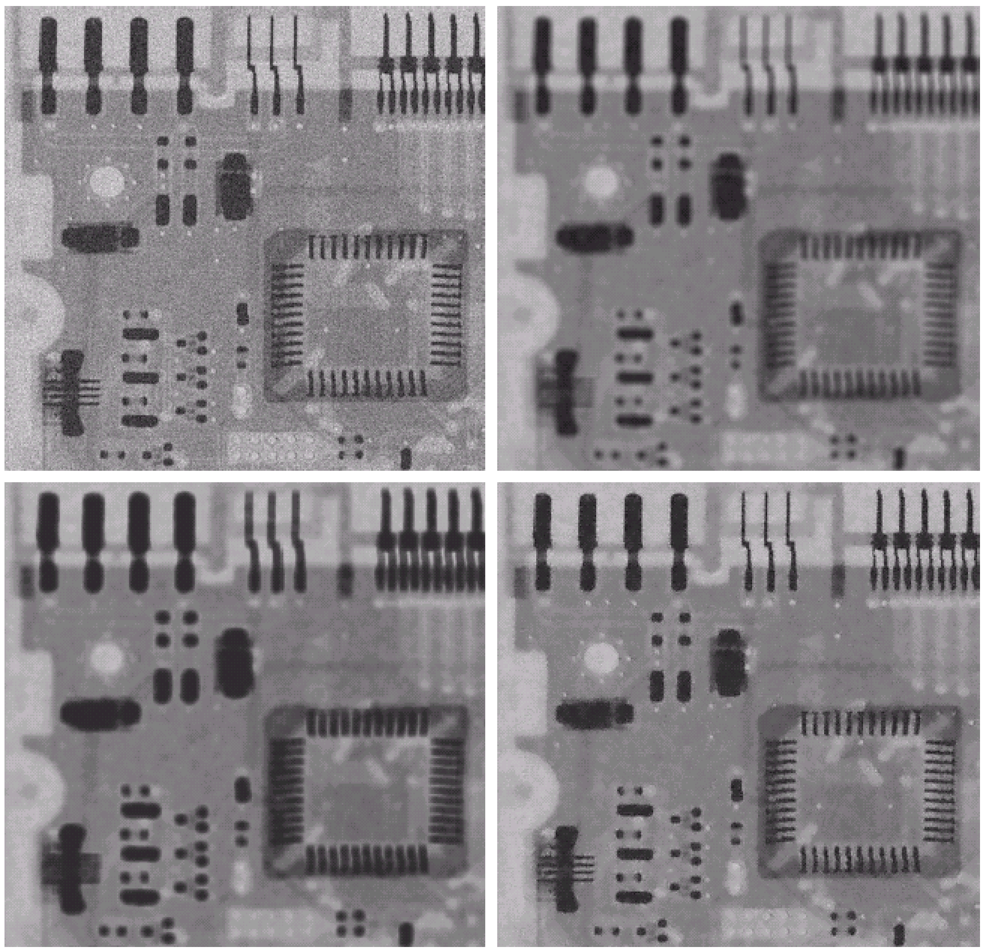
$$f(x,y) = \underset{(s,t)\in S_{xy}}{median}\{g(s,t)\}$$

- the filter output is the median value of the input data inside the filter window.
- **Effects**: for certain types of noises, it produces excellent results with considerably less blurring than linear filters.

Chapter 5
Image Restoration: **adaptive filtering**

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

σ_L^2 is the local variance of the pixels
σ_η^2 is the variance of the noise corrupting *f(x,y)*
m_L is the local mean



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.

Three cases:

1. Noise variance is *0* (no noise)
   No filtering should be done!
2. Local variance is high relative to image variance
   This indicates presence of details and thus a value close to g(x,y) should be returned.
3. Two variances are equal
   Reduce noise by averaging.

Chapter 5
Image Restoration: adaptive median

Consider the following notation:

$$z_{\min} = \text{minimum gray level value in } S_{xy}$$
$$z_{\max} = \text{maximum gray level value in } S_{xy}$$
$$z_{\text{med}} = \text{median of gray levels in } S_{xy}$$
$$z_{xy} = \text{gray level at coordinates } (x, y)$$
$$S_{\max} = \text{maximum allowed size of } S_{xy}.$$

The adaptive median filtering algorithm works in two levels, denoted level A and level B, as follows:

Level A:
- $A1 = z_{\text{med}} - z_{\min}$
- $A2 = z_{\text{med}} - z_{\max}$
- If $A1 > 0$ AND $A2 < 0$, Go to level B
- Else increase the window size
- If window size $\leq S_{\max}$ 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}.

Chapter 5
Image Restoration: Optimum notch filtering

Objective: optimal filtering by minimizing the local variances

IDEA: Isolate the principle contributions of the interference pattern and then subtract a variable, weighted portion of the pattern from the corrupted image.

Let *H(u,v)* be a notchpass filter placed at the location of each spike. Then the interference noise pattern is :

$$\boxed{N(u,v)=H(u,v)G(u,v)} \quad \text{Eq. (1)}$$

where G(u,v) is the FT of the corrupted image.

H(u,v) is iteratively selected by observing *G(u,v)* on a display.

From Eq. (1) it follows that: $\eta(x,y) = FT^{-1}\{H(u,v)G(u,v)\}$

Let

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

where $\hat{f}(x,y)$ is the estimate of *f(x,y)* and *w(x,y)* is a weighting function to be determined.

Chapter 5
Image Restoration: Linear, Position-invariant Degradations

Recall the image degration/restoration process

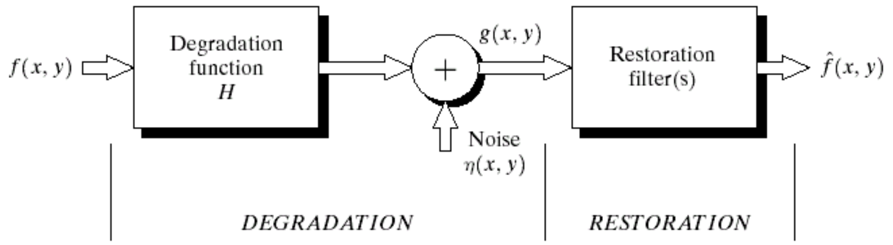


FIGURE 5.1 A model of the image degradation/restoration process.

Before the restoration stage, we have

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

Assume that the noise term is absent and *H* is linear, i.e.

$$H[af + bg] = aH[f] + bH[g]$$

that is *H* is *additive* and *homogenous*

Chapter 5
Image Restoration: **Inverse Filtering**

Assume that we have estimated *H* using any of the previous techniques. Let's try now to restore the image.

In the absence of any information concerning noise, we get

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

Problem: even if we know *H*, still cannot recover *f(x,y)* because we don't know *N(.,.)!*
More problems: what happens when *H* is zero or has very small values? The second part may dominate the restored image!

Can get around this by limiting the filter frequencies to values near the origin, i.e. **pseudo-inverse filtering**, see example next.

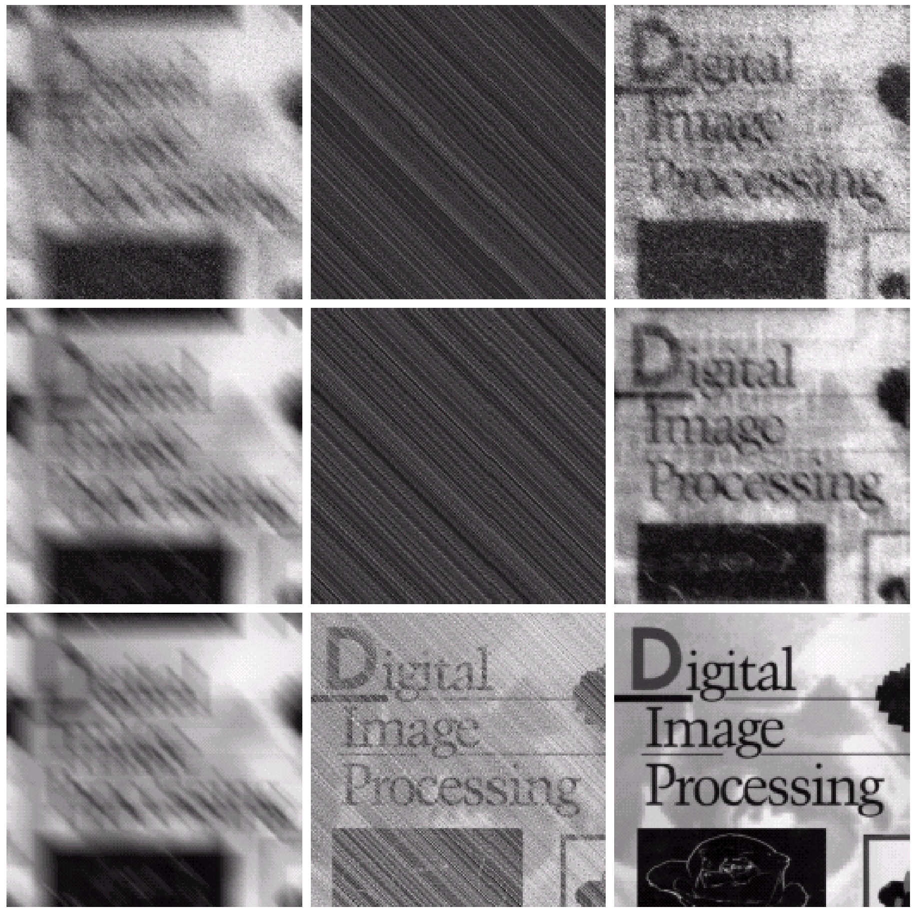
(a) image corrupted with motion blur and additive noise

(b) result of inverse filtering

(c) result of Wiener filtering

(d)-(f) same sequence but with noise variance one order of magnitude less

(g)-(j) same sequence but noise variance reduced by five orders of magnitude from (a)



a b c
d e f
g h i

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

Chapter 5
Image Restoration: **Wiener Filtering**

Remark:
Inverse and pseudo-inverse filtering reverse the effects of system only; but can do nothing about the random noise in the signal.
Alternative Solution: Wiener Filtering
Wiener filtering has been successfully used to filter images corrupted by noise and blurring. The idea of Wiener filtering is to find the "best" estimate of the true input $x(m,n)$ from the observed image $y(m,n)$ by modeling the input and output images as random sequences.
"best" in the mean square error sense.

*** In mathematics, Wiener deconvolution is an application of the Wiener filter to the noise problems inherent in deconvolution. It works in the frequency domain, attempting to minimize the impact of deconvoluted noise at frequencies which have a poor signal to noise ratio.**
*** The Wiener deconvolution method has widespread use in image deconvolution applications, as the frequency spectrum of most visual images is fairly well behaved and may be estimated easily.**
Wiener deconvolution is named after Norbert Wiener.

Chapter 5
Image Restoration: **Wiener Filtering**

Given a system:

$$y(t) = h(t) * x(t) + v(t)$$

where * denotes convolution. and:

- $x(t)$ is some input signal (unknown) at time t.
- $h(t)$ is the known impulse response of a linear time-invariant system
- $v(t)$ is some unknown additive noise, independent of $x(t)$
- $y(t)$ is our observed signal

Our goal is to find some $g(t)$ so that we can estimate $x(t)$ as follows:

$$\hat{x}(t) = g(t) * y(t)$$

where $\hat{x}(t)$ is an estimate of $x(t)$ that minimises the mean square error.

The Wiener deconvolution filter provides such a $g(t)$. The filter is most easily described in the frequency domain:

$$G(f) = \frac{H^*(f)S(f)}{|H(f)|^2 S(f) + N(f)}$$

where:

- $G(f)$ and $H(f)$ are the Fourier transforms of g and h, respectively at frequency f.
- $S(f)$ is the mean power spectral density of the input signal $x(t)$
- $N(f)$ is the mean power spectral density of the noise $v(t)$
- the superscript $*$ denotes complex conjugation.

The filtering operation may either be carried out in the time-domain, as above, or in the frequency domain:

$$\hat{X}(f) = G(f)Y(f)$$

Chapter 5
Image Restoration: **Wiener Filtering**

The operation of the Wiener filter becomes apparent when the filter equation above is rewritten:

$$G(f) = \frac{1}{H(f)}\left[\frac{|H(f)|^2}{|H(f)|^2 + \frac{N(f)}{S(f)}}\right]$$
$$= \frac{1}{H(f)}\left[\frac{|H(f)|^2}{|H(f)|^2 + \frac{1}{\mathrm{SNR}(f)}}\right]$$

Here, $1/H(f)$ is the inverse of the original system, and $\mathrm{SNR}(f) = S(f)/N(f)$ is the signal-to-noise ratio. When there is zero noise (i.e. infinite signal-to-noise), the term inside the square brackets equals 1, which means that the Wiener filter is simply the inverse of the system, as we might expect. However, as the noise at certain frequencies increases, the signal-to-noise ratio drops, so the term inside the square brackets also drops. This means that the Wiener filter attenuates frequencies dependent on their signal-to-noise ratio.

The Wiener filter equation above requires us to know the spectral content of a typical image, and also that of the noise. Often, we do not have access to these exact quantities, but we may be in a situation where good estimates can be made. For instance, in the case of photographic images, the signal (the original image) typically has strong low frequencies and weak high frequencies, and in many cases the noise content will be relatively flat with frequency.

Chapter 6: Color Image Processing

- Review of colors and color models (RGB, CMYK, HSI)
- Pseudocoloring
  - intensity slicing
  - greylevel to color transformations
- Pseudocoloring multispectral images
- Full color image processing
  - Color slicing
  - Tonal and color corrections
  - Histogram equalization of color images
  - Color image smoothing
  - Color image sharpening

Chapter 6
Color Image Processing

1965 Experimental curves:

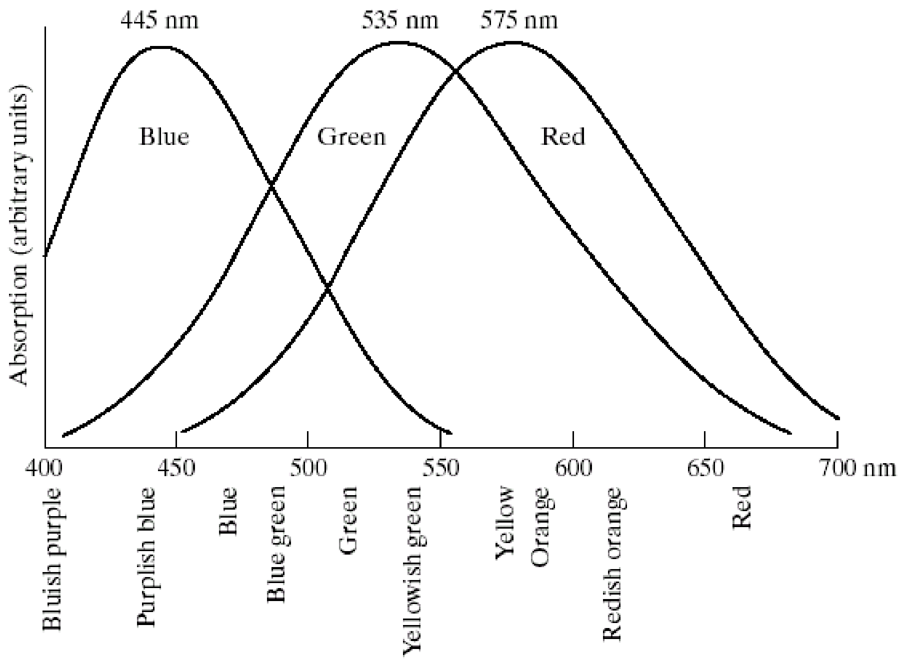


FIGURE 6.3 Absorption of light by the red, green, and blue cones in the human eye as a function of wavelength.

Due to these absorption characteristics, colors are seen as variable combinations of so called "primary" colors red, green and blue.

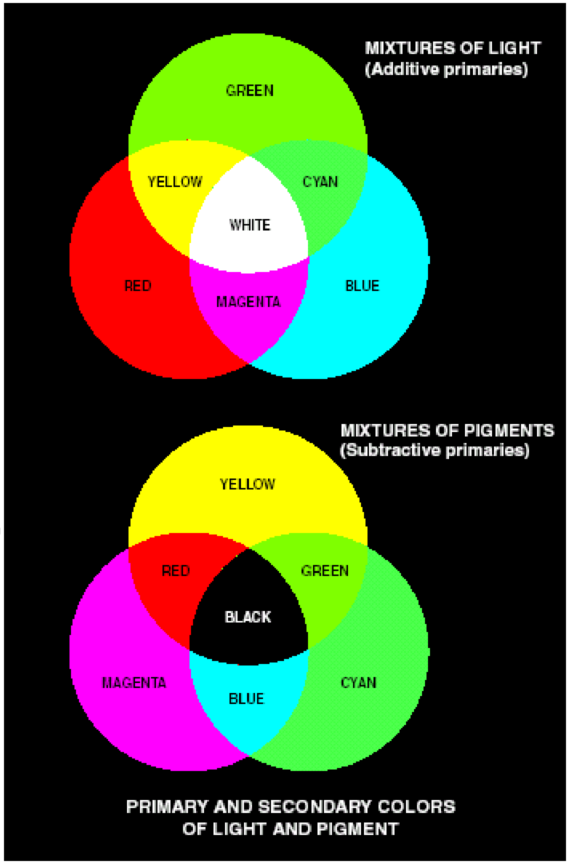
In **1931**, CIE designated the following:
Blue = 435.8nm;
Green = 546.1nm; and
Red = 700nm

- Remember that there is no single color called red, green or blue in the color spectrum!
- Also, these fixed RGB components cannot generate ALL spectrum colors!

Chapter 6
Color Image Processing

Primary colors can be added in pairs to procude secondary colors of light: e.g. magenta, cyan and yellow.
Mixing the three primaries produces white color.

A primary color of pigments or colorants is defined as one that Subtracts or absorbs a primary color of light and reflects the other two.



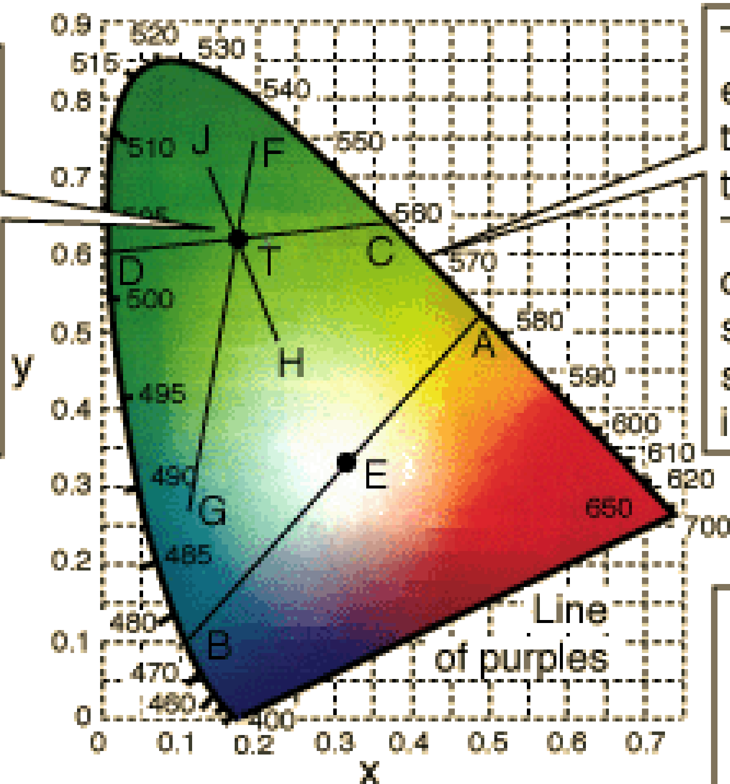
a
b

FIGURE 6.4 Primary and secondary colors of light and pigments. (Courtesy of the General Electric Co., Lamp Business Division.)

primary colors of pigments are magenta, cyan and yellow and their secondary colors are red, green and blue

Chapter 6
Color Image Processing

The combination of light wavelengths to produce a given perceived color is not unique. The pairs CD, FG and JH can each produce the color T if combined in the right proportions.



The solid line outline encompasses all the hues that are perceivable to the normal human eye. The horseshoe shaped curve contains the spectral colors. The straight line at the bottom is the line of purples.

Any point within the curve represents a unique perceivable hue. But there are many combinations that will produce that hue.

E is the achromatic point. AB or any pair for which the connecting line passes through E can form a complementary color pair.

Chapter 6
Color Image Processing: Color Models

Color models or color spaces refer to a color coordinate system in which each point represents one color.

Different models are defined (standardized) for different purposes, e.g.

Hardware oriented models:

- RGB for color monitors (CRT and LCD) and video cameras,
- CMYK (cyan, magenta, yellow and black) for color printers

Color manipulation models:

- HSI (hue, saturation and brightness) is closest to the human visual system
- Lab is most uniform color space
- YCbCr (or YUV) is often used in video where chroma is down-sampled (recall that the human visual system is much more sensitive to luminance than to color)
- XYZ is known as the raw format
- others

Two important aspects to retain about color models:

1. conversion between color models can be either linear or nonlinear,
2. some models can be more useful as they can decouple color and gray-scale components of a color image, e.g. HSI, YUV.

Chapter 6
Color Image Processing: Color Models

CMY and CMYK Color Models

Most devices that deposit color pigments on paper, e.g. printers and copiers, use CMY inputs or perform RGB to CMY conversion internally:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Recall that all color values have been normalised in the range [0,1].

Remarks:

1. Note that, e.g. a surface coated with cyan does not contain red, that is $C = 1 - R$.
2. since equal amounts of the pigment primaries should produce black.

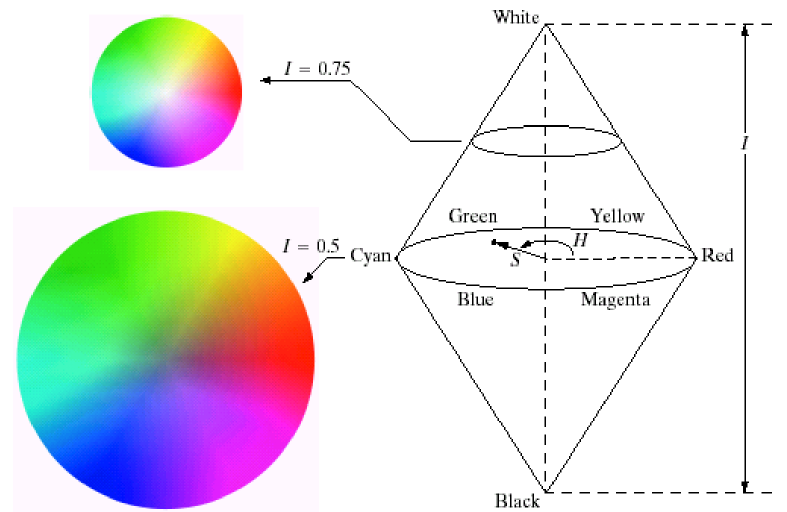
In printing this appears as muddy-looking black; therefore, a fourth color, black is added, leading to CMYK color model (four-color printing).

Chapter 6
Color Image Processing: Color Image Representation

Three Perceptual Measures

1. **Brightness**: varies along the vertical axis and measures the extent to which an area appears to exhibit light. It is proportional to the electromagnetic energy radiated by the source.

2. ***Hue****:* denoted by H and varies along the circumference. It measure the extent to which an area matches colors red, orange, yellow, blue or purple (or a mixture of any two). In other words, hue is a parameter which distinguishes the color of the source, i.e., is the color red, yellow, blue, etc.



3. **Saturation**: the quantity which distinguishes a pure spectral light from a pastel shade of the same hue. It is simply a measure of white light added to the pure spectral color. In other words, saturation is the colorfulness of an area judged in proportion to the brightness of the object itself. Saturation varies along the radial axis.

Chapter 6
Pseudo-color Image Processing

- Pseudocolor or false color image processing consists of assigning (false) colors to gray level values based on some specific criterion.
- Goal and Motivation
  - improve human visualization
    - human can distinguish at most 20-30 gray shades but thousands of colors!
  - attract attention
- Major techniques
  - intensity slicing
  - gray level to color transformation

Chapter 6
Color Image Processing: **Intensity slicing**

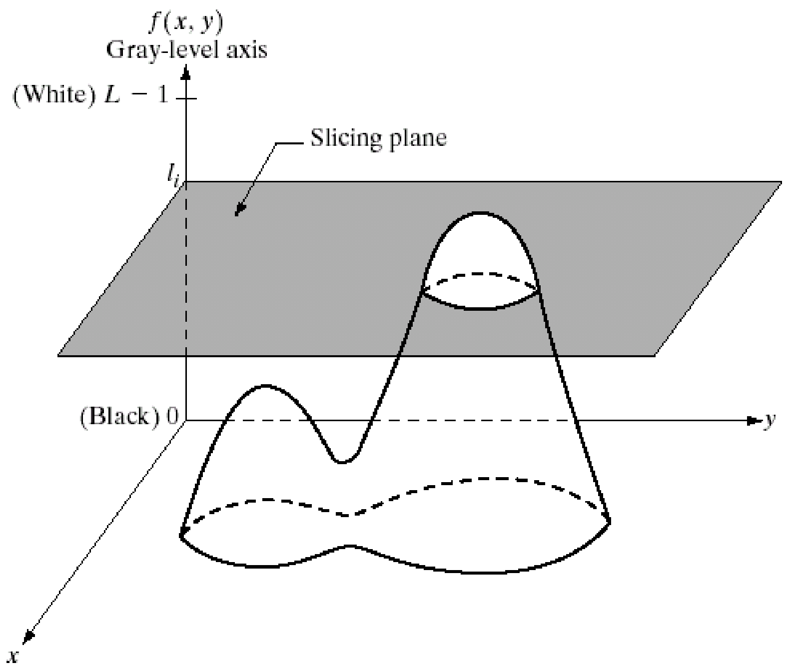


FIGURE 6.18 Geometric interpretation of the intensity-slicing technique.

assign different colors to levels above and below the slicing plane

usually, several levels are used.

Chapter 6
Color Image Processing: Gray level to color transformation

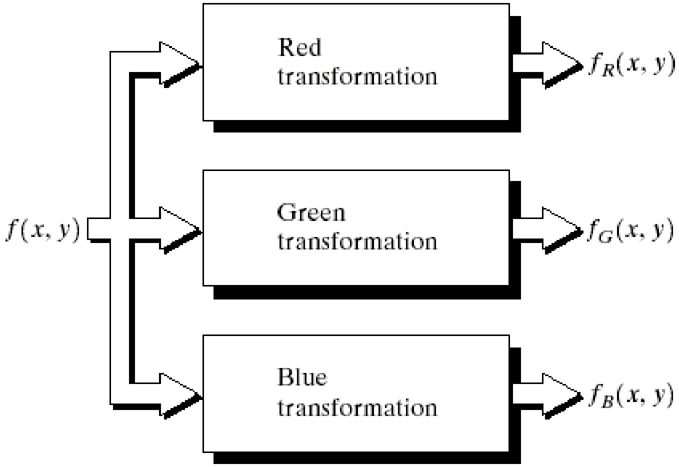


FIGURE 6.23 Functional block diagram for pseudocolor image processing. f_R, f_G, and f_B are fed into the corresponding red, green, and blue inputs of an RGB color monitor.

Chapter 6
Color Image Processing: **multi-spectral images**

Many images are multispectral, i.e. they have been acquired by different sensors at different wavelengths. Combining them to obtain a color image can be achieved as follows:

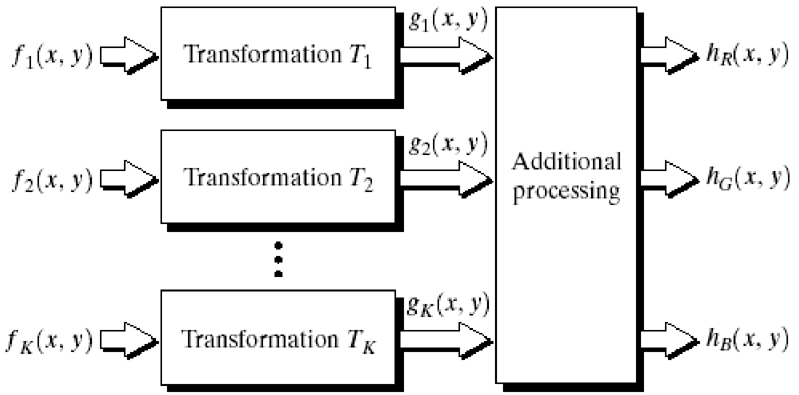


FIGURE 6.26 A pseudocolor coding approach used when several monochrome images are available.

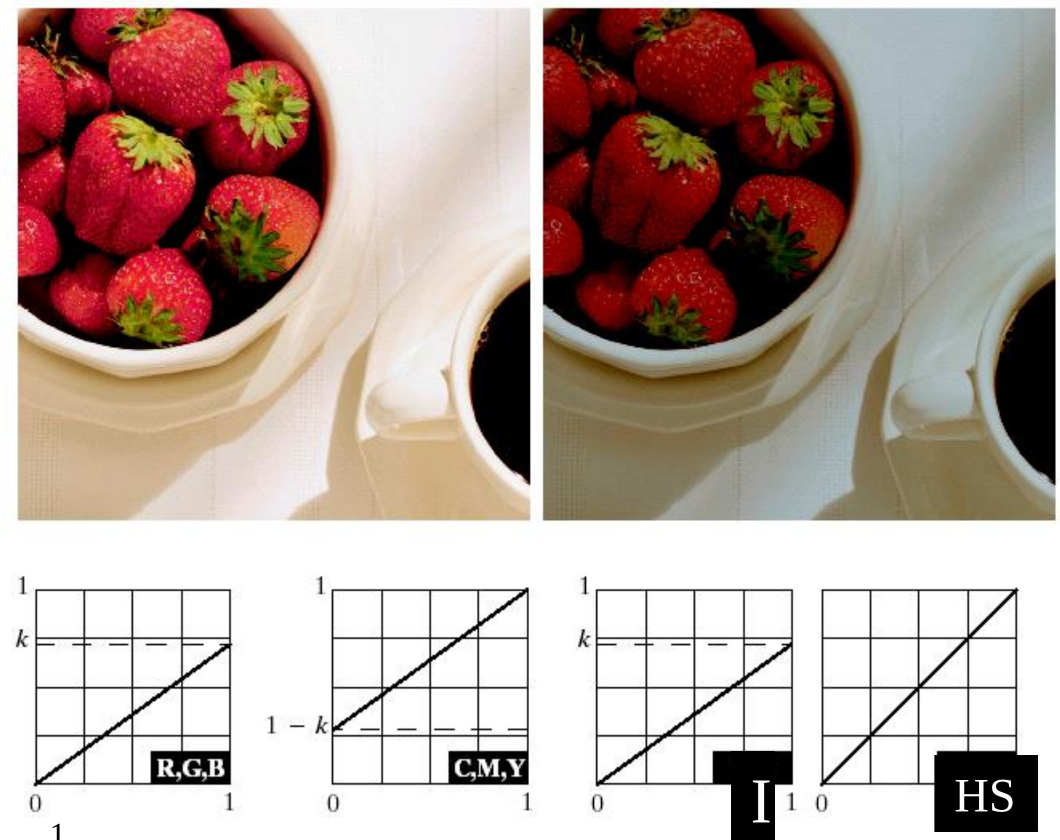
additional processing may include color balancing, combining images and selecting three of them for display, etc.

Chapter 6
Color Image Processing

Example: decrease intensity component by 30%. In RGB and CMY, must apply transformation to all components, on the other hand, in HSI, only I component is transformed.

| a | b |
| c d | e |

FIGURE 6.31 Adjusting the intensity of an image using color transformations. (a) Original image. (b) Result of decreasing its intensity by 30% (i.e., letting $k = 0.7$). (c)–(e) The required RGB, CMY, and HSI transformation functions. (Original image courtesy of MedData Interactive.)



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

Chapter 6
Color Image Processing: **color slicing**

Idea: highlight a range of colors in an image in order to

- separate them from background, or
- use the region defined by color mask for further processing, e.g. segmentation

This is a complex extension of gray level slicing due to the multi-valued nature of color images

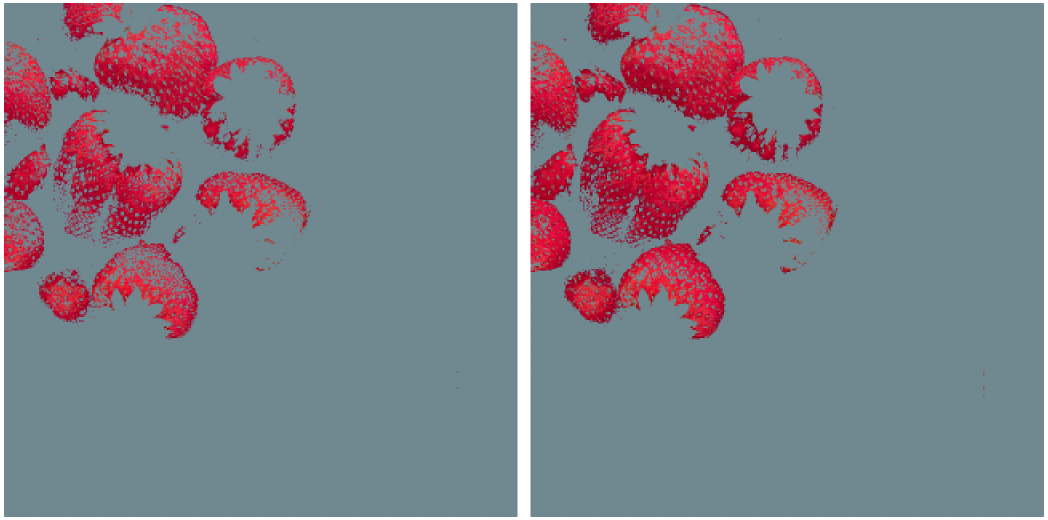
How can this be done? Can map the colors outside some range of interest to some neutral color and leave the rest as they are. Let $w=(a_1,a_2,a_3)$ be the average of the color region of interest and W the width of this region, then

$$s_i = \begin{cases} 0{,}5 & \text{if } \left[|r_j - a_j| > \dfrac{W}{2} \right]_{\forall 1 \le j \le 3} \\ r_i & \end{cases} \qquad \text{for } i=1,2,3$$

If a sphere is used to specify the region of interest, then

$$s_i = \begin{cases} 0{,}5 & \text{if } \sum_{j=1}^{3} (r_j - a_j)^2 \succ R_0^2 \\ r_i & \end{cases}$$

Chapter 6
Color Image Processing: **color slicing example**



a b

FIGURE 6.34 Color slicing transformations that detect (a) reds within an RGB cube of width $W = 0.2549$ centered at $(0.6863, 0.1608, 0.1922)$, and (b) reds within an RGB sphere of radius 0.1765 centered at the same point. Pixels outside the cube and sphere were replaced by color $(0.5, 0.5, 0.5)$.

Chapter 6
Color Image Processing: Tone and color corrections

Goal: correct color image through pixel transformations to get a better visualization and / or print out.

L*a*b* color space is perceptually uniform, i.e. color differences are perceived uniformly.

Like HSI, L*a*b* decouples intensity from color

Example: tonal correction for three common tonal imbalances: flat, light and dark images, see images next.

Chapter 6
Color Image Processing: Tonal transformations

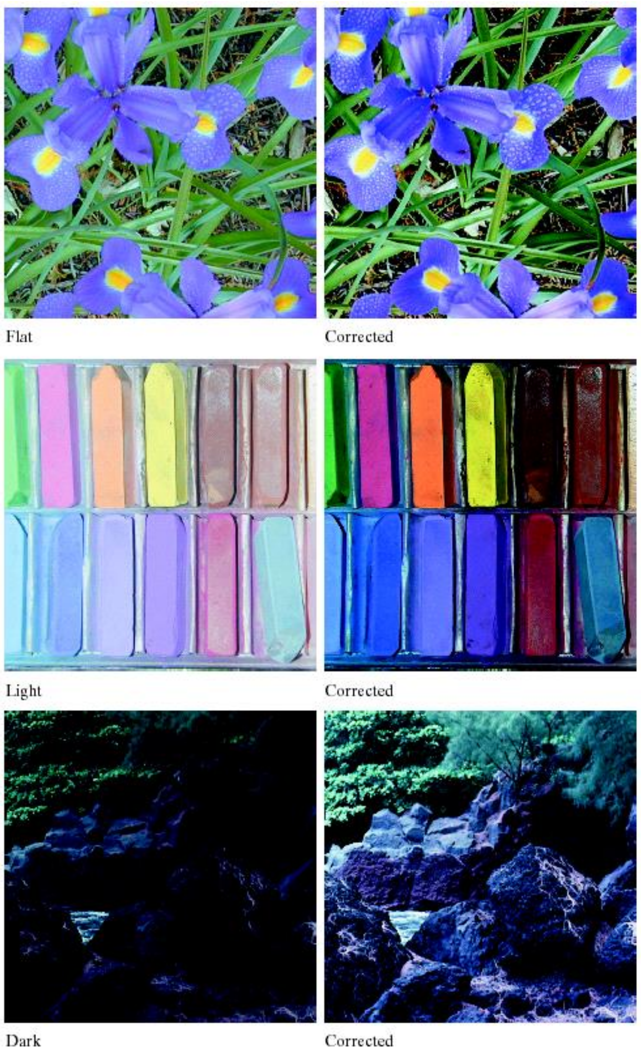
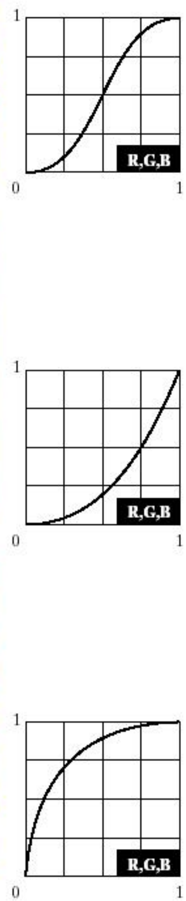


FIGURE 6.35 Tonal corrections for flat, light (high key), and dark (low key) color images. Adjusting the red, green, and blue components equally does not alter the image hues.



S-shaped transformation for boosting contrast

power-law-like transformation to correct light and dark details, as in B&W images.

Chapter 6
Color Image Processing: color balancing

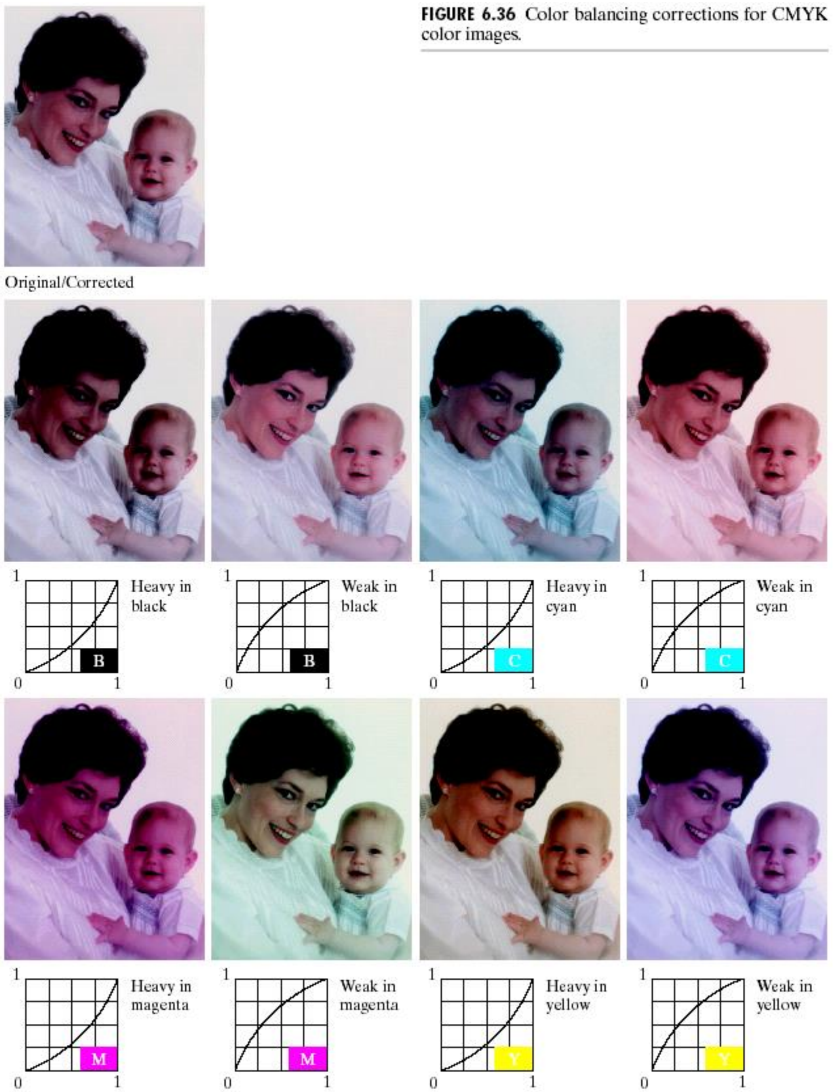


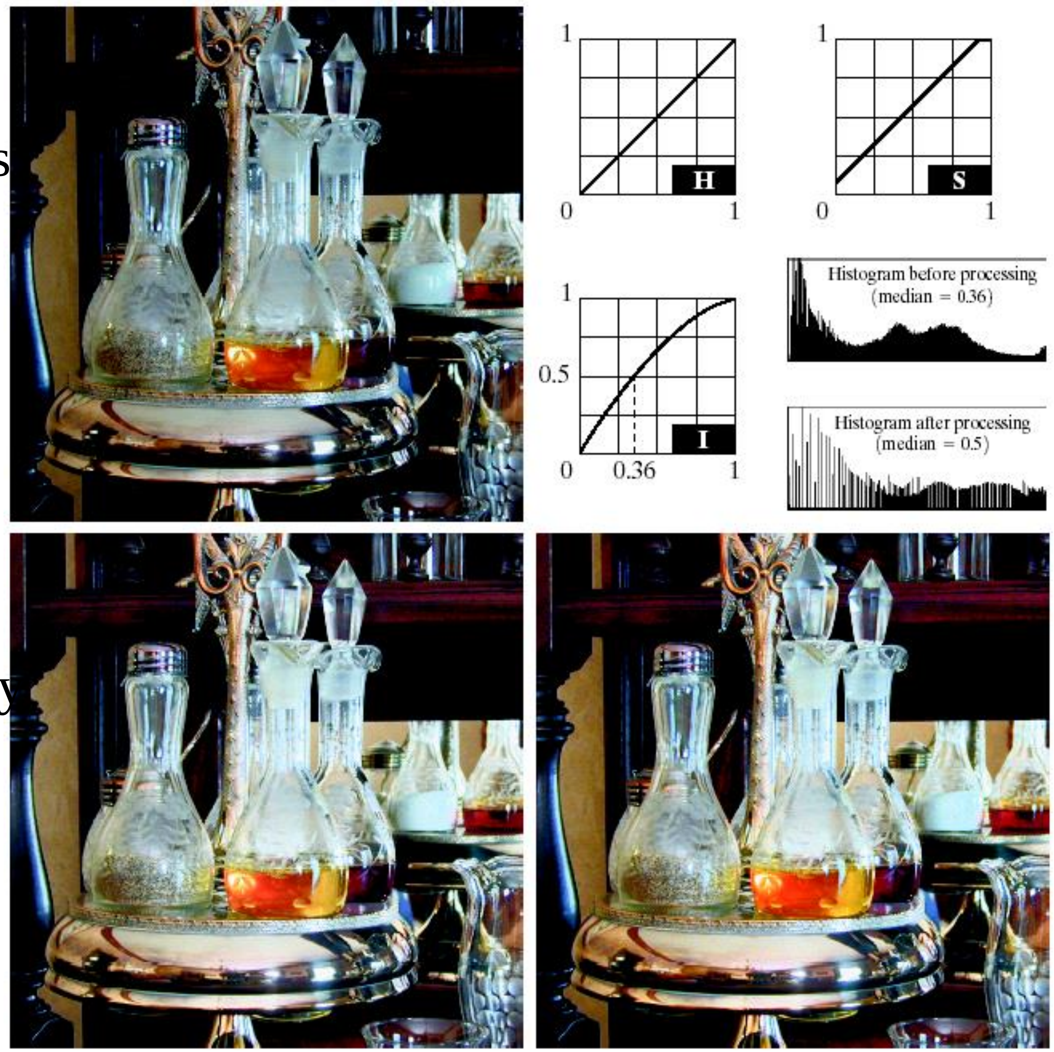
FIGURE 6.36 Color balancing corrections for CMYK color images.

Chapter 6
Color Image Processing: Histogram equalization

Q: would it be wise to equalize color components independently?

A: not so clever, this way colors change!

Solution: equalize intensity component only, e.g. in HSI color space.



a b
c d

FIGURE 6.37 Histogram equalization (followed by saturation adjustment) in the HSI color space.

Chapter 6
Color Image Processing: color image sharpening

only intensity component was processed

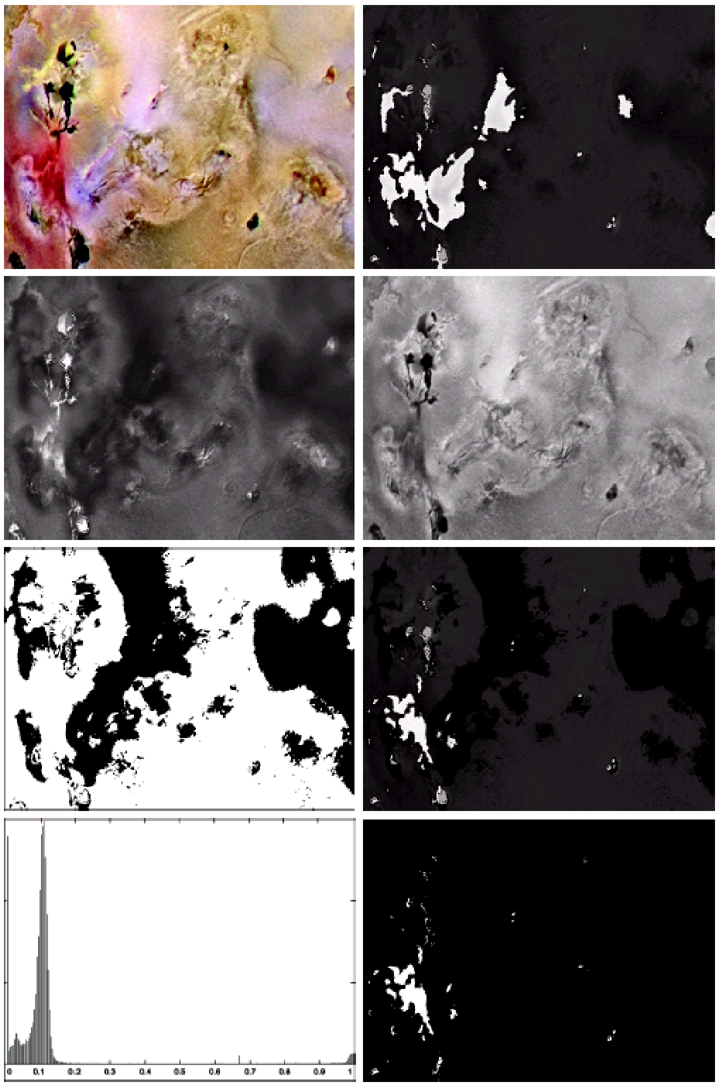
a b c

FIGURE 6.41 Image sharpening with the Laplacian. (a) Result of processing each RGB channel. (b) Result of processing the intensity component and converting to RGB. (c) Difference between the two results.

Laplacian of a color image can be computed component-wise:

$$\nabla^2[c(x,y)] = \begin{bmatrix} \nabla^2 R(x,y) \\ \nabla^2 G(x,y) \\ \nabla^2 B(x,y) \end{bmatrix}$$

Chapter 6
Color Image Processing: **segmentation**



(a) original (b) Hue

(c) saturation (d) intensity

(e) thresholding saturation (@10%)

(f) product of hue and saturation

(g) histogram of (f)

(h) segmentation of red component in (a)

a b
c d
e f
g h

FIGURE 6.42 Image segmentation in HSI space. (a) Original. (b) Hue. (c) Saturation. (d) Intensity. (e) Binary saturation mask (black = 0). (f) Product of (b) and (e). (g) Histogram of (f). (h) Segmentation of red components in (a).

Chapter 6
Color Image Processing: Noise in color images

a b
c d

FIGURE 6.48 (a)–(c) Red, green, and blue component images corrupted by additive Gaussian noise of mean 0 and variance 800. (d) Resulting RGB image. [Compare (d) with Fig. 6.46(a).]

Consider the RGB components, each was corrupted with Gaussian noise (0,800).

None of the components look very objectionable including the color image!

Chapter 07
Basics of Digital Video

Why Digital?

"Exactness"

- Exact reproduction without degradation
- Accurate duplication of processing result

Convenient & powerful computer-aided processing

- Can perform rather sophisticated processing through hardware or software

Easy storage and transmission

- 1 DVD can store a three-hour movie !!!
- Transmission of high quality video through network in reasonable time

Digital Video Coding

The basic idea is to remove redundancy in video and encode it

Perceptual redundancy

- The Human Visual System is less sensitive to color and high frequencies

Spatial redundancy

- Pixels in a neighborhood have close luminance levels
  - Low frequency

How about temporal redundancy?

- Differences between subsequent frames can be small. Shouldn't we exploit this?

Hybrid Video Coding

"Hybrid" combination of Spatial, Perceptual, & Temporal redundancy removal

Issues to be handled

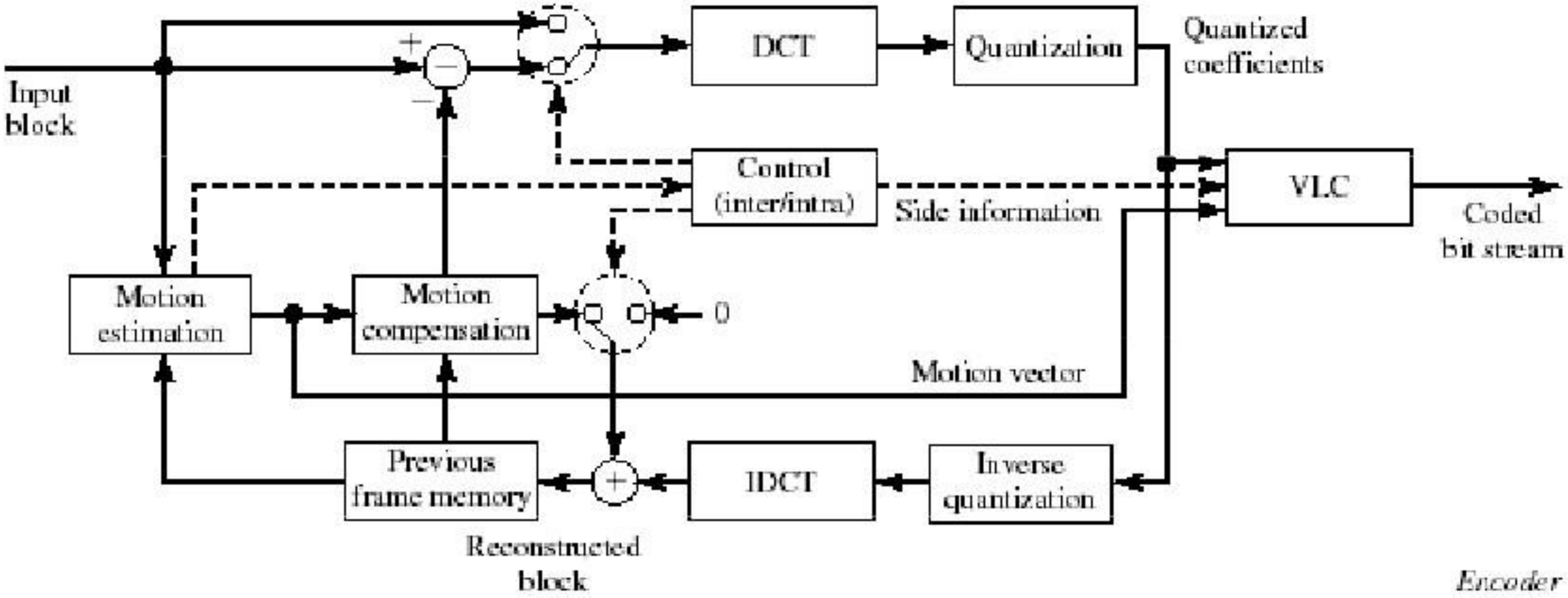
- Not all regions are easily inferable from previous frame
  - Occlusion is solved by backward prediction using future frames as reference
  - The decision of whether to use prediction or not is made adaptively
- Drifting and error propagation
  - Solved by encoding reference regions or frames at constant intervals of time
- Random access
  - Solved by encoding frame without prediction at constant intervals of time
- Bit allocation
  - according to statistics (more frequently values are encoded with fewer bits!)
  - constant and variable bit-rate requirement

MPEG combines all of these features !!!

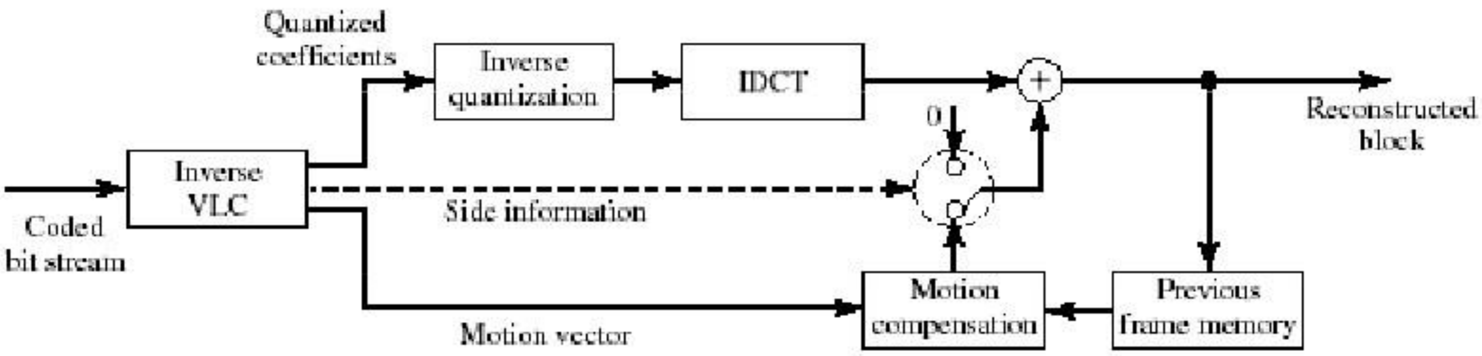
Key Idea in Video Compression

- Predict a new frame from a previous frame and only code the prediction error --- **Inter prediction**
- Predict a current block from previously coded blocks in the same frame --- **Intra prediction** (introduced in the latest standard H.264)
- Prediction error will be coded using the DCT method
- Prediction errors have smaller energy than the original pixel values and can be coded with fewer bits
- Those regions that cannot be predicted well will be coded directly using DCT --- **Intra coding without intra-prediction**
- Work on each macroblock (MB) (16x16 pixels) independently for reduced complexity
  - Motion compensation done at the MB level
  - DCT coding of error at the block level (8x8 pixels)

Encoder Block Diagram of a Typical Block-Based Video Coder (Assuming No Intra Prediction)



Decoder Block Diagram



Decoder

Perceptual Redundancy

Here is an image represented with 8-bits per pixel

Perceptual Redundancy

The same image at 7-bits per pixel

Perceptual Redundancy

At 6-bits per pixel

Perceptual Redundancy

At 5-bits per pixel

Perceptual Redundancy

At 4-bits per pixel

Perceptual Redundancy

It is clear that we don't need all these bits!

- Our previous example illustrated the eye's sensitivity to luminance

We can build a perceptual model

- Give more importance to what is perceivable to the Human Visual System
  - Usually this is a function of the spatial frequency

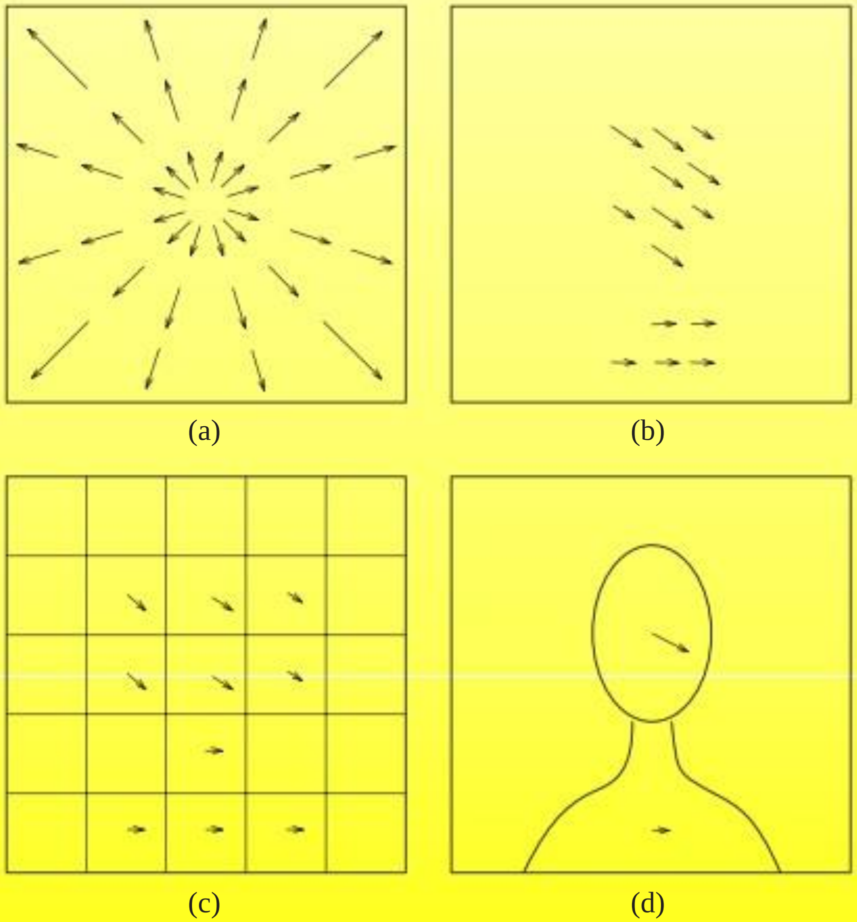
General Considerations for Motion Estimation

- Two categories of approaches:
  - Feature based (more often used in object tracking, 3D reconstruction from 2D)
  - Intensity based (based on constant intensity assumption) (more often used for motion compensated prediction, required in video coding, frame interpolation) -> Our focus
- Three important questions
  - How to represent the motion field?
  - What criteria to use to estimate motion parameters?
  - How to search motion parameters?

Motion Representation

Global:
Entire motion field is represented by a few global parameters

Pixel-based:
One motion vector at each pixel, with some smoothness constraint between adjacent motion vectors.

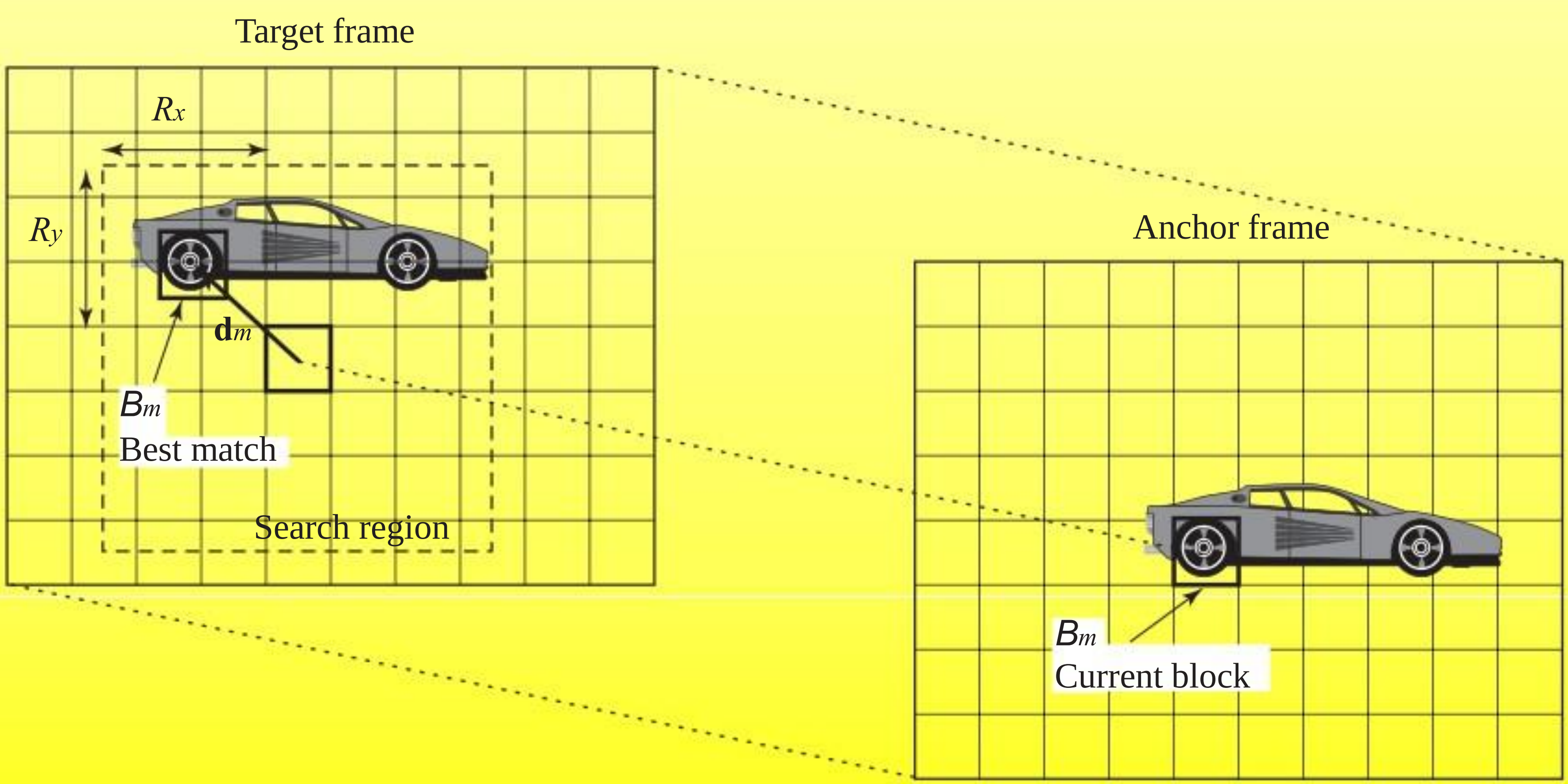


Block-based:
Entire frame is divided into blocks, and motion in each block is characterized by a few parameters.

Region-based:
Entire frame is divided into regions, each region corresponding to an object or sub-object with consistent motion, represented by a few parameters.

Other representation: mesh-based (control grid)

Exhaustive Block Matching Algorithm (EBMA)



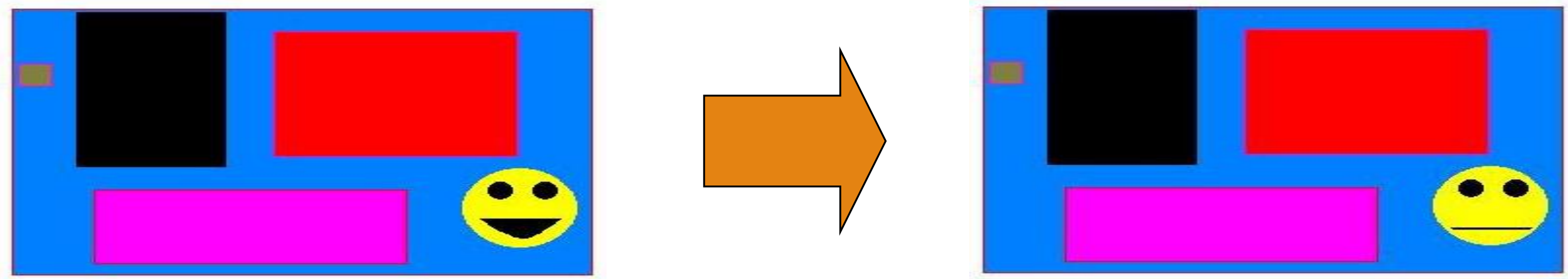
Pros and Cons with EBMA

- Blocking effect (discontinuity across block boundary) in the predicted image
  - Because the block-wise translation model is not accurate
  - Fix: Deformable BMA
- Motion field somewhat chaotic
  - because MVs are estimated independently from block to block
  - Fix 1: Mesh-based motion estimation
  - Fix 2: Imposing smoothness constraint explicitly
- Wrong MV in the flat region
  - because motion is indeterminate when spatial gradient is near zero
- Nonetheless, widely used for motion compensated prediction in video coding
  - Because its simplicity and optimality in minimizing prediction error

Random Access and Inter-frame Compression

Temporal Redundancy

- Only perform repeated encoding of the parts of a picture frame that are rapidly changing
- Do not repeatedly encode background elements and still elements



Random access capability

– Prediction that does not depend upon the user accessing the first frame (skipping through movie scenes, arbitrary point pick-up)

Motion Compensated Prediction

Divide current frame, i, into disjoint 16×16 macroblocks

Search a window in previous frame, i-1, for closest match

Calculate the prediction error

For each of the four 8×8 blocks in the macroblock, perform DCT-based coding

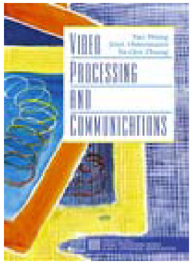
Transmit motion vector + entropy coded prediction error (lossy coding)

H.264/AVC Standards

- Developed by the joint video team (JVT) including video coding experts from the ITU-T and the ISO MPEG
- Finalized March 2003
- Improved video coding efficiency, up to 50% over H.263++/MPEG4
  - Half the bit rate for similar quality
  - Significantly better quality for the same bit rate
- Reference & figures for this section are from
  - *Ostermann et al., Video coding with H.264/AVC: Tools, performance, and complexity, IEEE Circuits and Systems Magazine, First Quarter, 2004*

New Video Coding Tools

- Intra-prediction
- Integer DCT with variable block sizes
- Adaptive deblocking filtering
- Multiple reference frame prediction



Spatial prediction

- H.261
  - Motion vector prediction using previously encoded MV
- MPEG-1
  - DC coefficients coded predictively
- H.263
  - MV prediction using the median of three neighbors
  - Optional: Intra DC prediction (10-15% improvement)
- MPEG-4
  - DC prediction: can predict DC coefficient from *either* the previous block or the block above
  - AC prediction: can predict one column/row of AC coefficients from *either* the previous block or the block above
- H.264
  - Pixel domain directional intra prediction

H.264 Intra prediction

•Instead of the simple DC coefficient prediction to exploit the correlation between nearby pixels in the same frame, more sophisticated spatial prediction is used
•Apply prediction to the entire 16*16 block (INTRA_16x16), or apply prediction separately to sixteen 4*4 blocks (INTRA_4x4)
•Adaptive directional prediction

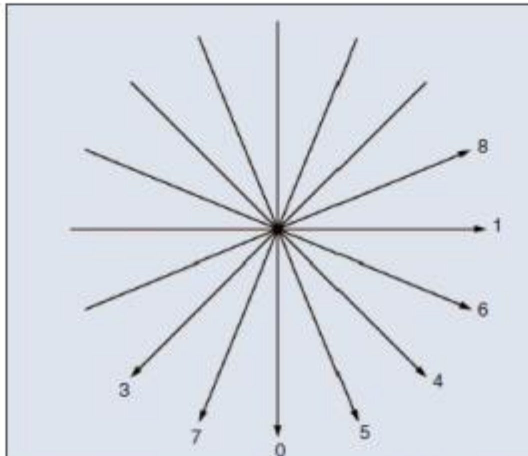


Figure 7. Possible prediction directions for INTRA_4×4 mode.

8 possible directions

Multiple Reference Frame Temporal Prediction

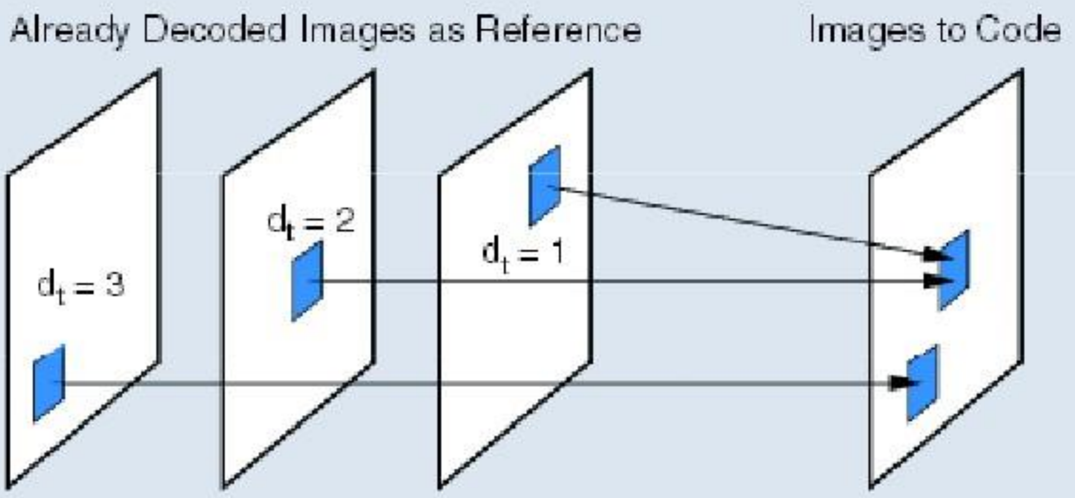


Figure 9. Motion-compensated prediction with multiple reference images. In addition to the motion vector, also an image reference parameter d_t is transmitted.

When multiple references are combined, the best weighting coefficients can be determined using ideas similar to minimal mean square error predictor

Spatial Prediction

- General idea:
  - A pixel in the new block is predicted from previously coded pixels in the same frame
  - What neighbors to use?
  - What weighting coefficients to use?
- Content-adaptive prediction
  - No edges: use all neighbors
  - With edges: use neighbors along the same direction
  - The best possible prediction pattern can be chosen from a set of candidates, similar to search for best matching block for inter-prediction
    - H.264 has many possible intra-prediction pattern

H.264 Intra-Prediction

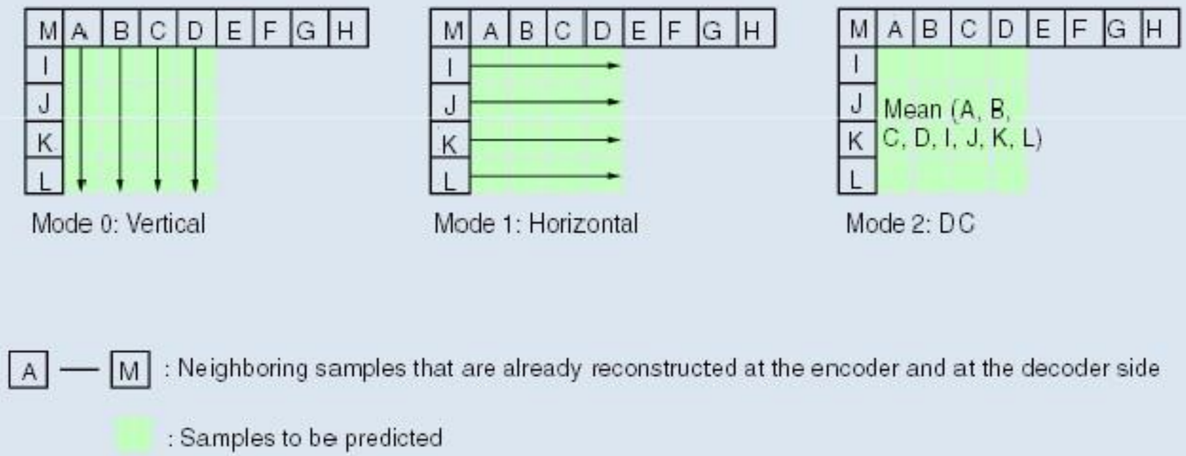


Figure 6. Three out of nine possible intra prediction modes for the intra prediction type INTRA_4×4.

From: Ostermann et al.,Video coding with H.264/AVC: Tools,performance,and complexity, IEEE Circuits and Systems Magazine, First Quarter, 2004

Coding of Prediction Error Blocks

- Error blocks typically still have spatial correlation
- To exploit this correlation:
    - Vector quantization
    - Transform coding
- Vector quantization
    - Can effectively exploit the typically error patterns due to motion estimation error
    - Computationally expensive, requires training
- Transform coding
    - Can work with a larger block under the same complexity constraint
    - Which transform to use?

Transform Coding of Error Blocks

- Theory: Karhunen Loeve Transform is best possible block-based transform
- Problems with theory:
  - Finding an accurate model (covariance matrix) of the source is difficult
  - Model and KLT change over time and in different regions
  - Decoder and encoder need to use same KLT
  - Implementation complexity: a full matrix multiply is necessary to implement KLT
- Practice: Discrete Cosine Transform
  - When the inter-pixel correlation approaches one, the KLT approaches the DCT

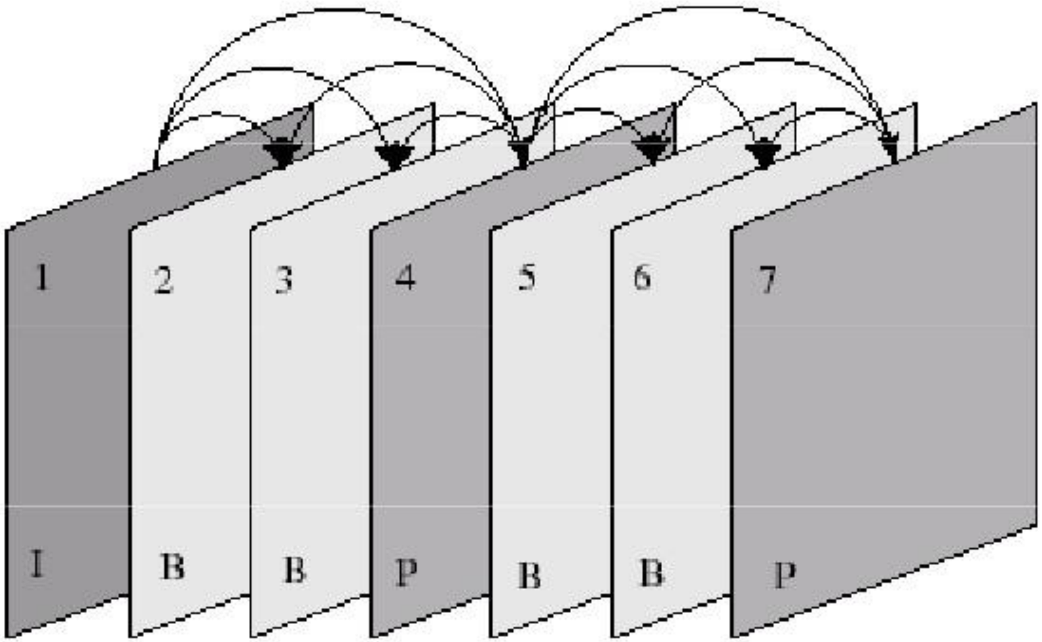
Transform Coding: What block size?

- Theory: Larger transform blocks (using more pixels) are more efficient
- Problem with theory:
  - Hard to get an accurate model of the correlation of distant pixels
  - In the limit as the inter-pixel correlation approaches one, the KLT approaches the DCT; however, the inter-pixel correlation of distant pixels is not close to one
- Practice:
  - Small block transforms - usually 8x8 pixels, although in more recent systems we can use 4x4 blocks or 16x16 blocks

Group-of-picture structure

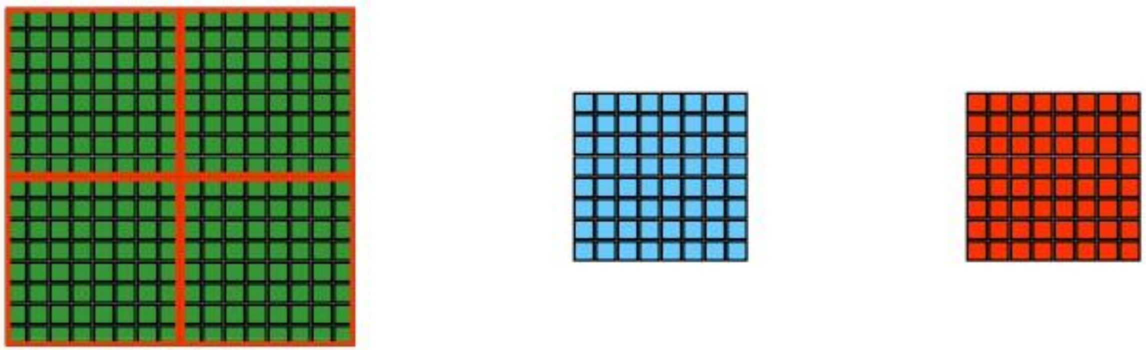
- I-frames coded without reference to other frames
  - To enable random access (channel change), fast forward, stopping error propagation
- P-frames coded with reference to previous frames
- B-frames coded with reference to previous and future frames
  - Requires extra delay!
  - Enable frame skip at receiver (temporal scalability)
- *Typically*, an I-frame every 15 frames (0.5 seconds)
- *Typically*, two B frames between each P frame
  - Compromise between compression and delay

Group of Picture Structure



Encoding order: 1 4 2 3 7 5 6

MB Structure in 4:2:0 Color Format



4 8x8 Y blocks 1 8x8 Cb blocks 1 8x8 Cr blocks

Macroblock Coding in I-Mode (assuming no intra-prediction)

DCT transform each 8x8 DCT block

↓

Quantize the DCT coefficients with properly chosen quantization matrices

↓

The quantized DCT coefficients are zig-zag ordered and run-length coded

With intra-prediction, after the best intra-prediction pattern is found, the prediction error block is coded using DCT as above.

Macroblock Coding in P-Mode

Estimate one MV for each macroblock (16x16)

↓

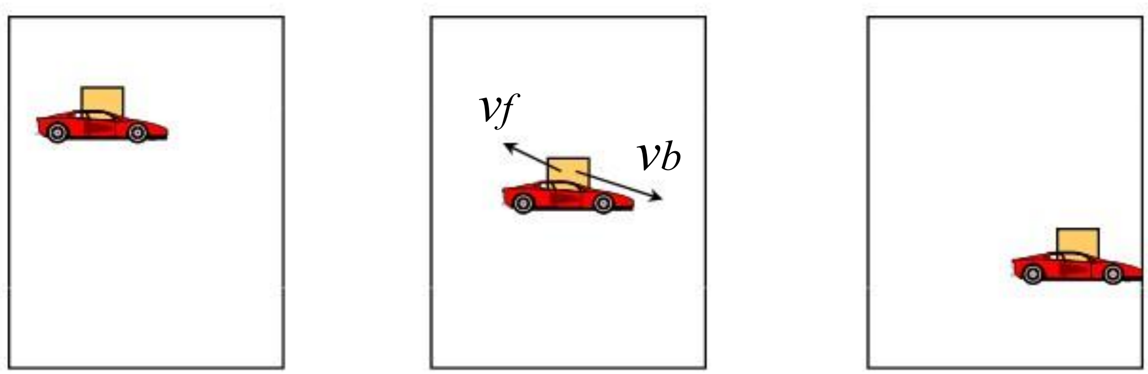
Depending on the motion compensation error, determine the coding mode (intra, inter-with-no-MC, inter-with-MC, etc.)

↓

The original values (for intra mode) or motion compensation errors (for inter mode) in each of the DCT blocks (8x8) are DCT transformed, quantized, zig-zag/alternate scanned,and run-length coded

Macroblock Coding in B-Mode

- Same as for the P-mode, except a macroblock can be predicted from a previous picture, a following one, or both.



Operational control of a video coder

- Typical video sequences contain varying content and motion
- Different content is compressed well with different techniques
- Encoders should match coding techniques to content

- Coding parameters
  - Macroblock type (coding mode)
  - Motion vectors
  - Quantizer step size
- Each leads to different rate and distortion

Rate Control: Why

- The coding method necessarily yields variable bit rate
- Active areas (with complex motion and/or complex texture) are hard to predict and require more bits under the same QP (quantization process)
- Rate control is necessary when the video is to be sent over a constant bit rate (CBR) channel, where the rate when averaged over a short period should be constant
- The fluctuation within the period can be smoothed by a buffer at the encoder output
  - Encoded bits (variable rates) are put into a buffer, and then drained at a constant rate
  - The encoder parameters (QP, frame rate) need to be adjusted so that the buffer does not overflow or underflow

Rate Control: How

- General ideas:
  - Step 1) Determine the target rate based on the current buffer fullness
  - Step 2) Satisfy the target rate by varying frame rate (skip frames when necessary) and QP
    - Determination of QP requires an accurate model relating rate with Q (quantization stepsize)
      - Model used in MPEG2: $R \sim A/Q + B/Q^2$
- A very complex problem in practice