Review of
COMP.SGN.110 Introduction to Image and Video Processing

Final exam (correction)

Moodle exam on

Wednesday 16 Dec 2020, 17:00 – 20:00 (sharp)

Make-up exam in March 2021.

Check Moodle, it should have the latest updates!

Course Outline

Chapter 1: Introduction to Digital Image Processing

Chapter 2: Digital Image Fundamentals

Chapter 3: Intensity Transformations and Spatial Filtering

Chapter 4: Filtering in the Frequency Domain

Chapter 5: Image Restoration and Reconstruction

Chapter 7: Color Image Processing

Chapter Video: Basics of Digital Video (different reference book)

Chapter 1: Introduction to Digital Image Processing

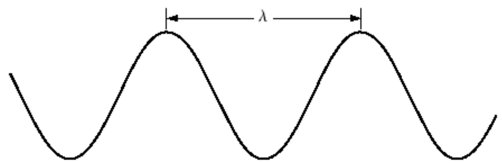
- Digital images (wide range of different types of images, imaging systems and applications
- The electromagnetic spectrum
- Digital images processing systems
- How are pictures made?
- Goals of image processing

Chapter 1: Introduction
Radiation-based images

Images based on radiation from Electro-Magnetic (EM) spectrum are most familiar, e.g. visible spectrum and X-ray 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.

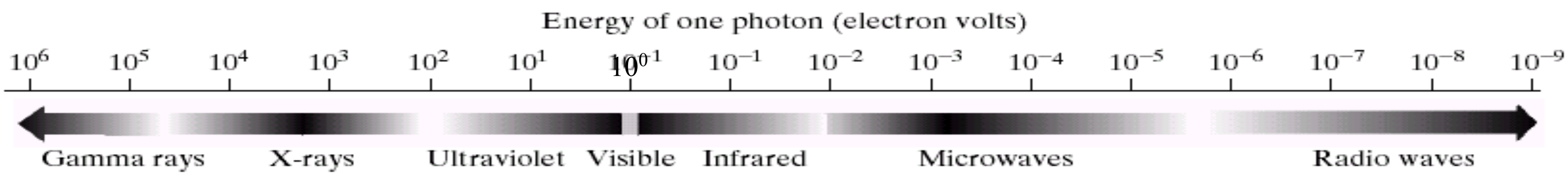


FIGURE 1.5 The electromagnetic spectrum arranged according to energy per photon.

Chapter 1
Introduction

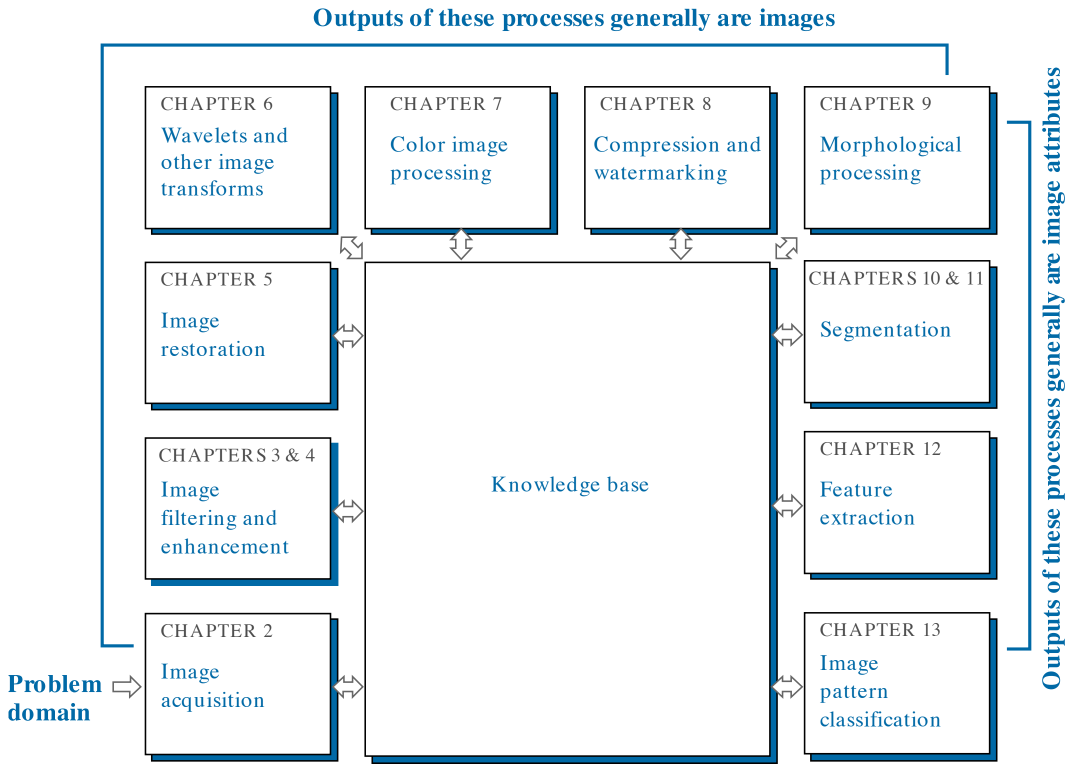


FIGURE 1.23 Fundamental steps in digital image processing. The chapter(s) indicated in the boxes is where the material described in the box is discussed.

Chapter 2: Digital Image Fundamentals

Outline

- Components of a Digital Imaging System
- 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 patters

Chapter 2: Digital Image Fundamentals

Structure of the Human Eye

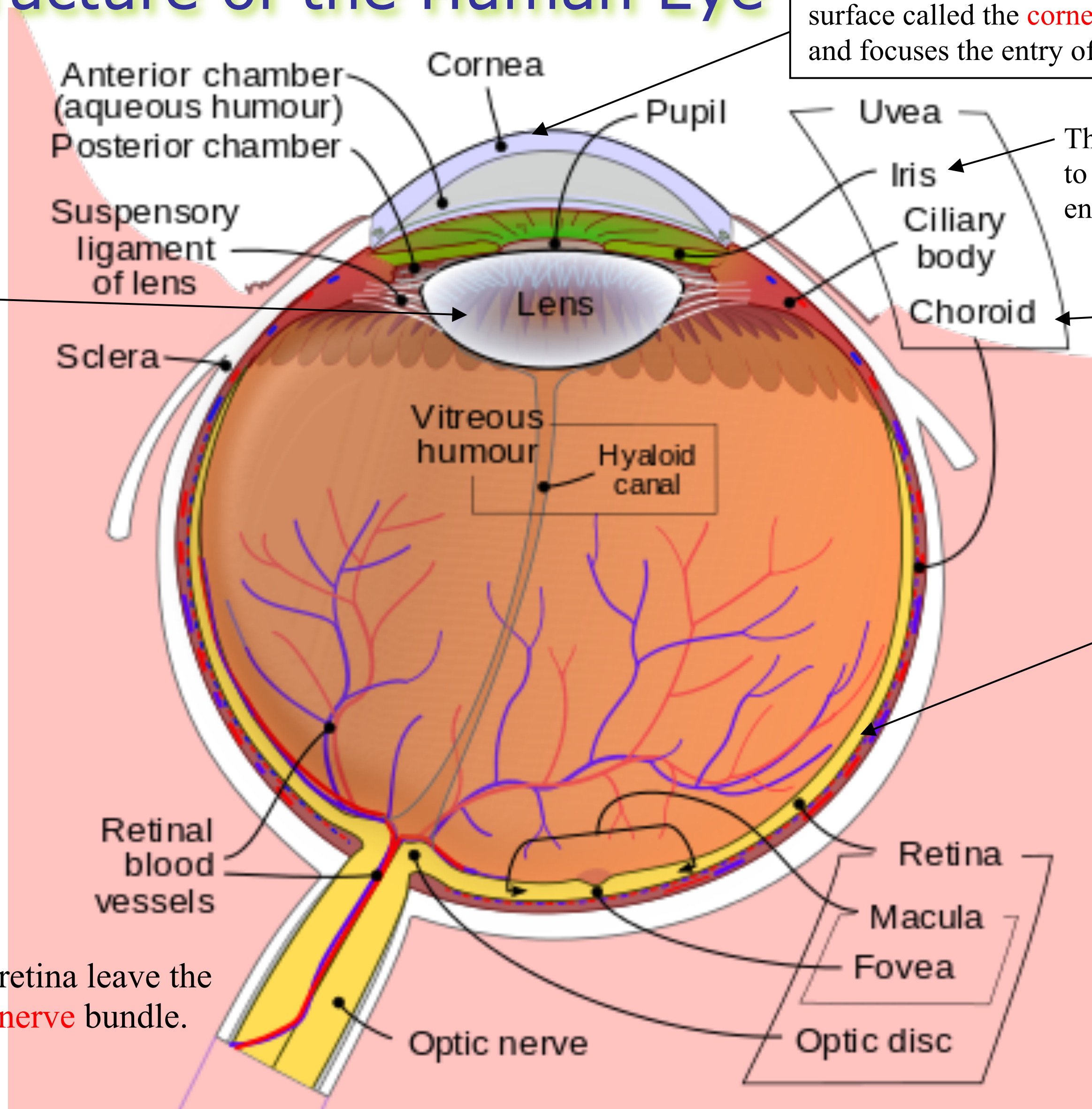
Front of eye is covered by a transparent surface called the cornea (a window that controls and focuses the entry of light into the **eye)**

The *iris* acts as a diaphragm to control the amount of light entering the eye.

Light entering the cornea is focused on the retina surface by a Lens

Inside the *choroid* is the

The **retina** is composed of two types of receptors: ***rods*** and ***cones***.



Nerves connecting to the retina leave the eyeball through the optic nerve bundle.

Chapter 2: Digital Image Fundamentals

Cones are responsible for photopic (color or bright-light) vision; while **rods** are for **scotopic** (dim-light) vision. Distribution of rods and cones in the retina.

Subjective (perceived) brightness is NOT a simple function of intensity, see **Mach bands and simultaneous contrast examples. Brightness adaptation and brightness discrimination, see Weber ratio.**

Weber ratio and **brightness**: a small Weber ratio indicates ”good” brightness where a small percentage change in illumination is discriminable. A large Weber ratio represents ”poor” brightness, i.e. a large percentage change in intensity is needed.

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

Chapter 2: Digital Image Fundamentals

high energies

high frequencies

short wavelegths

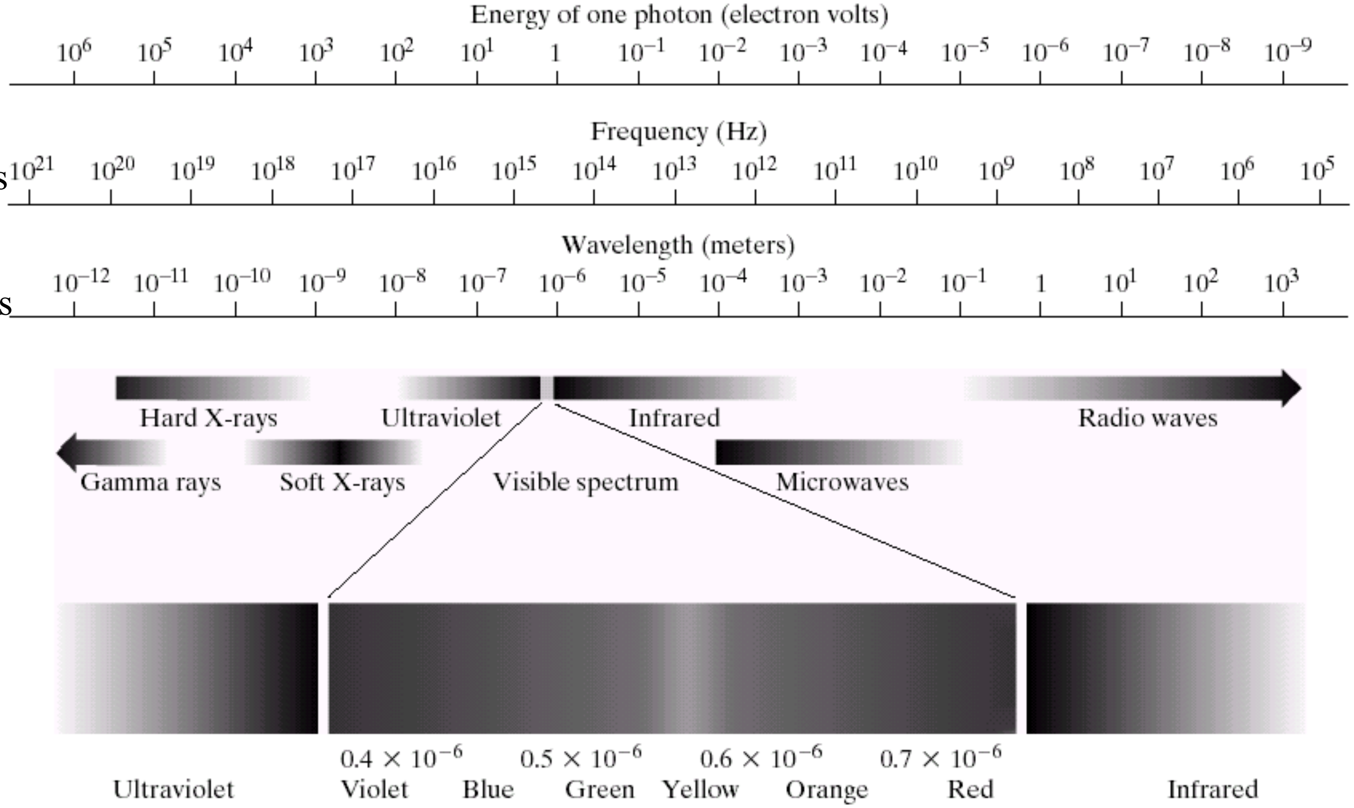


FIGURE 2.10 The electromagnetic spectrum. The visible spectrum is shown zoomed to facilitate explanation, but note that the visible spectrum is a rather narrow portion of the EM spectrum.

Digital Image Fundamentals

- Image acquisition: generating 2D images via scanning sensors
- Image formation models (illumination, reflectance, constraints)
- Image sampling and quantization (how many samples (size or spatial resolution) and how many gray levels (number of bits per sample) are needed (isopreference curves)?)
- Contouring effects
  - 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 contouring effect, we can use Contrast Quantization and dithering.
- Image interpolation (NN, Bilinear and bicubic)
- Image pixel neighborhood and common distance measures

Chapter 3: Intensity Transformations and Spatial Filtering

- Basic Intensity Transformation Functions
  - Contrast stretching
  - Grey level transformations
  - Image negatives
  - Log transformations
  - Power-law transformations
  - Piece-wise linear transformations
- Histogram processing (equalization and shaping)
- Enhancement using spatial filtering
- Smoothing spatial filtering: linear and nonlinear filtering
- Sharpening filters: first and second order derivatives (Laplacian)
- Laplacian with high-boost filtering
- Combining spatial enhancement methods

Chapter 3: Intensity Transformation and Spatial Filtering

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.
- lack of a general standard of **image quality**

Bit plane slicing and partial reconstruction

Spatial domain operations
- Intensity Transformations
  - Negative,
  - Contract stretching
    - Intensity level slicing (via piece-wise linear transformations)
    - Log and n’th root, Power law, Gamma correction
    - Image histogram processing (equalization, matching), global versus local histogram equalization
- Spatial filtering

Image Intensity Histogram

The **histogram** of an image with gray levels in the range [0,L-1] is a discrete function $h(r_k) = n_k$

r_k: the k-th gray level
n_k: number of pixels in the image having gray level r_k.

Probability Density Function (PDF) of image intensity values:
Normalized histogram → $p(r_k) = n_k / n$

A scanning electron microscope image of pollen magnified 700 times.

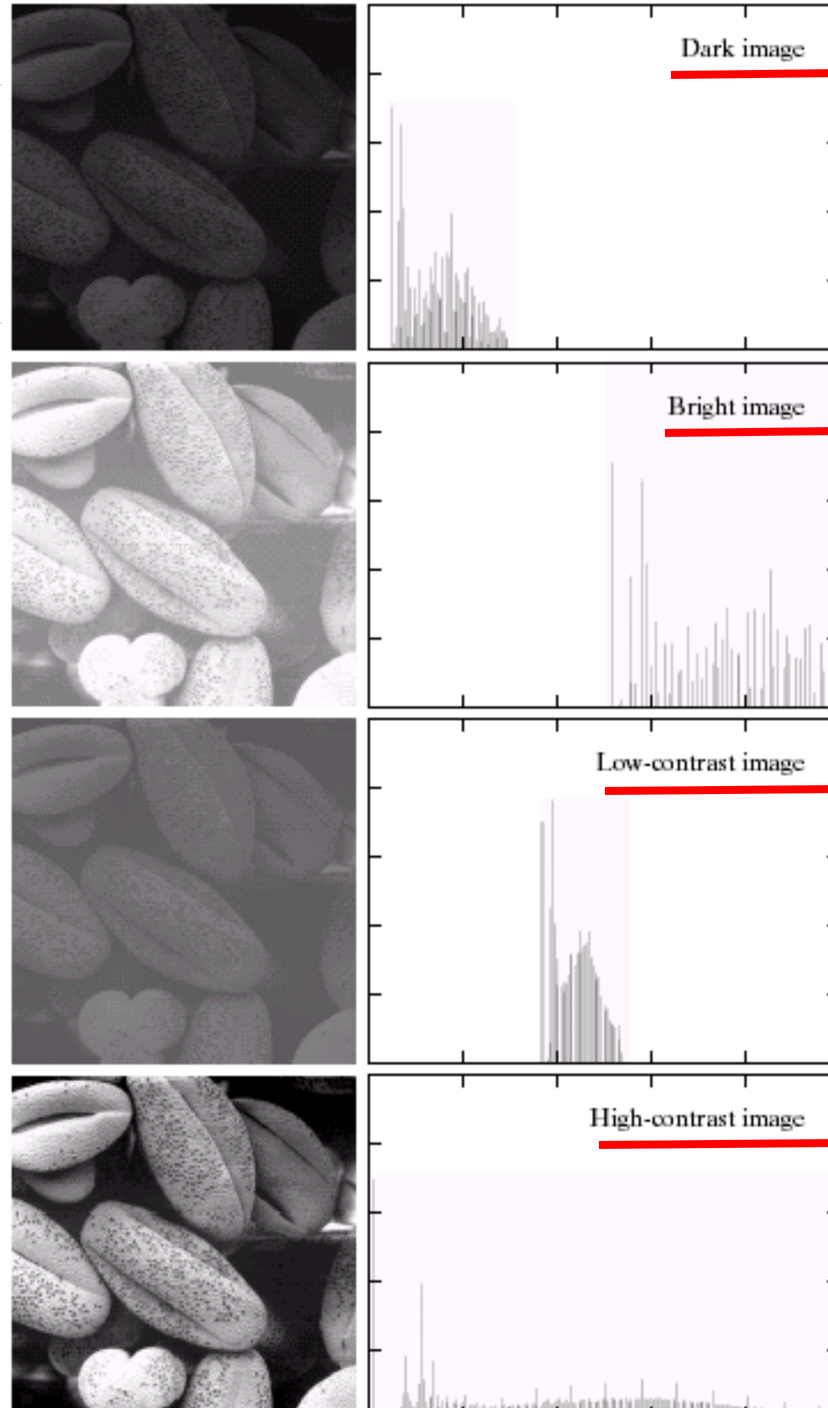
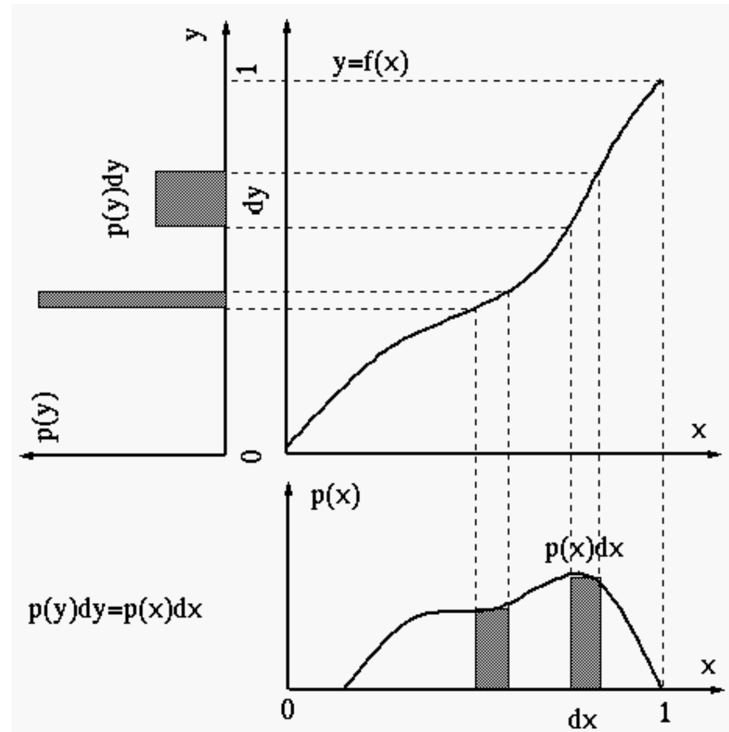


FIGURE 3.20 Left column: Images from Fig. 3.16. Center column: Corresponding histogram-equalized images. Right

Chapter 3: Intensity Transformations and Spatial Filtering
Histogram Equalization

- Histogram equalization transforms image gray levels in such a way that the histogram of the resulting image is equalized, i.e., it becomes a constant:

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



- The goal of histogram equalization:
  - equally use all available gray levels;

- 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.

Enhancement with Averaging Operations

When images are displayed (or printed), they often have suffered from noise and interferences from several sources including:

- electrical sensor noise,
- photographic grain noise, and
- channel errors.

These noise effects can be removed by simple ad hoc "noise-cleaning" techniques or more advanced methods, including deep (machine) learning) applied to local neighborhoods of input pixels.

Noisy image

Chapter 3: Intensity Transformations and Spatial Filtering

Enhancement with Averaging Operations

- ❖ Averaging K different noisy images:

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

  - ❖(check the effect on the mean and variance of the output)
- ❖ Spatial domain (local window) operations
- ❖ Correlation and convolution
- ❖ Smoothing linear filters
- ❖ Smoothing nonlinear filters
- ❖ Sharpening spatial filters (1st derivative (Roberts and Sobel operators), and 2nd derivative (Laplacian, and Laplacian with high boosting, concept of isotropy)
- ❖ Combining spatial enhancement methods

Chapter 3: Intensity Transformations and Spatial Filtering

Spatial Filtering Introduction

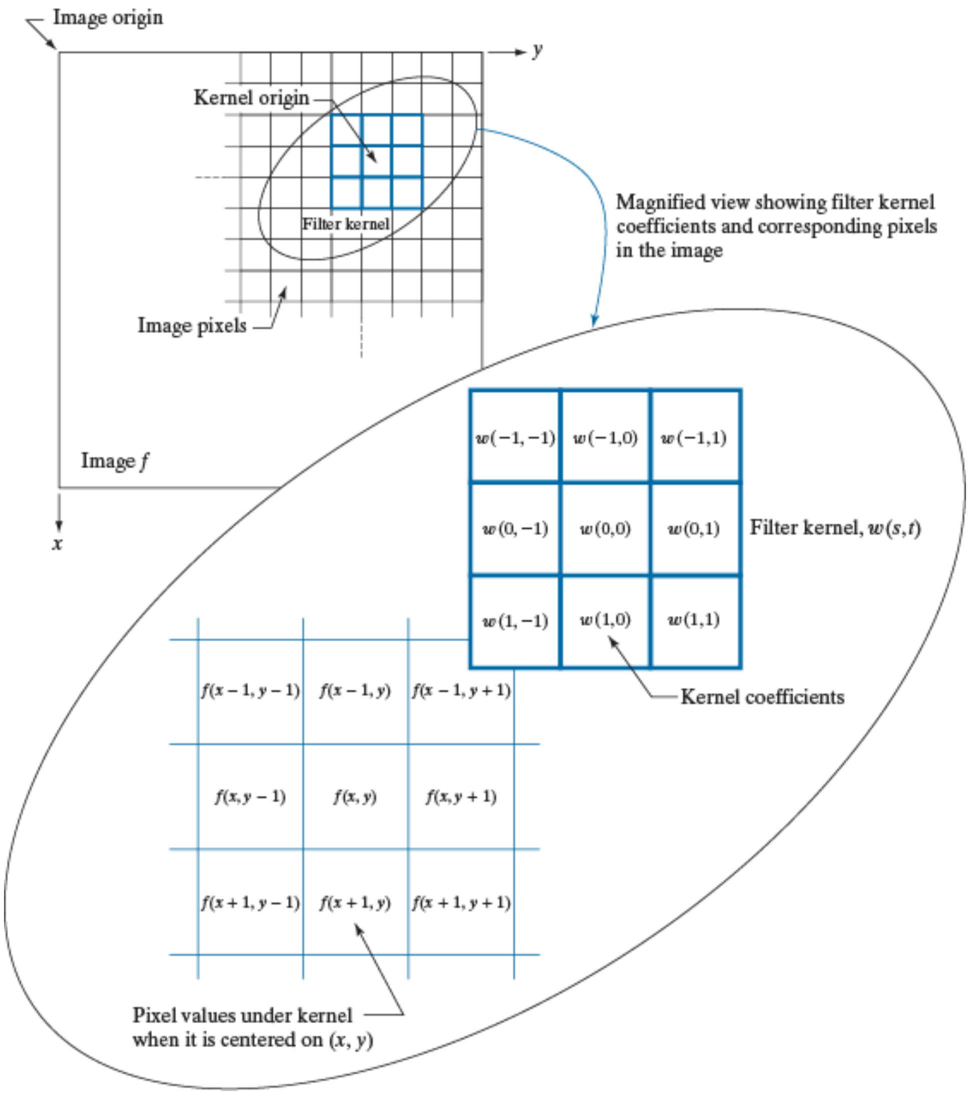
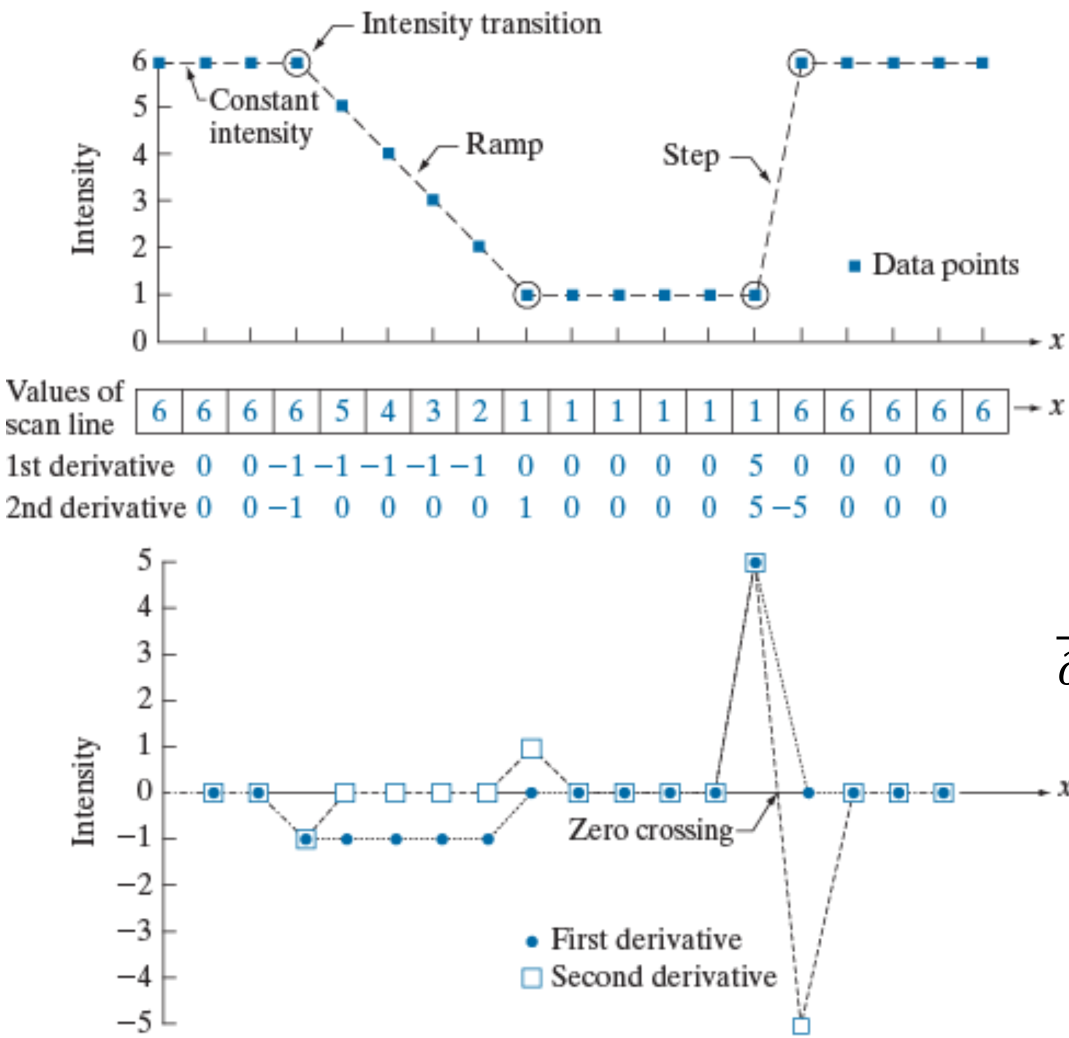


FIGURE 3.34 The mechanics of linear spatial filtering using a 3×3 kernel. The pixels are shown as squares to simplify the graphics. Note that the origin of the image is at the top left, but the origin of the kernel is at its center. Placing the origin at the center of spatially symmetric kernels simplifies writing expressions for linear filtering.

Each pixel is replaced by a weighted sum of its neighbourhood pixels in a sliding window

Chapter 3: Intensity Transformations and Spatial Filtering

Sharpening Spatial Filters



First derivative:

$$\frac{\partial f}{\partial x} = f(x+1) - f(x)$$

Second derivative:

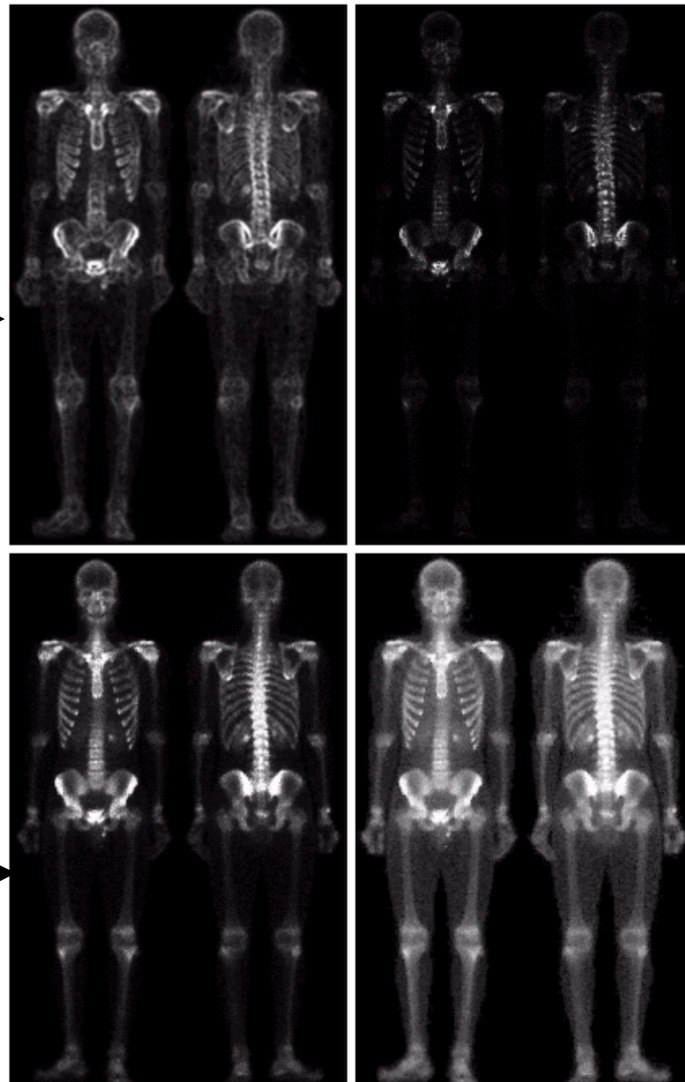
$$\frac{\partial^2 f}{\partial^2 x^2} = f(x+1) + f(x-1) - 2f(x)$$

Chapter 3: Intensity Transformations and Spatial Filtering

Combining Spatial Enhancement Methods

smoothed Sobel

Mask obtained by:
(orig + Lap) x (smoothed Sobel)



orig + mask

Final result power law to orig. + mask

e f
g h

FIGURE 3.63 *(Continued)* (e) Sobel image smoothed with a 5 × 5 box filter. (f) Mask image formed by the product of (b) and (e). (g) Sharpened image obtained by the adding images (a) and (f). (h) Final result obtained by applying a power-law transformation to (g). Compare images (g) and (h) with (a). (Original image courtesy of G.E. Medical Systems.)

Chapter 4
Filtering in the Frequency Domain

- Fourier Transform, Fourier Series, Discrete-time Fourier Transform, Discrete Fourier Transforms
- Basics of Frequency Domain Filtering
- Low-pass, High-pass, Bandpass/band-reject, Notch-pass/Notch-reject,
- Ideal, Butterworth, Gaussian filters
- Laplacian, High-boost, Homomorphic and Selective filtering

The Fourier Family

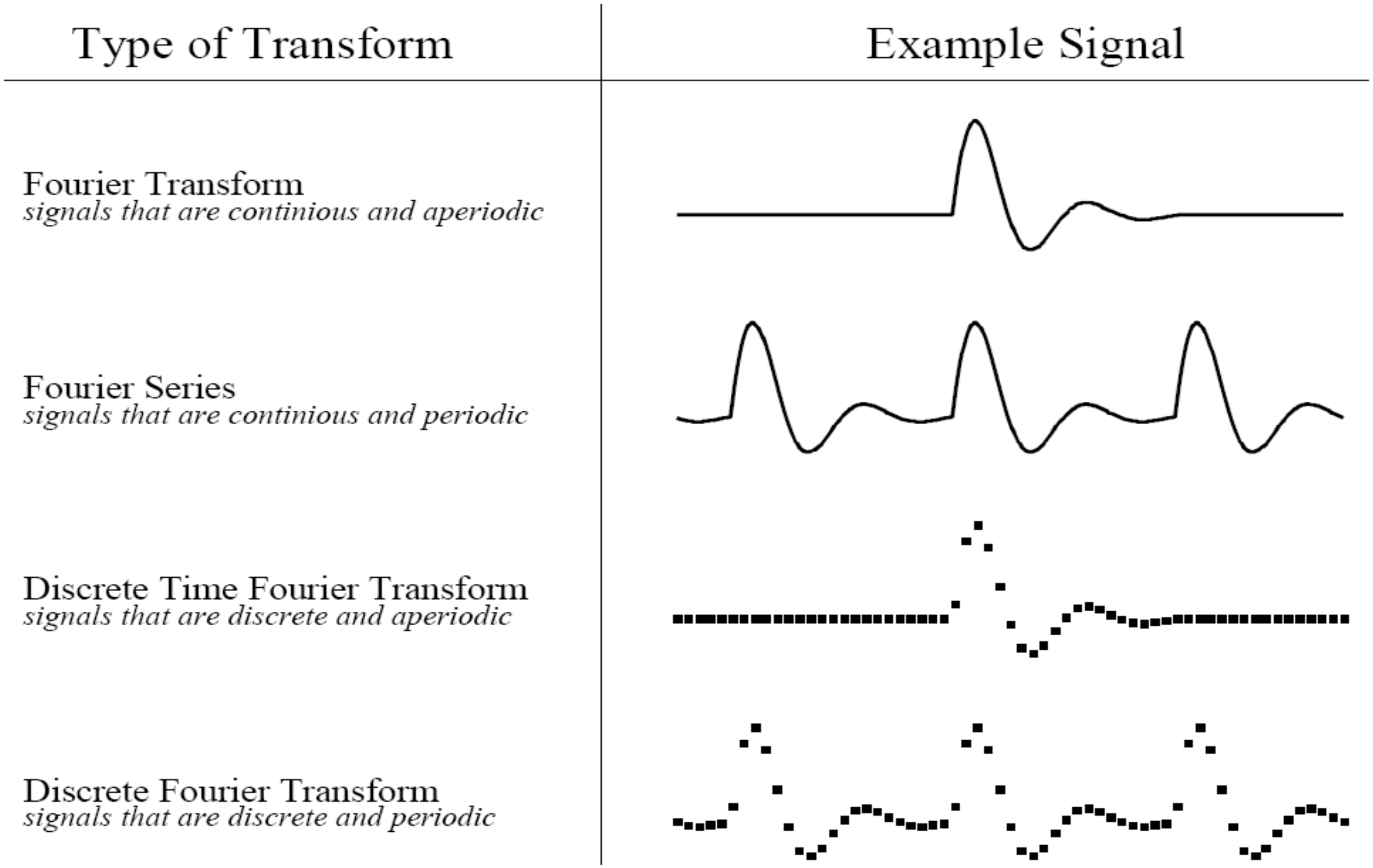


FIGURE 8-2
Illustration of the four Fourier transforms. A signal may be continuous or discrete, and it may be periodic or aperiodic. Together these define four possible combinations, each having its own version of the Fourier transform. The names are not well organized; simply memorize them.

Impulses and Their Sifting Property

- Unit impulse

$$\delta(t) = \begin{cases} \infty & \text{if } t = 0 \\ 0 & \text{if } t \neq 0 \end{cases} \qquad \int_{-\infty}^{\infty} \delta(t)dt = 1$$

- Sifting property

$$\int_{-\infty}^{\infty} f(t)\delta(t)dt = f(0) \qquad \int_{-\infty}^{\infty} f(t)\delta(t - t_0)dt = f(t_0)$$

- Unit discrete impulse

$$\delta(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{if } x \neq 0 \end{cases} \qquad \sum_{x=-\infty}^{\infty} \delta(x) = 1$$

- Sifting property

$$\sum_{x=-\infty}^{\infty} f(x)\delta(x) = f(0) \qquad \sum_{x=-\infty}^{\infty} f(x)\delta(x - x_0) = f(x_0)$$

Fourier Transform

$$\Im(f(t)) = F(\mu) = \int_{-\infty}^{\infty} f(t)e^{-j2\pi\mu t}\,dt$$

- Inverse Fourier Transform

$$f(t) = \int_{-\infty}^{\infty} F(\mu)e^{j2\pi\mu t}\,d\mu$$

- Using Euler formula:

$$F(\mu) = \int_{-\infty}^{\infty} f(t)[\cos(2\pi\mu t) - j\sin(2\pi\mu t)]\,dt$$

2-D Discrete Fourier Transform (DFT)

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

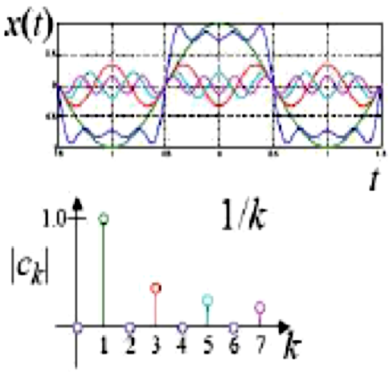
- Inverse discrete Fourier transform

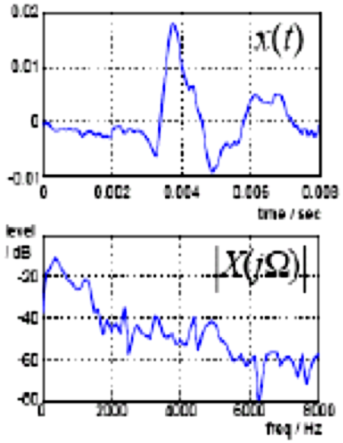
$$f(x,y) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} F(u,v)\, e^{j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)}$$

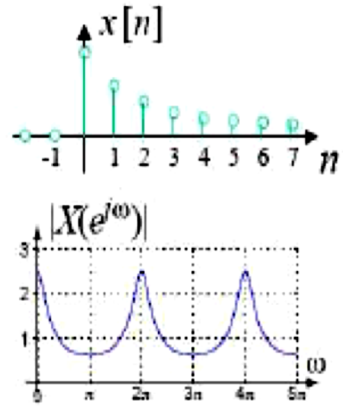
where $f(x,y)$ is a digital image of size MxN

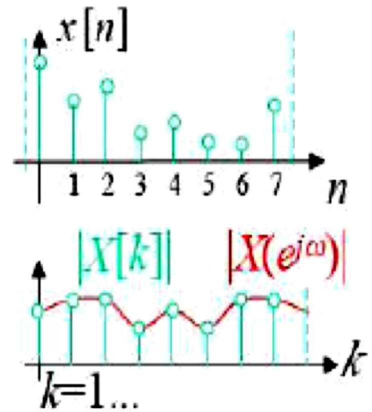
- Fourier spectrum and phase angle
- Translation property (centering F(0,0))
- DC term (zero-frequency)

SUMMARY









Fourier series:

- time: periodic continuous
- frequency: discrete

Fourier transform:

- time: non-periodic, continuous
- frequency: continuous

DTFT:

- time: discrete
- frequency: periodic, continuous

DFT:

- time: "periodic", discrete
- frequency: periodic, discrete

$T \rightarrow \infty \Rightarrow, \omega_0 \rightarrow 0$

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

Uniform sampling in [0,2π]

Convolution

- Convolution of two continuous functions $f(t)$ and $h(t)$

$$f(t) \star h(t) = \int_{-\infty}^{\infty} f(\tau)h(t-\tau)d\tau$$

- Convolution theorem

$$f(t) \star h(t) \Leftrightarrow H(\mu)F(\mu)$$
$$f(t)h(t) \Leftrightarrow H(\mu) \star F(\mu)$$

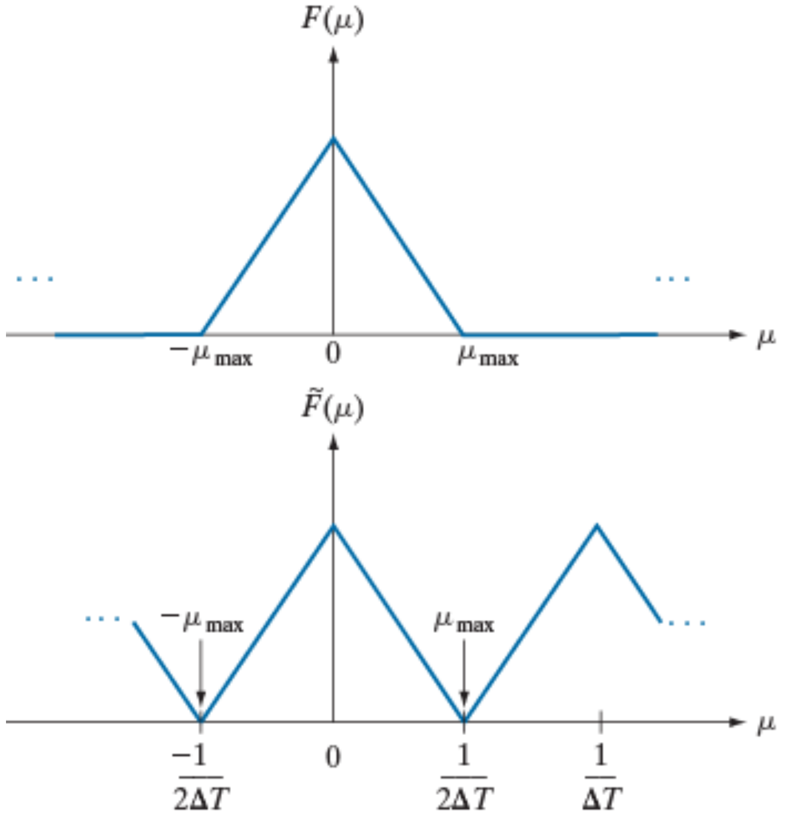
- Convolution with and without zero-padding!

Sampling Theorem

- Signal $F(\mu)$ can be recovered uniquely from $\tilde{F}(\mu)$ if $\frac{1}{\Delta T} > 2\mu_{\max}$
- Here $\frac{1}{\Delta T}$ is the Nyquist rate

a
b

FIGURE 4.7 (a) Illustrative sketch of the Fourier transform of a band-limited function. (b) Transform resulting from critically sampling that band-limited function.



Aliasing

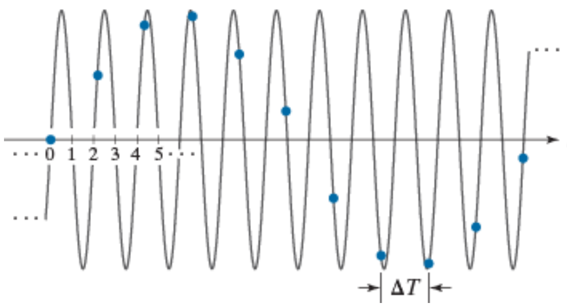
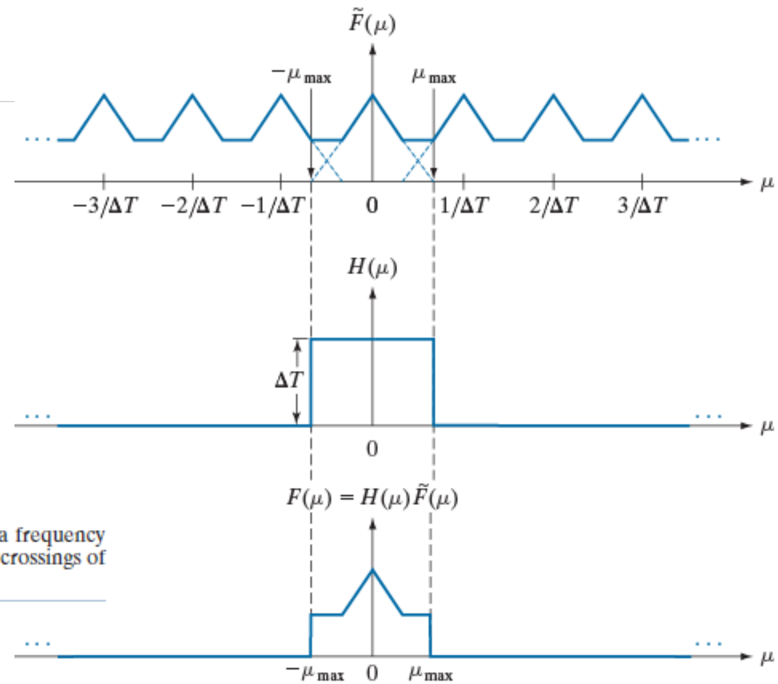


FIGURE 4.11 Illustration of aliasing. The under-sampled function (dots) looks like a sine wave having a frequency much lower than the frequency of the continuous signal. The period of the sine wave is 2 s, so the zero crossings of the horizontal axis occur every second. ΔT is the separation between samples.



a
b
c

FIGURE 4.10 (a) Fourier transform of an under-sampled, band-limited function. (Interference between adjacent periods is shown dashed). (b) The same ideal lowpass filter used in Fig. 4.8. (c) The product of (a) and (b). The interference from adjacent periods results in aliasing that prevents perfect recovery of $F(\mu)$ and, consequently, of $f(t)$.

Aliasing in Images

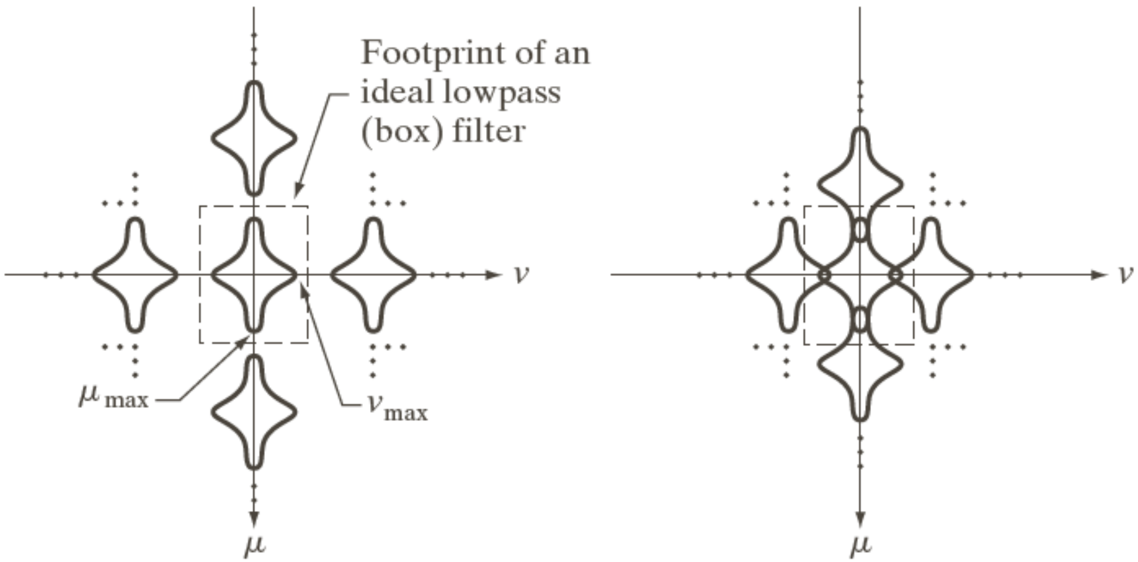


FIGURE 4.16 Two-dimensional Fourier transforms of (a) an over-sampled, and (b) an under-sampled, band-limited function.

- To avoid aliasing $\frac{1}{\Delta T} > 2\mu_{\max}$ and $\frac{1}{\Delta Z} > 2v_{\max}$

Basics of Filtering in Frequency Domain

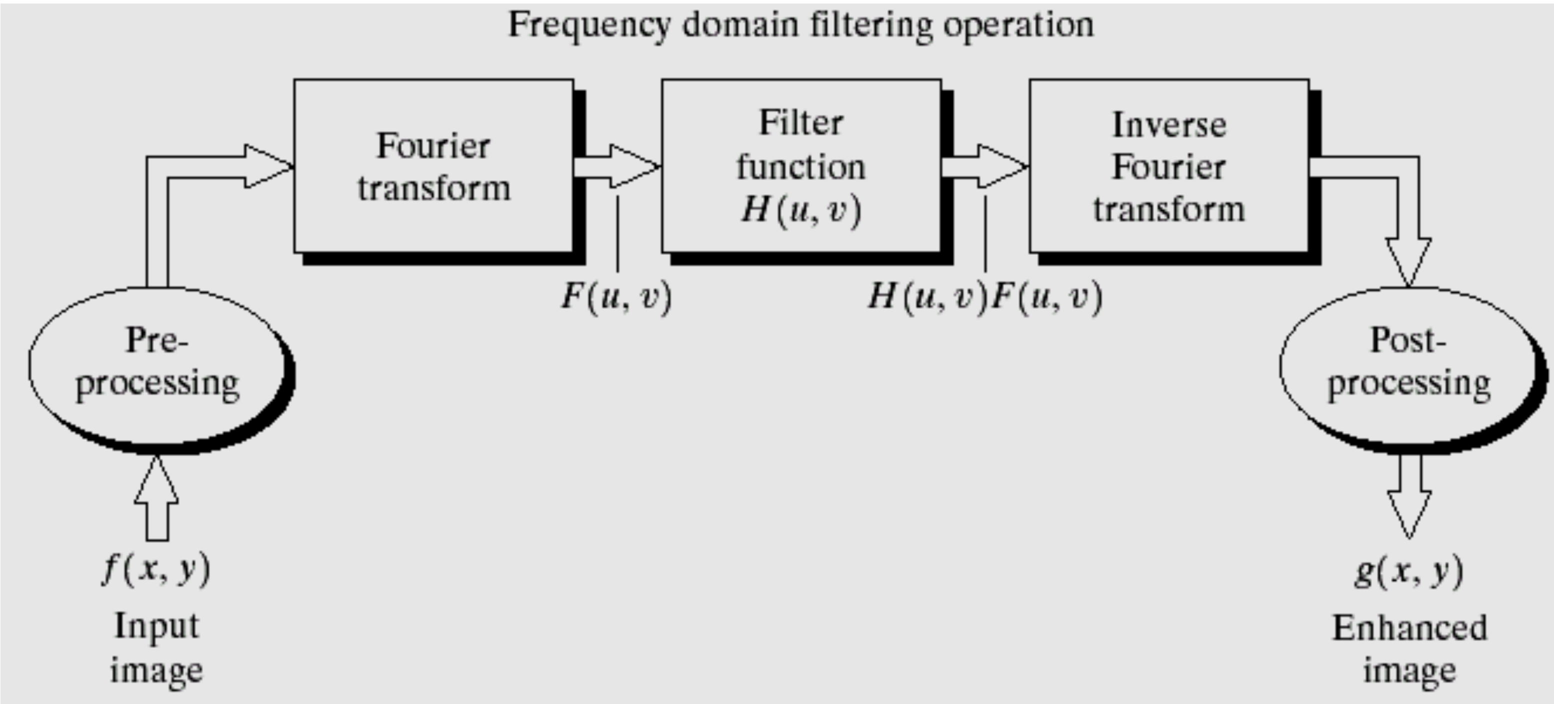


FIGURE 4.5 Basic steps for filtering in the frequency domain.

Image domain ⟷ *Fourier domain*

convolution ⟷ *multiplication*

Basics of Filtering in the Frequency Domain

To filter an input image $f(x, y)$ of size MxN with a real symmetric filter $H(u, v)$, we need to do the following:

1) Pad image $f(x, y)$ with zeros to the size PxQ. Typically, P=2M and Q=2N $\Longrightarrow$ call the padded image $f_P(x, y)$
2) Multiply $f_P(x, y)$ by $(-1)^{x+y}$ to center its transformation
3) Compute DFT, $F(u, v)$
4) Compute the product $G(u, v) = H(u, v)F(u, v)$
5) Compute IDFT and take the real part
$$g_P(x, y) = \text{Real}(DFT^{-1}[G(u, v)])(-1)^{x+y}$$
6) Obtain g(x,y) by extracting the MxN region from top left quadrant

Frequency Domain Filtering Guidelines

- Low frequencies correspond to slowly changing intensities in the spatial domain, e.g. monotonous regions
- High frequencies correspond to sharp transitions, such as edges and noise
- *Low-pass filtering* suppresses high frequencies
  $\Longrightarrow$ *smoothing* (blurring)
- *High-pass filtering* suppresses low frequencies
  $\Longrightarrow$ *sharpening*
- Low-pass and high-pass filters are naturally related via:

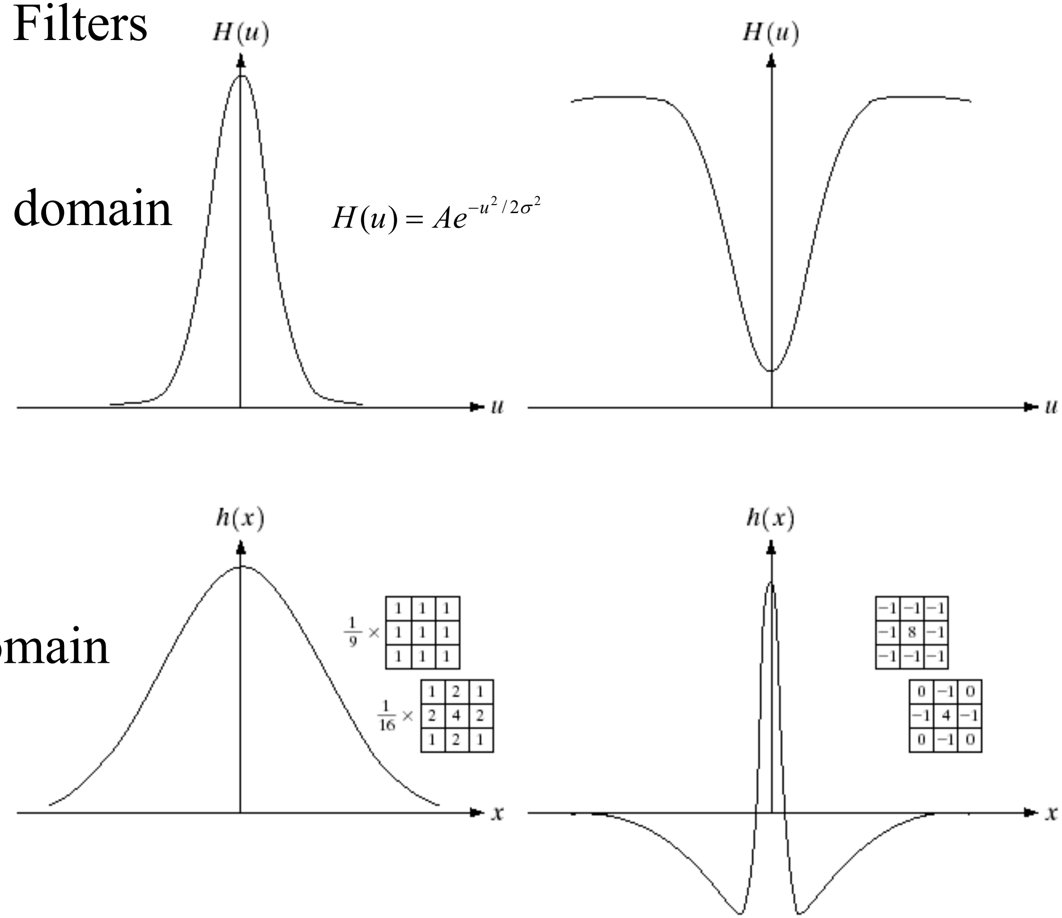
$$H_{HP}(u, v) = 1 - H_{LP}(u, v)$$

Chapter 4: Filtering in the Frequency Domain
Blurring and sharpening in spatial and frequency domains

3. Gaussian Filters

frequency domain

spatial domain



Low-pass high-pass

a b
c d

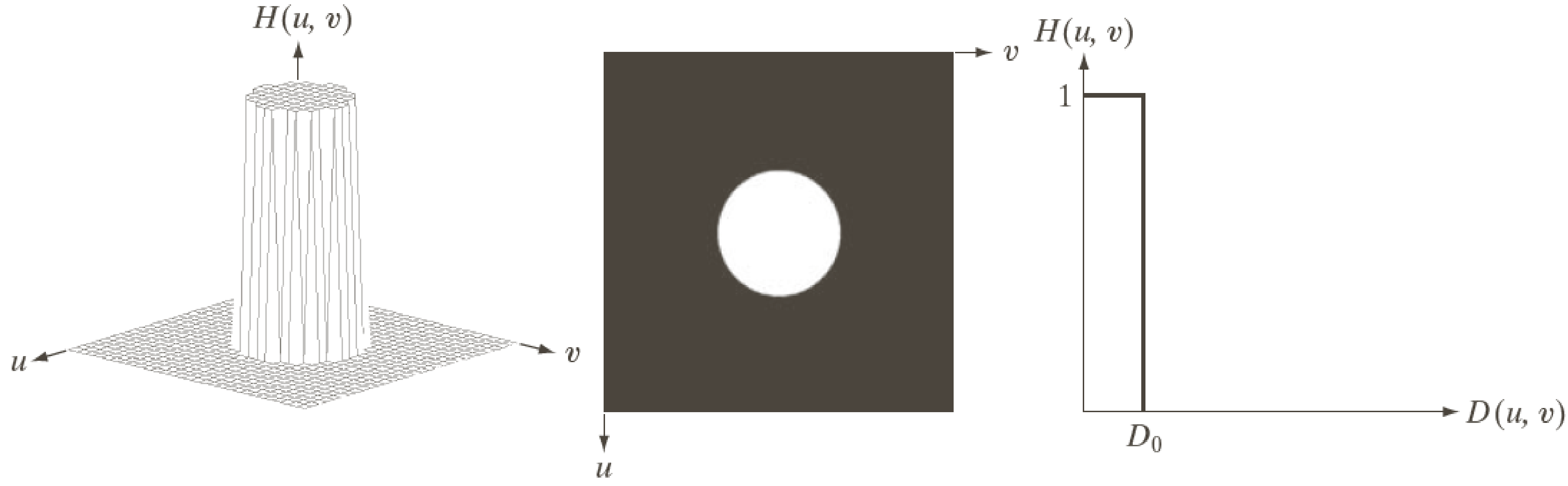
FIGURE 4.9
(a) Gaussian frequency domain lowpass filter.
(b) Gaussian frequency domain highpass filter.
(c) Corresponding lowpass spatial filter.
(d) Corresponding highpass spatial filter. The masks shown are used in Chapter 3 for lowpass and highpass filtering.

Chapter 4
Filtering in the Frequency Domain

4. Ideal low-pass filter

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

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



a b c

FIGURE 4.40 (a) Perspective plot of an ideal lowpass-filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

Ideal Low-Pass Filters

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

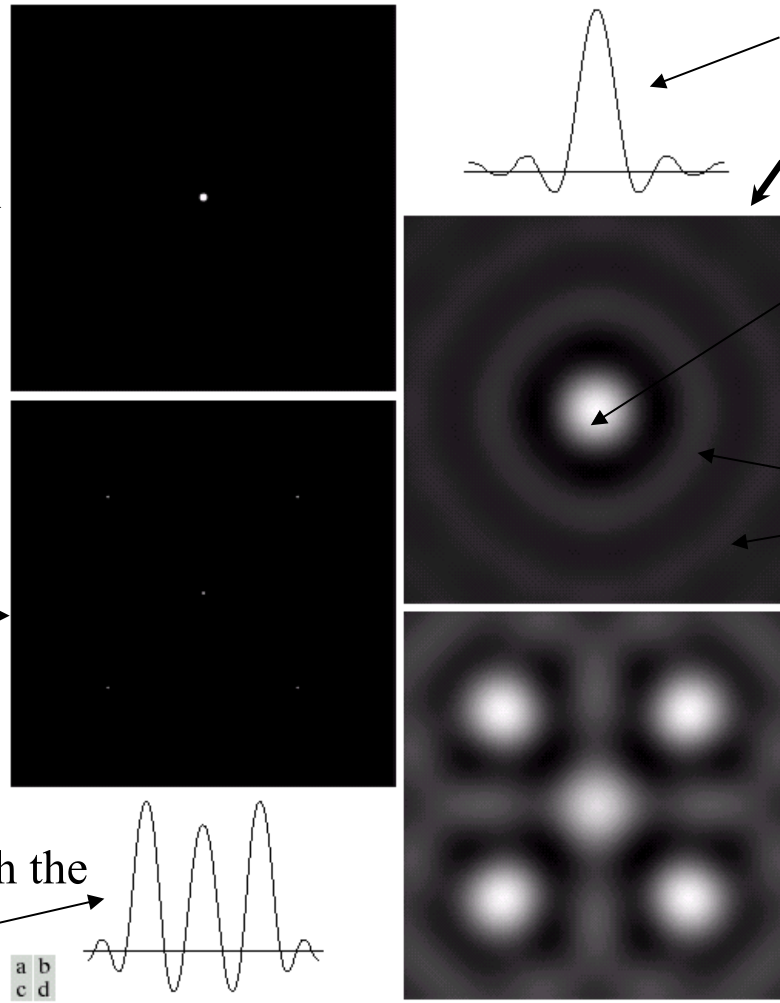


FIGURE 4.13 (a) A frequency-domain ILPF of radius 5. (b) Corresponding spatial filter (note the ringing). (c) Five impulses in the spatial domain, simulating the values of five pixels. (d) Convolution of (b) and (c) in the spatial domain.

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!

Butterworth Low-Pass Filters

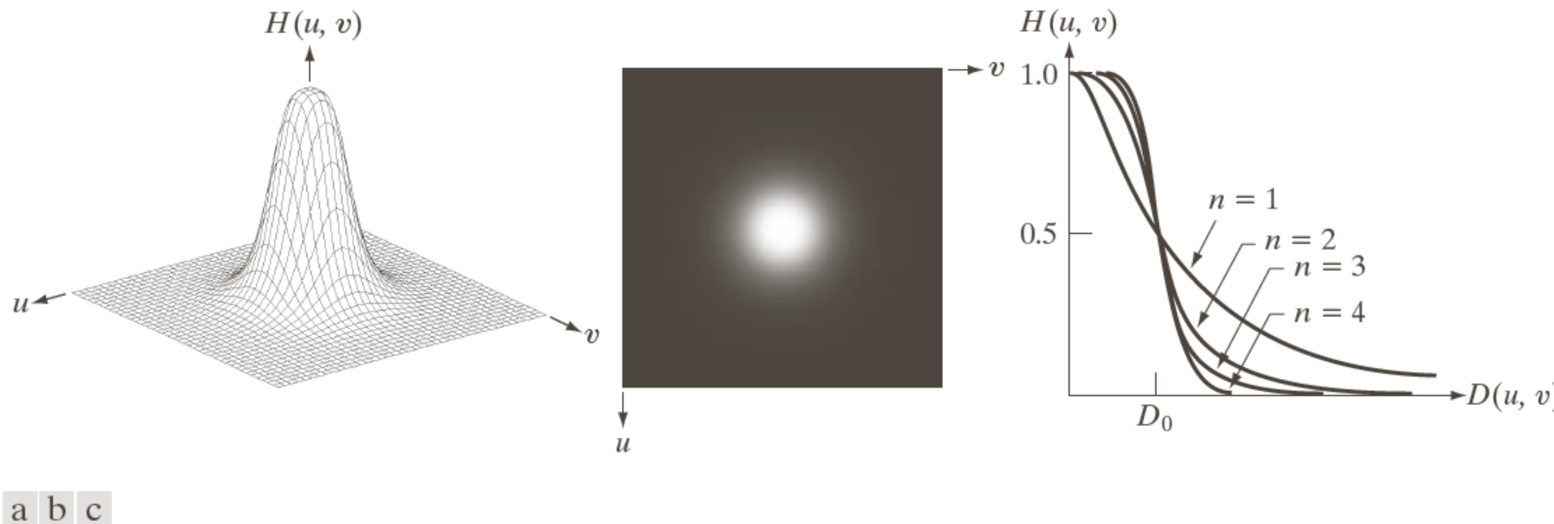


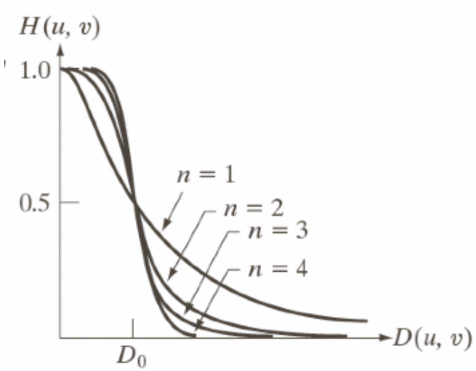
FIGURE 4.44 (a) Perspective plot of a Butterworth lowpass-filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.

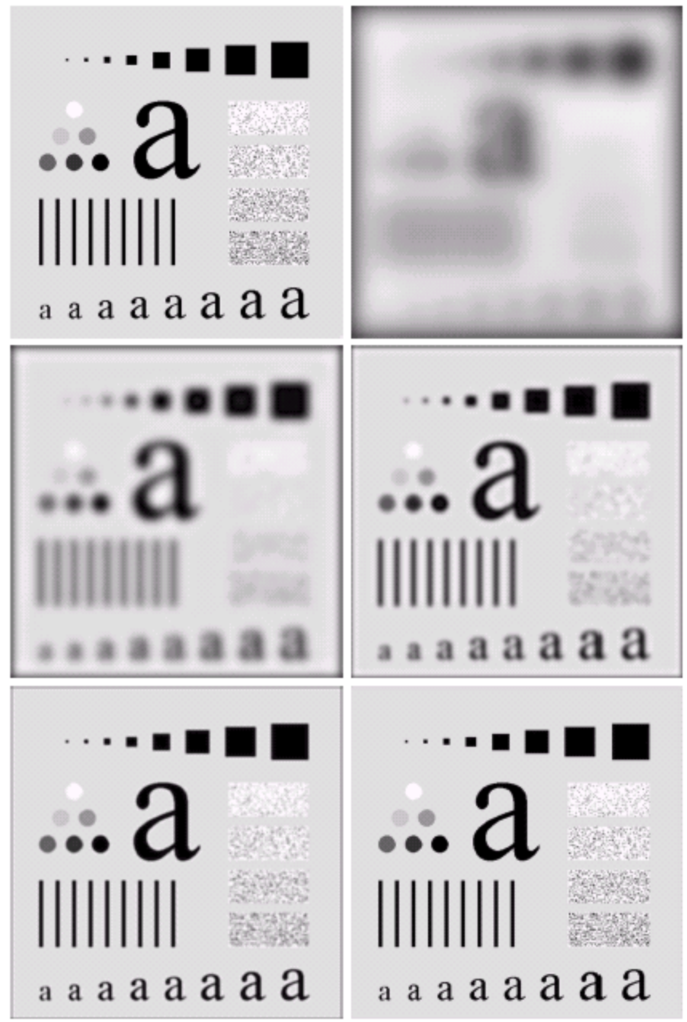
$$H(u,v) = \frac{1}{1 + [D(u,v)/D_0]^{2n}}$$

n – order of the filter

Butterworth Low-Pass Filters

Filtering with BLPF with **n=2** and increasing cut-off as was done with the Ideal LPF





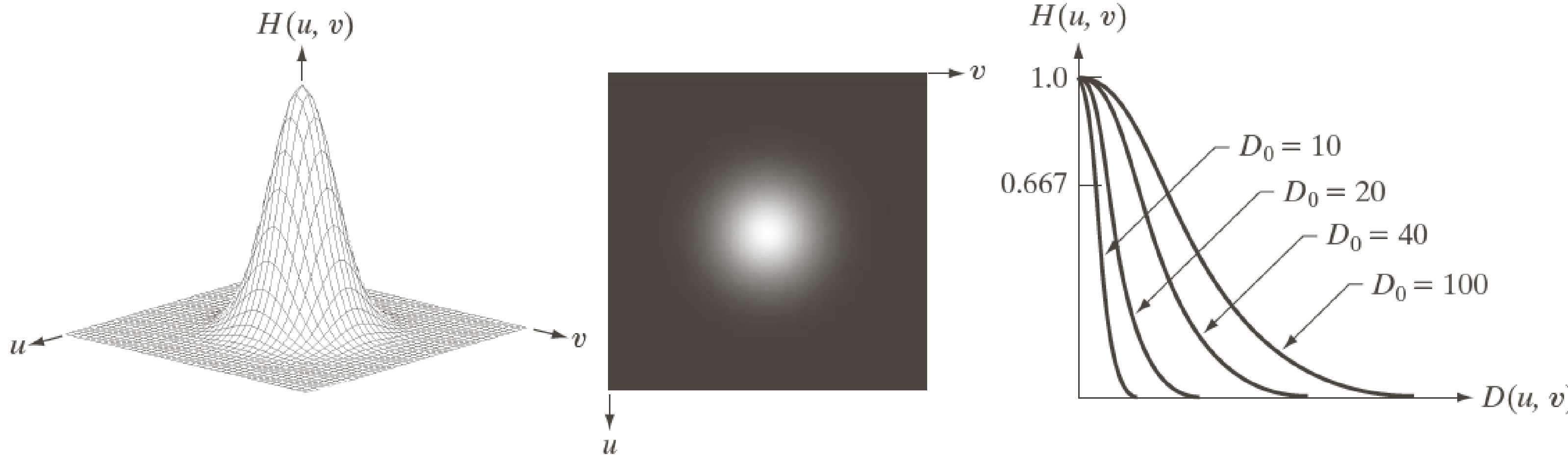
a b
c d
e f

FIGURE 4.15 (a) Original image. (b)–(f) Results of filtering with BLPFs of order 2, with cutoff frequencies at radii of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Fig. 4.12.

Note the smooth transition in blurring achieved as a function of increasing cutoff but no ringing is present in any of the filtered images with this particular BLPF (with n=2)

this is attributed to the smooth transition between low and high frequencies

Gaussian Low-Pass Filters



a b c

FIGURE 4.47 (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.

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

Does this filter suffer from ringing artefacts?

+ IDFT of the Gaussian filter is also Gaussian $\Longrightarrow$ **no ringing!**

– transition less sharp than BLFP

High-Pass Filters in Frequency Domain

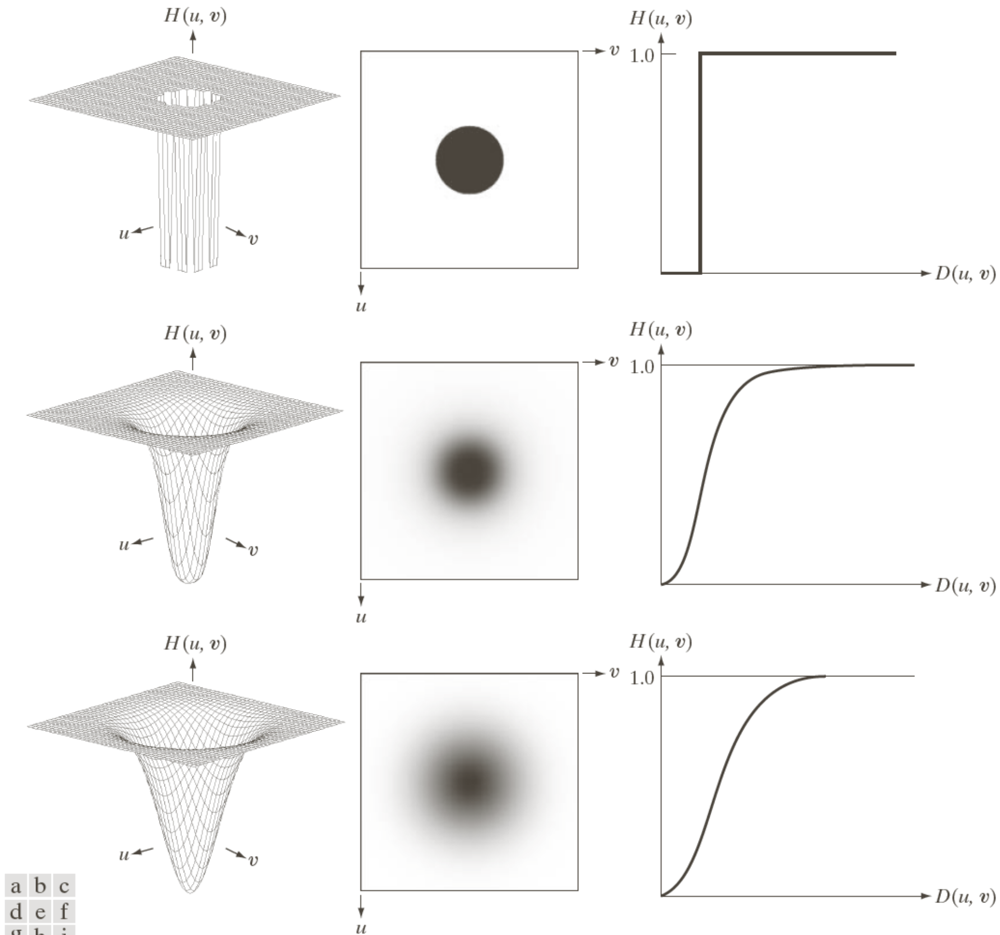


FIGURE 4.52 Top row: Perspective plot, image representation, and cross section of a typical ideal highpass filter. Middle and bottom rows: The same sequence for typical Butterworth and Gaussian highpass filters.

Ideal

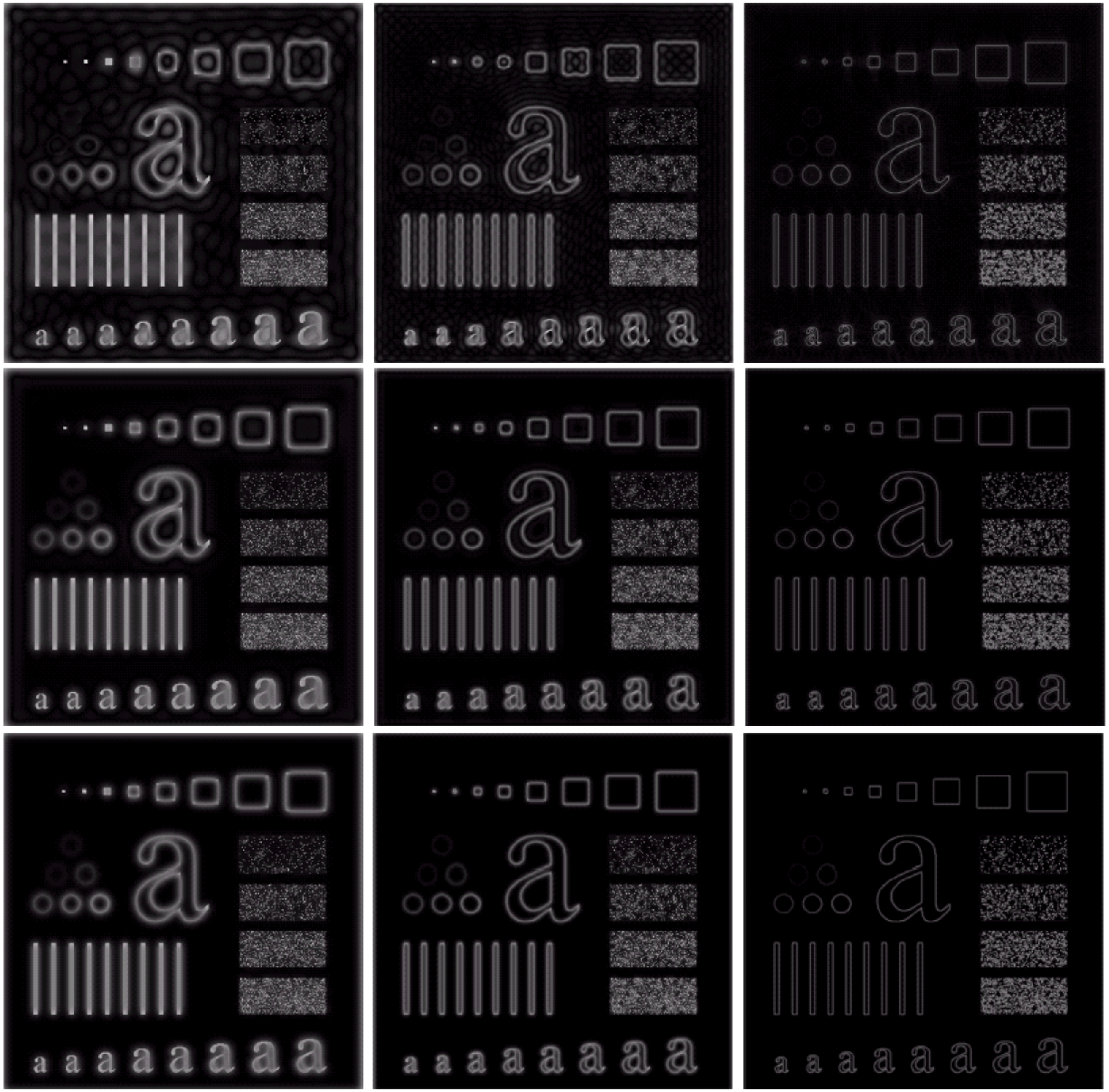
$$H(u,v) = \begin{cases} 0 & if\ D(u,v) \leq D_0 \\ 1 & if\ D(u,v) \succ D_0 \end{cases}$$

Butterworth

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

Gaussian

$$H(u,v) = 1 - e^{-D^2(u,v)/2{D_0}^2}$$



Ideal HPF

BHPF

GHPF

a b c

FIGURE 4.26 Results of highpass filtering the image of Fig. 4.11(a) using a GHPF of order 2 with $D_0 = 15$, 30, and 80, respectively. Compare with Figs. 4.24 and 4.25.

Laplacian in the Frequency Domain

- **Chapter 3**: edges can be enhanced by adding the Laplacian (second derivative) of the image on top of the original:
$$g(x,y) = f(x,y) + c\nabla^2 f(x,y)$$
- *Laplacian filter* can be implemented in the frequency domain:
$$H(u,v) = -4\pi^2 D^2(u,v)$$
- Laplacian image can then be obtained:
$$\nabla^2 f(x,y) = \Im^{-1}\{H(u,v)F(u,v)\}$$
- $c = -1$ should be chosen in this case, as $H(u,v) < 0$
- Results are <u>visually identical</u> to the equivalent spatial filter

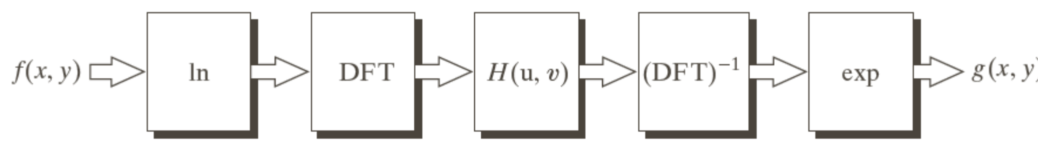
- Unsharp masking and high-boost filtering

Homomorphic Filtering

- Recall the *illumination-reflectance model*, in which the image can be expressed as the product of *illumination* and *reflectance* components:

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

FIGURE 4.60 Summary of steps in homomorphic filtering.

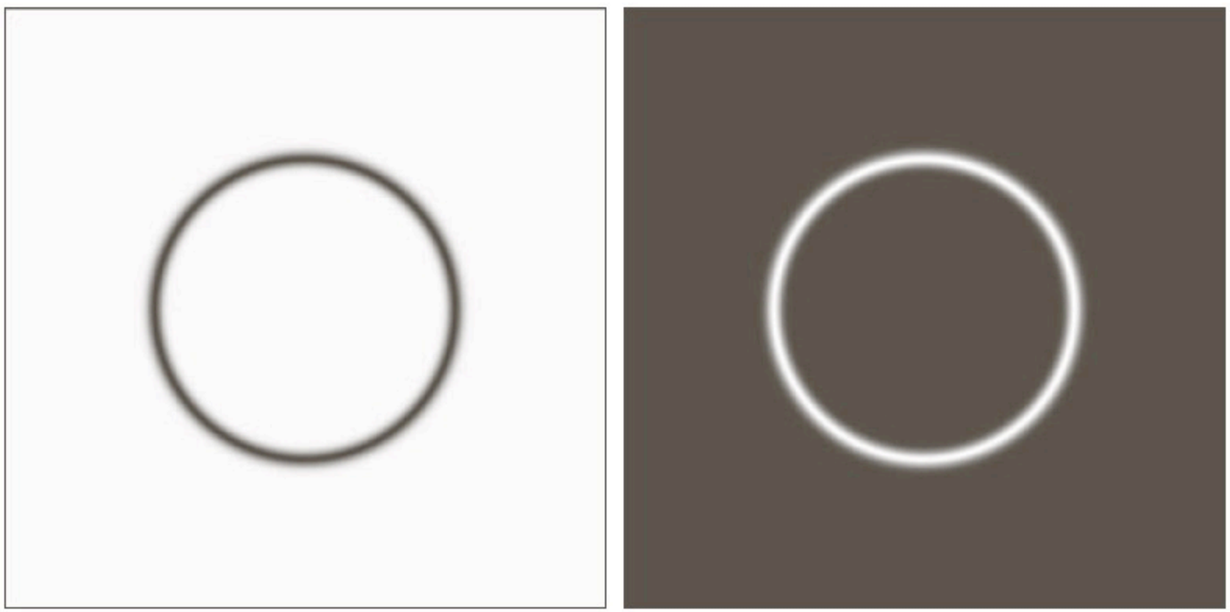


Homomorphic filtering approach allows for both low frequencies (≈ illumination) and high frequencies (≈ reflectance) to be adjusted simultaneously

Selective Filtering Examples

Band-reject filter

Band-pass filter



a b

FIGURE 4.63 (a) Bandreject Gaussian filter. (b) Corresponding bandpass filter. The thin black border in (a) was added for clarity; it is not part of the data.

Chapter 5: Image Restoration and Reconstruction

- Image degradation and restoration model
- Common noise densities (Gaussian, Rayleigh, Gamma, Exponential, Uniform, Impulse and periodic)
- Noise parameter estimation
- Restoration in the presence of noise only
- Linear position invariant degradations
- Degradation function estimation
- Inverse filtering
- Minimum Mean Square Error Wiener filtering

General Degradation Model

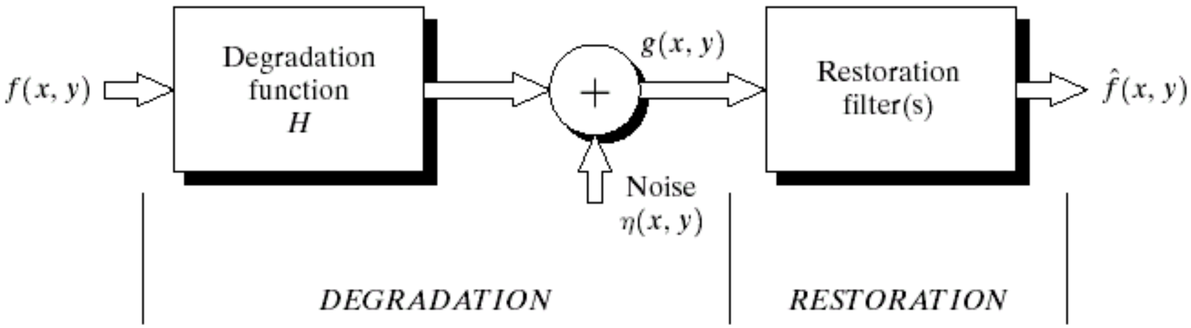


FIGURE 5.1 A model of the image degradation/restoration process.

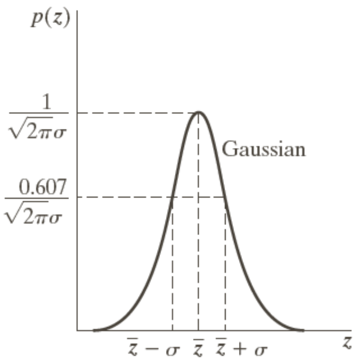
<u>Assumption</u>: *H* is linear and position-invariant

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

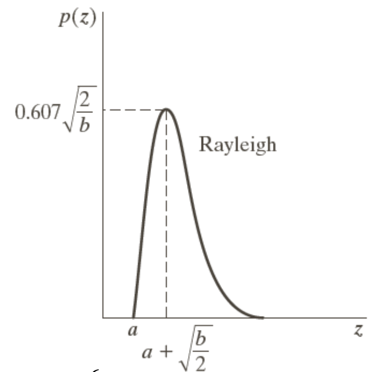
Frequency domain: $G(u,v) = H(u,v)F(u,v) + N(u,v)$

<u>Goal</u>: design restoration filter(s) s.t. $\hat{f}(x,y)$ is close to $f(x,y)$

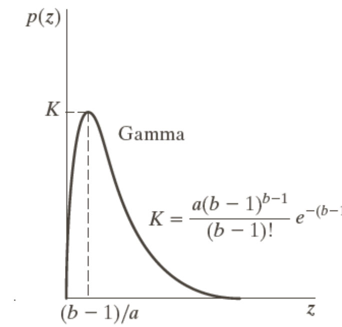
Noise Models, how to estimate their parameters



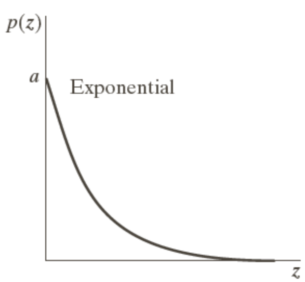
$$p(z) = \frac{1}{\sqrt{2\pi\sigma}} e^{-(z-\mu)^2/2\sigma^2}$$



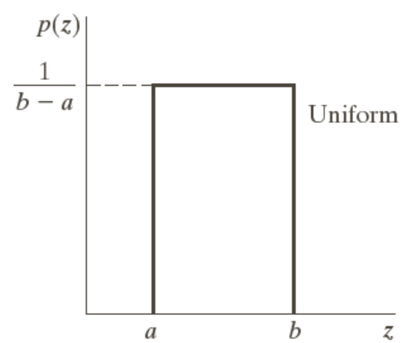
$$p(z) = \begin{cases} \frac{2}{b} e^{-(z-a)^2/b} & \text{for } z \ge a \\ 0 & \text{for } z < a \end{cases}$$



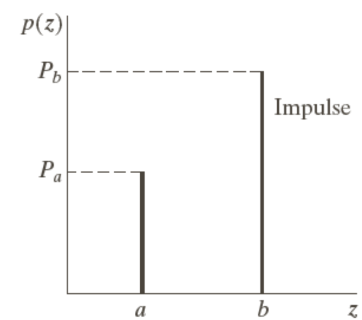
$$p(z) = \begin{cases} \frac{a^b z^{b-1}}{(b-1)!} e^{-az} & \text{for } z \ge 0 \\ 0 & \text{for } z < 0 \end{cases}$$



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



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



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

Restoration in the Presence of Noise Only

- Assume that the only degradation present in the image is noise:

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

- If noise is periodic, it may be possible to directly estimate $N(u,v)$ and subtract it
- Usually spatial filtering is a method of choice

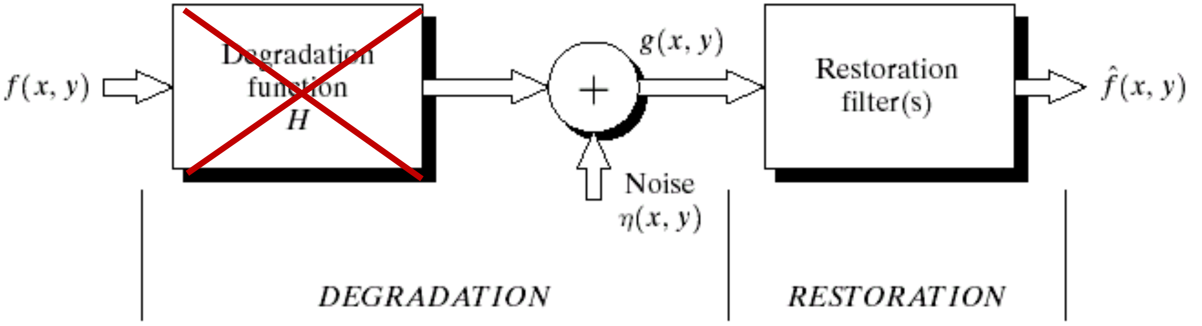


FIGURE 5.1 A model of the image degradation/restoration process.

Spatial Filters

- Arithmetic mean filter
- Geometric mean filter
- Harmonic and contra-harmonic mean filters
- Median and Order-Statistic filters (max, min, mid-point, alpha-trimmed mean)
- Adaptive mean and adaptive median filter

Frequency Domain Filters

- Band-reject filtering (Ideal, Butterworth, Gaussian)
- Band-pass and Band-reject filtering
- Notch-reject and Notch-pass filtering
- Optimal Notch filtering

General Degradation Model

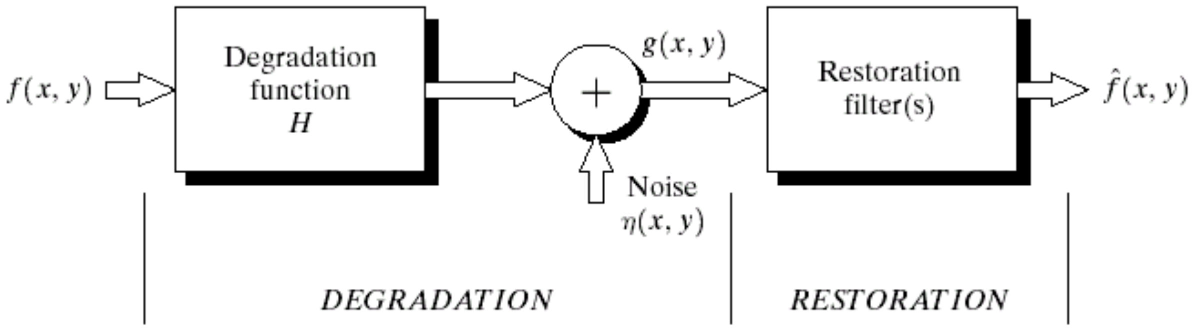


FIGURE 5.1 A model of the image degradation/restoration process.

Assumption: *H* is linear and position-invariant

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

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

Goal: design restoration filter(s) s.t. $\hat{f}(x, y)$ is close to $f(x, y)$

Linear Position-Invariant Degradations

- Assumptions: H is linear and position invariant
- This allows us to obtain the 2-D convolution integral:

$$g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta)\, h(x-\alpha, y-\beta)\, d\alpha\, d\beta$$

- The <u>impulse response</u> of the linear system is <u>sufficient</u> to compute its response to <u>any input</u>
- Taking the additive noise into account, we obtain:

$$g(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(\alpha,\beta)\, h(x-\alpha, y-\beta)\, d\alpha\, d\beta + \eta(x,y)$$

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

Estimating the Degradation Function

- Assumptions about the nature of the degradation function allow us to utilize tools of linear system theory
- Nonlinear, position-variant degradations can be approximated
- Degradation is modelled as a convolution with the degradation function "image", which can be estimated
- Estimation by image observation
- Estimation by Experimentation
- Estimation by Modeling

Chapter 5
Image Restoration: **Inverse and Pseudo-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.

Inverse and Pseudo-inverse Filtering Examples

a b
c d

FIGURE 5.27 Restoring Fig. 5.25(b) with Eq. (5.7-1). (a) Result of using the full filter. (b) Result with *H* cut off outside a radius of 40; (c) outside a radius of 70; and (d) outside a radius of 85.

Butterworth low pass filtering is used here

Wiener Filtering

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

- It follows from the derivation that the Wiener filter is <u>optimal</u> in the mean square error sense
  - given a particular estimate of degradation function!

- Define *signal-to-noise ratio* as follows:

$$\text{SNR}=\frac{\sum_M\sum_N|F(u,v)|^2}{\sum_M\sum_N|N(u,v)|^2}$$

- The Wiener filter can then be specified as:

$$W(u,v)=\frac{1}{H(u,v)}\left[\frac{|H(u,v)|^2}{|H(u,v)|^2+\text{SNR}^{-1}}\right]$$

- Assuming constant SNR, we can use it as a parameter for the Wiener filter and experimentally find appropriate values.

Wiener Filtering Interpretation

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

- No noise is present $\Longrightarrow$ inverse SNR is zero $\Longrightarrow$ the Wiener filter becomes a simple inverse filter

- No degradation $\Longrightarrow$ $H(u, v) = 1$ $\Longrightarrow$ the Wiener filter becomes a smoothing filter, suppressing areas with low SNR:

$$W(u,v)_{H(u,v)=1} = \frac{\text{SNR}}{\text{SNR}+1}$$

- Otherwise: when both noise and degradation are present, the Wiener filter seeks a compromise between lowpass noise smoothing and high-pass deblurring. The result is a bandpass filter. However, deblurring decreases rapidly as the noise power increases. Experiment with this!

Wiener Filtering Examples

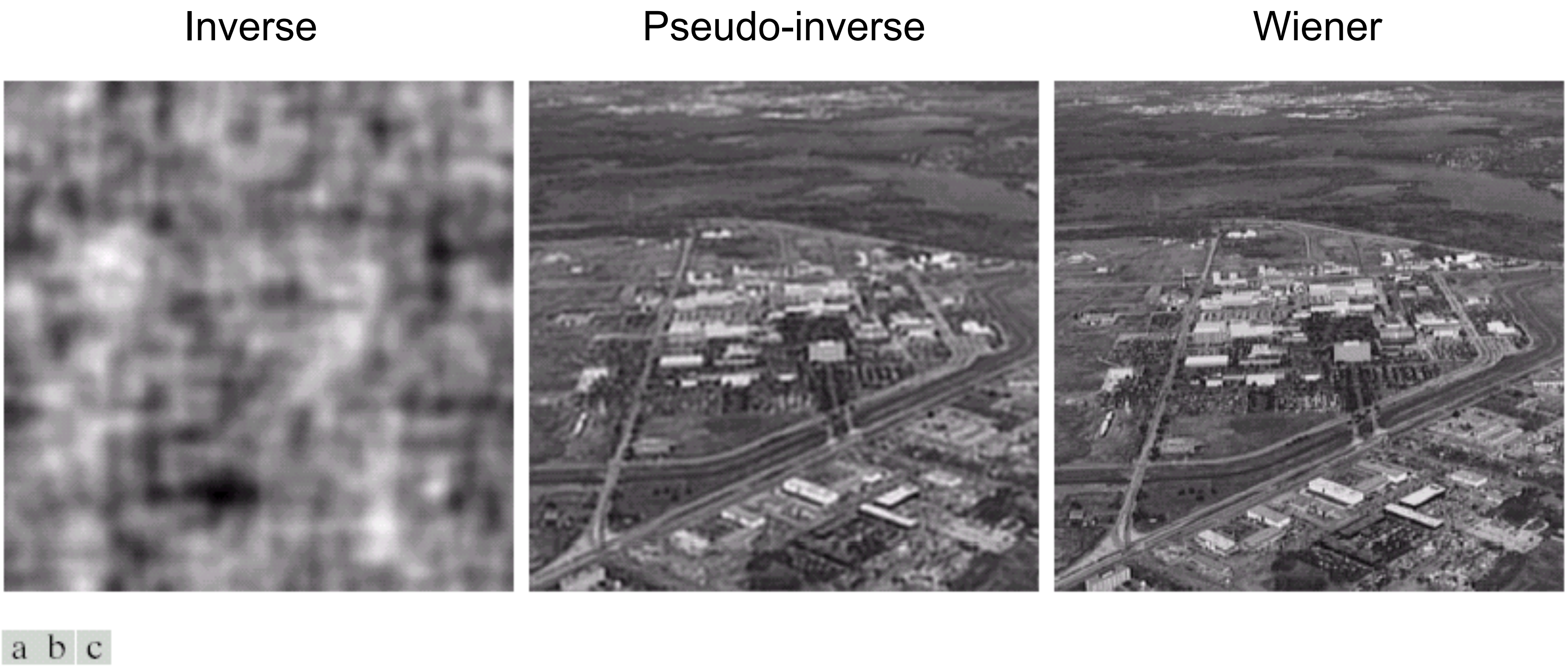
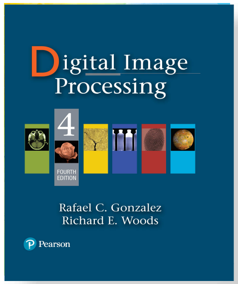


FIGURE 5.28 Comparison of inverse- and Wiener filtering. (a) Result of full inverse filtering of Fig. 5.25(b). (b) Radially limited inverse filter result. (c) Wiener filter result.

Geometric Mean Filter

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

- Further generalization of the Wiener filter
- Useful for software implementation, as it covers a multitude of possible filters, such as:
  - $\alpha = 1 \Longrightarrow$ <u>inverse filter</u>
  - $\alpha = 0 \Longrightarrow$ <u>parametric Wiener filter</u> (standard if $\beta = 1$)
  - $\alpha = ½ \Longrightarrow$ geometric mean of the two filters
  - $\alpha = ½$ and $\beta = 1 \Longrightarrow$ <u>*spectral equalization filter*</u>



Chapter 7
Color Image Processing

- Color image representation
- Brightness response of the cones (tristimulus functions)
- Primary and secondary color of light/pigment
- The chromaticity diagram (color gamut)
- Color models, color conversion, color manipulations
- Color image processing
    - Pseudo-coloring via
        - intensity slicing
        - color transformations, and
        - Multispectral imaging
    - Full-color image processing
        - Intensity and color component manipulation
        - Contract enhancement and tonal correction via histogram equalization
        - Color image smoothing and sharpening
        - Color segmentation
        - Computing gradients
        - Noise filtering

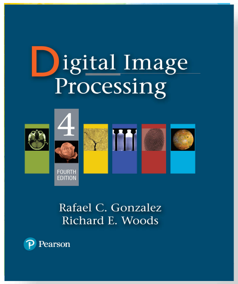
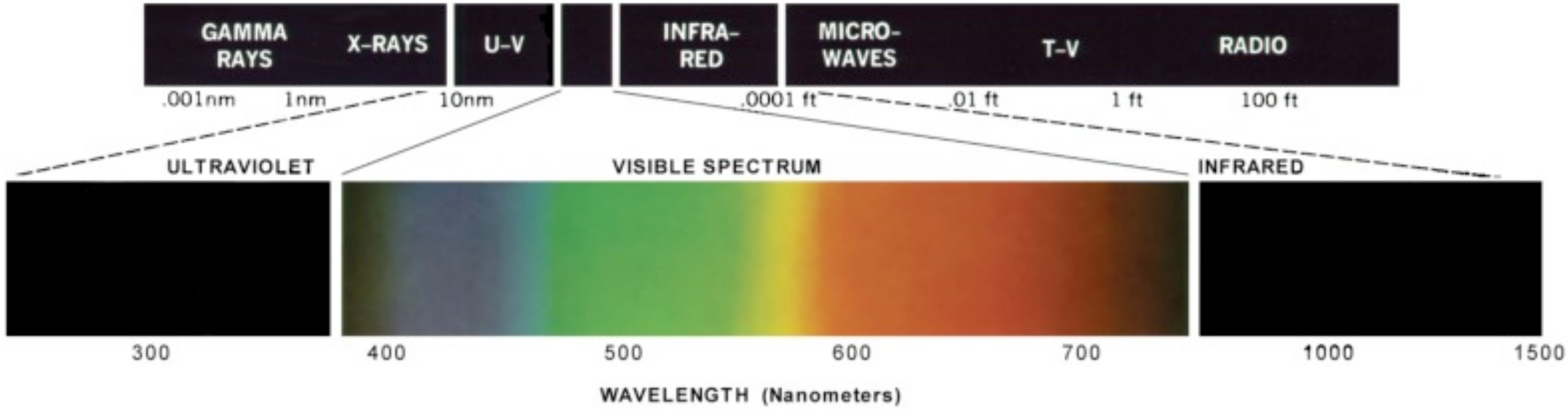


FIGURE 7.2
Wavelengths comprising the visible range of the electromagnetic spectrum. (Courtesy of the General Electric Co., Lighting Division.)

Electromagnetic Spectrum



Chapter 7
Color Image Processing:
Color Image Representation

Color images can be represented by an intensity function $C(x,y,\lambda)$ which depends on the wavelength λ of the reflected light. (so, for fixed λ, $C(x,y,\lambda)$ represents a monochrome image).

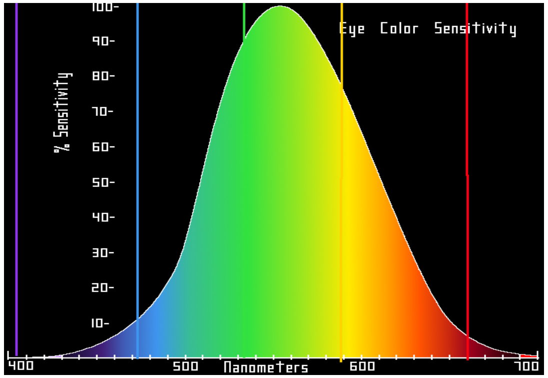
As in the monochrome case, $0 < C(x,y,\lambda) < C_{max}$

The brightness response of a human observer to an image will therefore be

$$f(x,y) = \int_0^\infty C(x,y,\lambda)\, V(\lambda)\, d\lambda$$

where $V(\lambda)$ is the response factor of the human eye at frequency λ, and it is called the relative luminous efficiency function of the visual system.

For the human eye, $V(\lambda)$ is a bell-shaped function.



Human Eye

- The human eye senses this spectrum using a combination of photoreceptor cells, called Rods and Cones to make vision possible.
- **Rod cells are effective for low-intensity light vision**, but can only sense the **intensity of light**, whereas **cone cells can discern color**, but they function best in **bright light**.
- There are 6-7 million Cones (some say 5M) and a. 100 million Rods in the human eye
- **Three types of cone cells exist in your eye**, with each being more sensitive to either short (S), medium (M), or long (L) wavelength light.
- These principal sensing categories correspond roughly to **red**, **green** and **blue**: (65% of cones are sensitive to red, 33% to green and 2% to blue)

We have three brightness response functions (tristimulus functions) describing the absorption of light in red, green and blue cones as measured by the relative sensitivity of each type of cone receptors.

$$f_R(x,y) = \int_0^\infty C(x,y,\lambda)\, V_R(\lambda)\, d\lambda$$

$$f_G(x,y) = \int_0^\infty C(x,y,\lambda)\, V_G(\lambda)\, d\lambda$$

$$f_B(x,y) = \int_0^\infty C(x,y,\lambda)\, V_B(\lambda)\, d\lambda$$

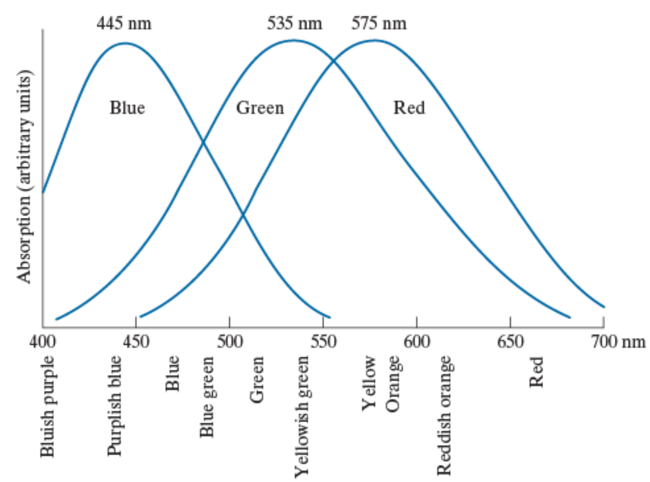
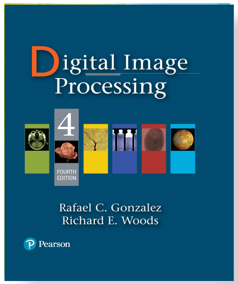


FIGURE 7.3 Absorption of light by the red, green, and blue cones in the human eye as a function of wavelength.

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

In **1931**, CIE (International Commission on Illumination) designated the following:

Blue = 435.8nm Green = 546.1nm Red = 700nm



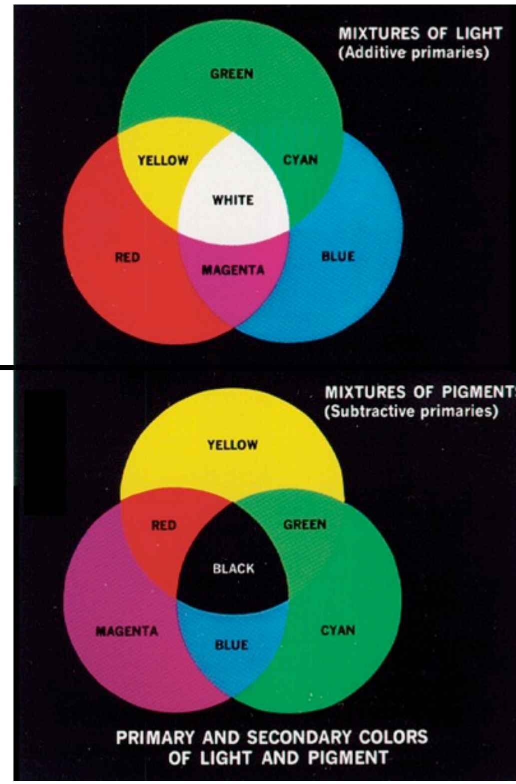
Primary colors of light can be added in pairs to produce secondary colors of light: e.g. magenta, cyan and yellow.

Mix all three primary colors in light 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.

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

Mix all three primary colors of pigment produces black color

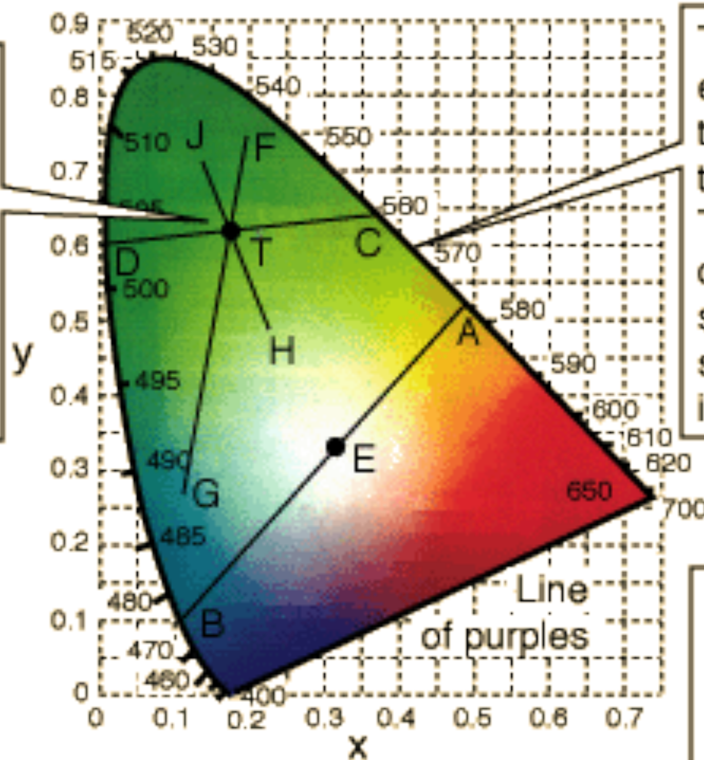


a
b

FIGURE 7.4 Primary and secondary colors of light and pigments. (Courtesy of the General Electric Co., Lighting Division.)

Chapter 7
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.



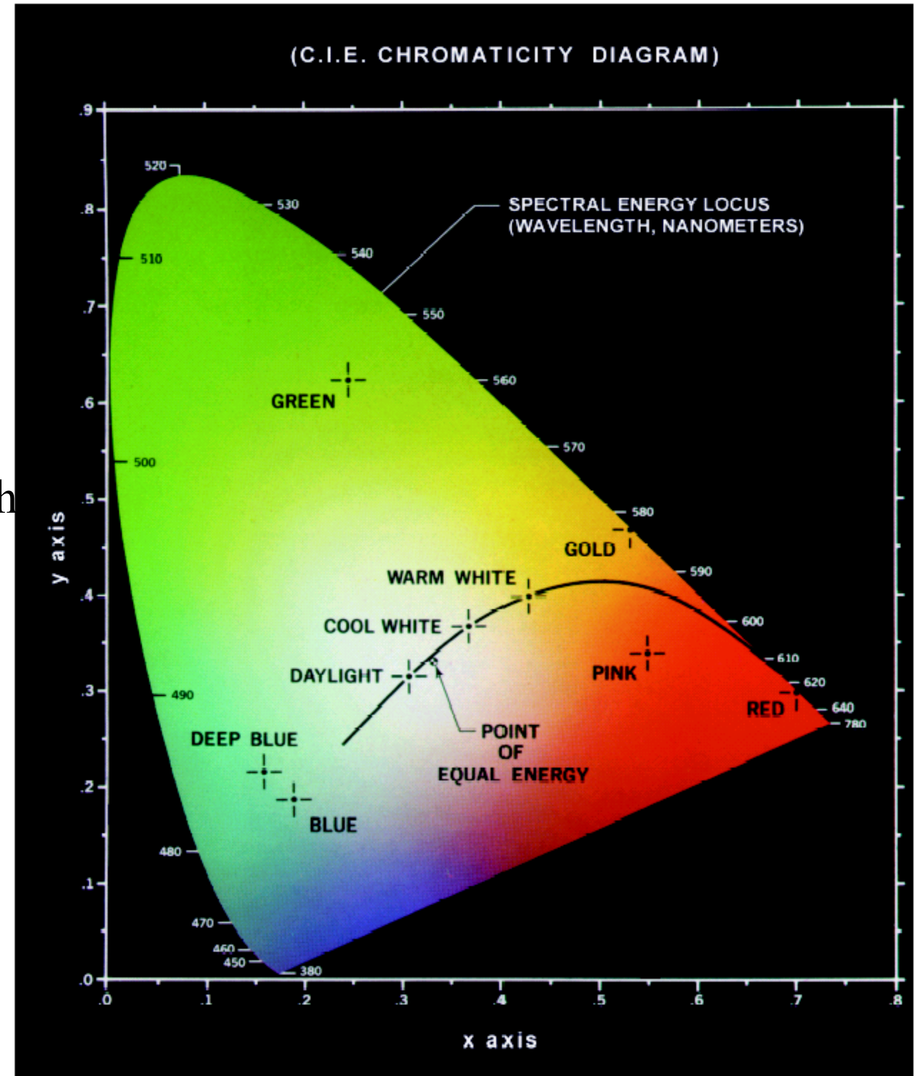
The solid line outline encompasses all the hues that are perceivable to the normal human eye. The horseshoe shaped curve contains the spectral colors. The straight line at the bottom is the line of purples.

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 7
Color Image Processing

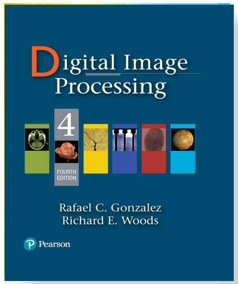
FIGURE 6.5 Chromaticity diagram. (Courtesy of the General Electric Co., Lamp Business Division.)



- Pure colors are mapped on the boundary of the chromaticity diagram, fully saturated colors
- Colors inside the diagram are combinations of these colors
- Reference white is the point of equal energy, with zero saturation value

The diagram is useful for color mixing, e.g.

- a straight line joining any two points defines all colors generated by adding those colors,
- in particular, if one of these points is reference white and the other is some color on the boundary, then the colors on the line in-between represent all the shades of that particular spectrum color

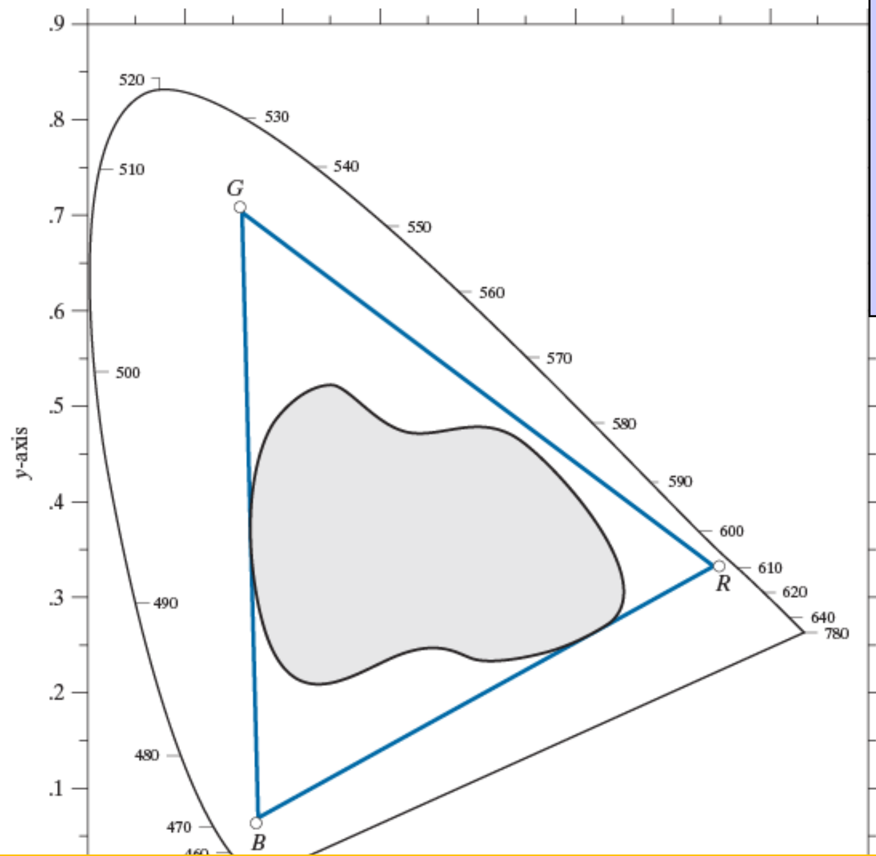


Chapter 7

Color Image Processing

FIGURE 7.6 Illustrative color gamut of color monitors (triangle) and color printing devices (shaded region).

Typical color gamut of an RGB display



color gamut of a high quality color printer, irregular shape is due to additive and subtractive color combinations

Remember that due to the shape of the chromaticity diagram, no fixed three colors can reproduce all colors inside the diagram!

Chapter 7
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, HSV (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:

❖ Conversion between color models can be either linear or nonlinear,

❖ Some models can be more useful as they can decouple color and gray-scale components of a color image, e.g. HSI, YUV.

“Additive” vs. “Subtractive” Colors

- CMY (Cyan, Magenta, Yellow) used in printing, etc.

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

- *E.g.*, when a cyan painted surface is illuminated by white light, no red is reflected

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:

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

Remarks:

- Recall that all color values have been normalized in the range [0,1]
- A surface coated with cyan does not contain red, that is C = 1 – R
- Equal amounts of 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).

HSI Color Model

Although RGB and CMY color models are very well suited for hardware and RGB reflects well the sensitivity of the human eye to these primary colors, both are not suited for describing color in a way that is easily interpreted by humans.

When humans see a color object, they tend to describe it by its **hue**, **saturation** and **brightness**, i.e. HSI model is used

In addition, HSI decouples brightness from the chroma components.

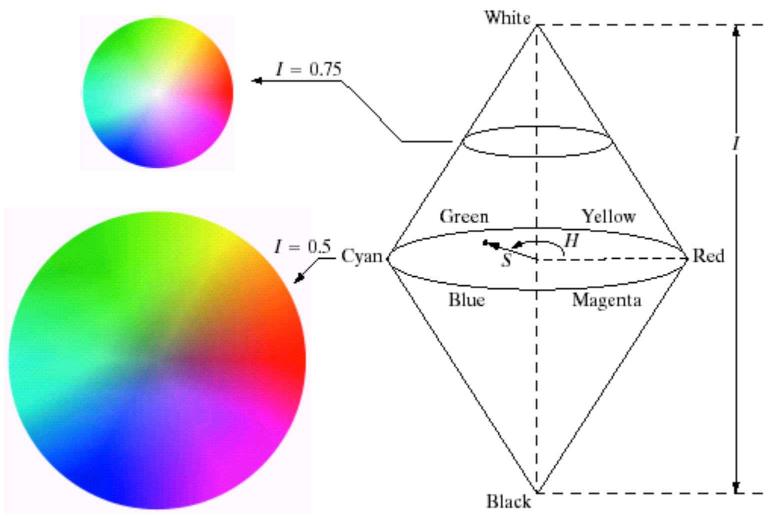
Chapter 7
Color Image Processing: Color Image Representation

Three Perceptual Measures

1. **Brightness I**: 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 H**:* 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 S**: 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.



HSI-RGB Color Model Conversions

From RGB to HSI:

the hue is: $$H = \begin{cases} \theta & if \quad B \le G \\ 360 - \theta & if \quad B > G \end{cases}$$

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

the saturation: $$S = 1 - \frac{3}{(R+G+B)}[\min(R,G,B)]$$

and the intensity is: $$I = \tfrac{1}{3}(R+G+B)$$

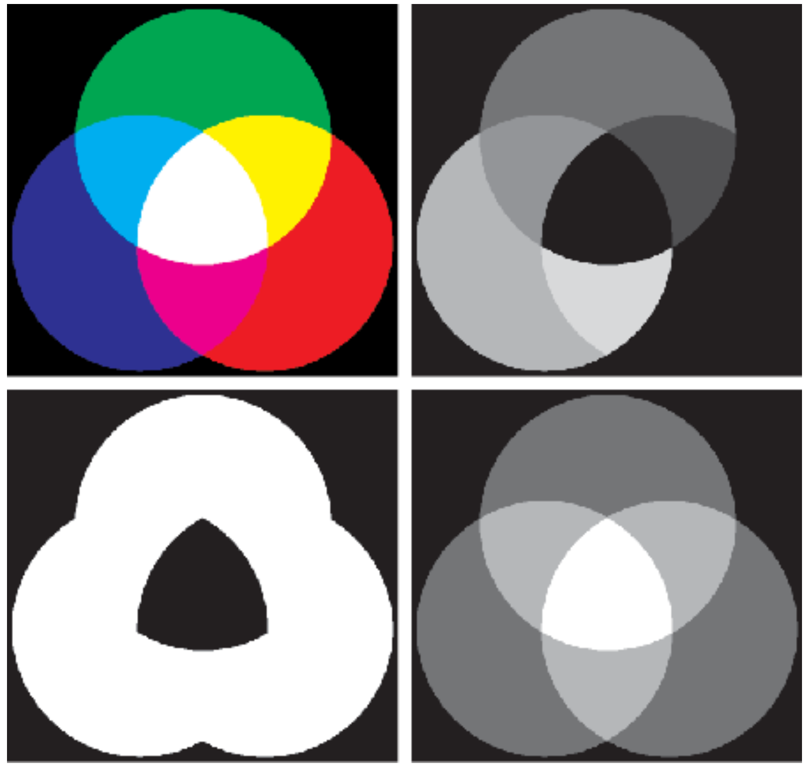
- It's assumed that RGB values are normalized and θ is measured with respect to the Red axis.
- To convert back from HSI to RGB, we need to split the Hue component into 3 sectors (next slide)

Chapter 7

Color Image Processing

Manipulating HSI Component Images

RGB image composed of primary and secondary colors



a b
c d

FIGURE 7.14 (a) RGB image and the components of its corresponding HSI image: (b) hue, (c) saturation, and (d) intensity.

Hue component

Notice how red is mapped to black since its hue is 0 (by convention)

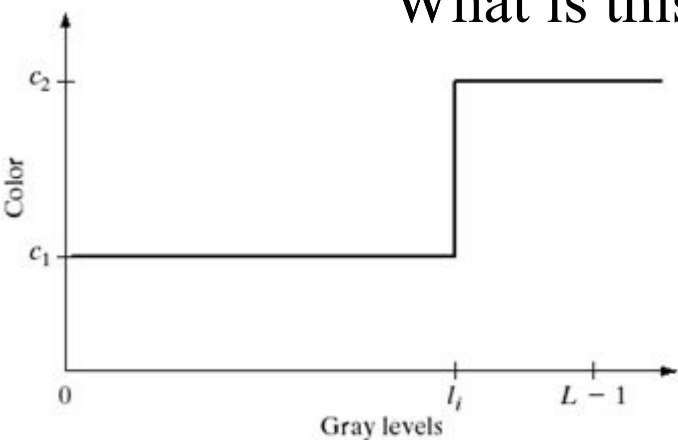
Saturation component

Intensity component

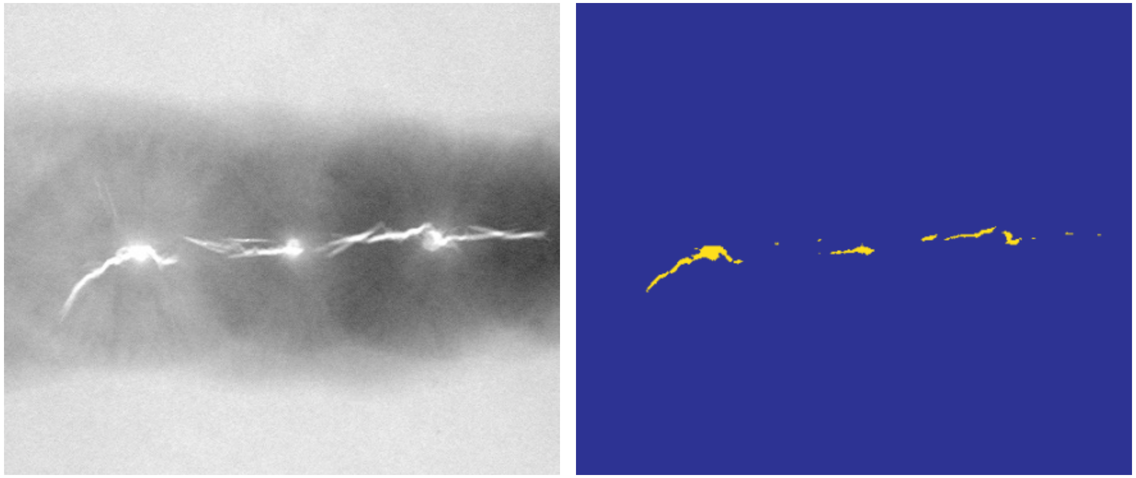
Idea: can change any of these color components independently by changing the gray levels of that component.

Pseudocoloring

What is this transformation doing to the image?



- Intensity slicing
- An example of pseudocoloring

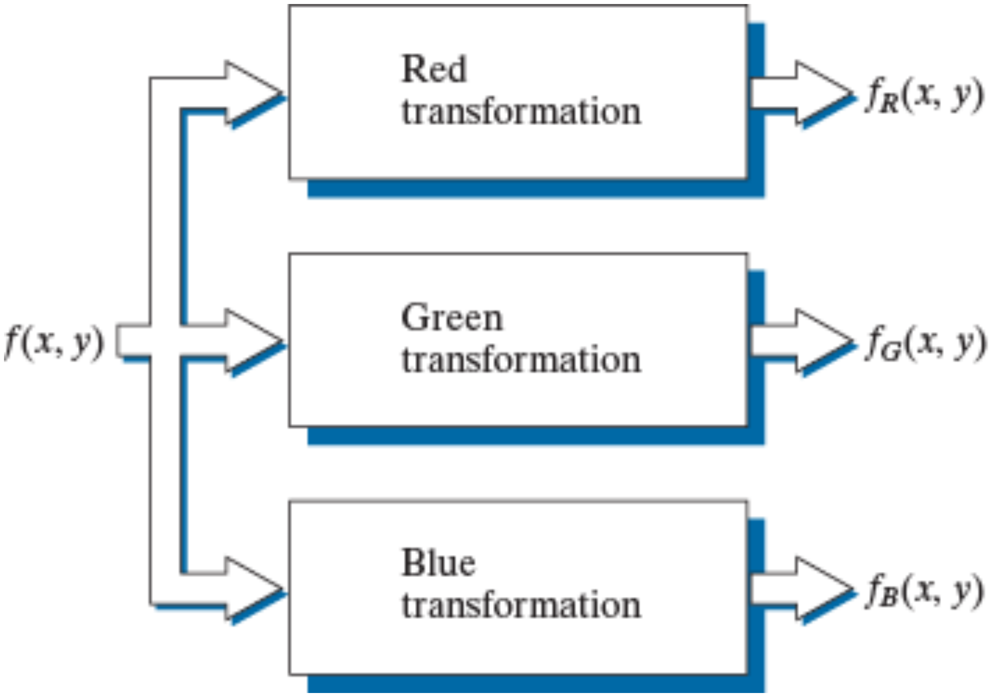


a b

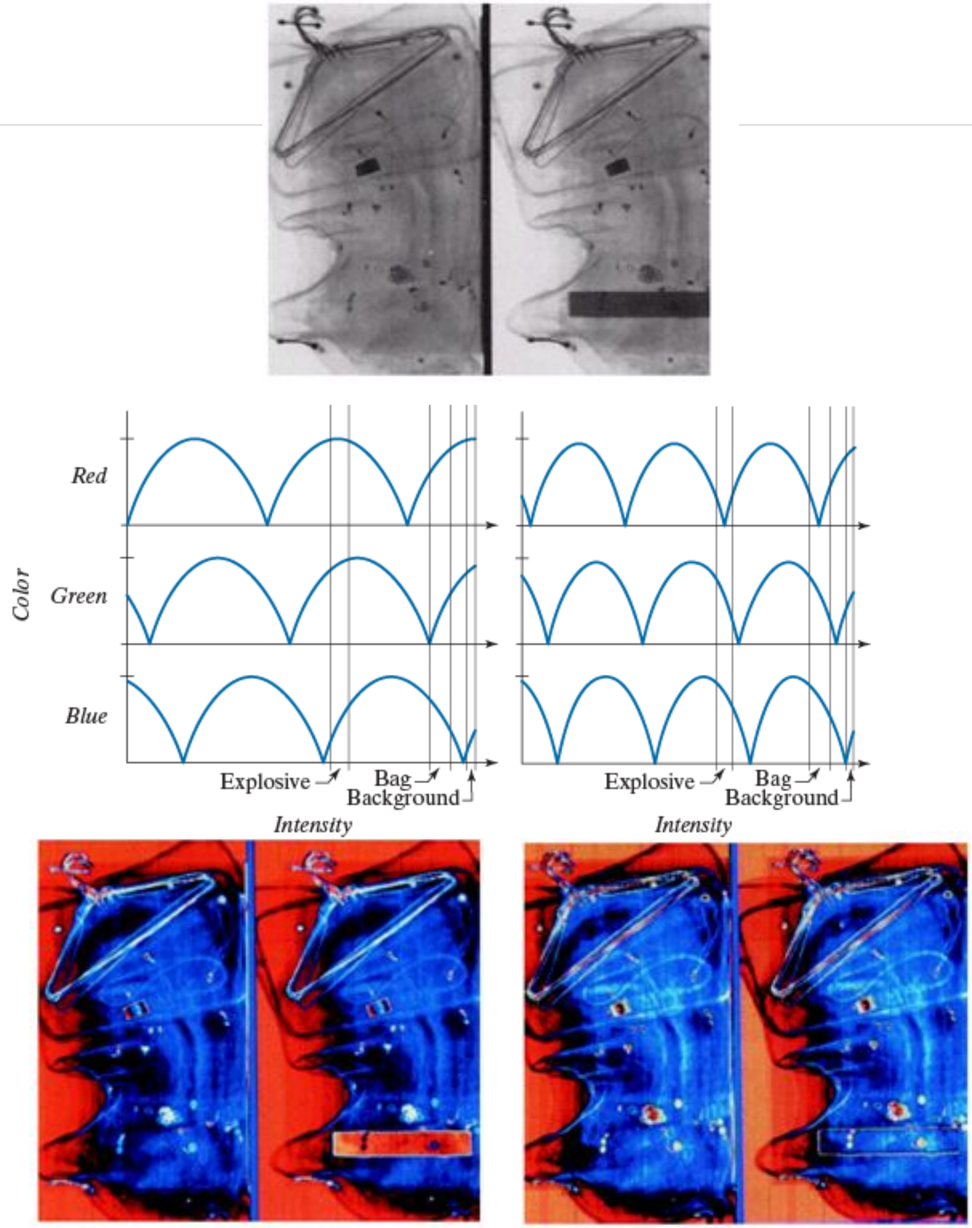
FIGURE 7.19
(a) X-ray image of a weld.
(b) Result of color coding. (Original image courtesy of X-TEK Systems, Ltd.)

Pseudocoloring with color transformations

FIGURE 7.21 Functional block diagram for pseudocolor image processing. Images f_R, f_G, and f_B are fed into the corresponding red, green, and blue inputs of an RGB color monitor.

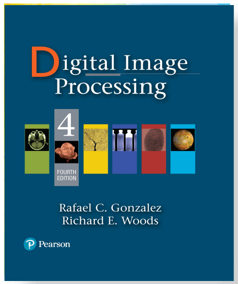


Match the transformation and the pseudocolor image



a b

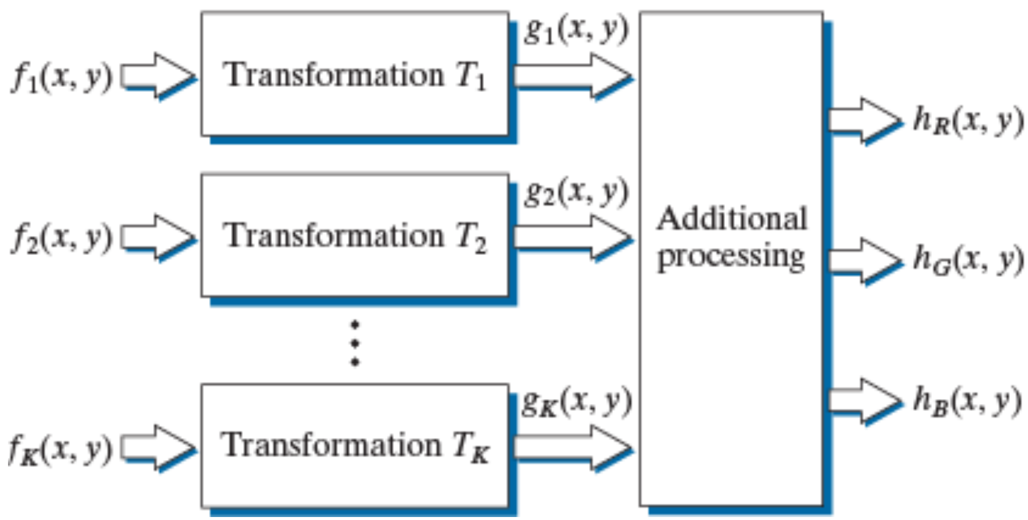
FIGURE 7.23 Transformation functions used to obtain the pseudocolor images in Fig. 7.22.



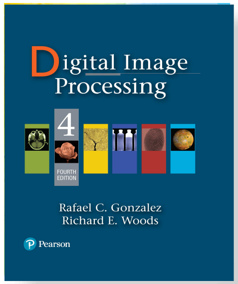
Pseudocoloring for Multispectral Images

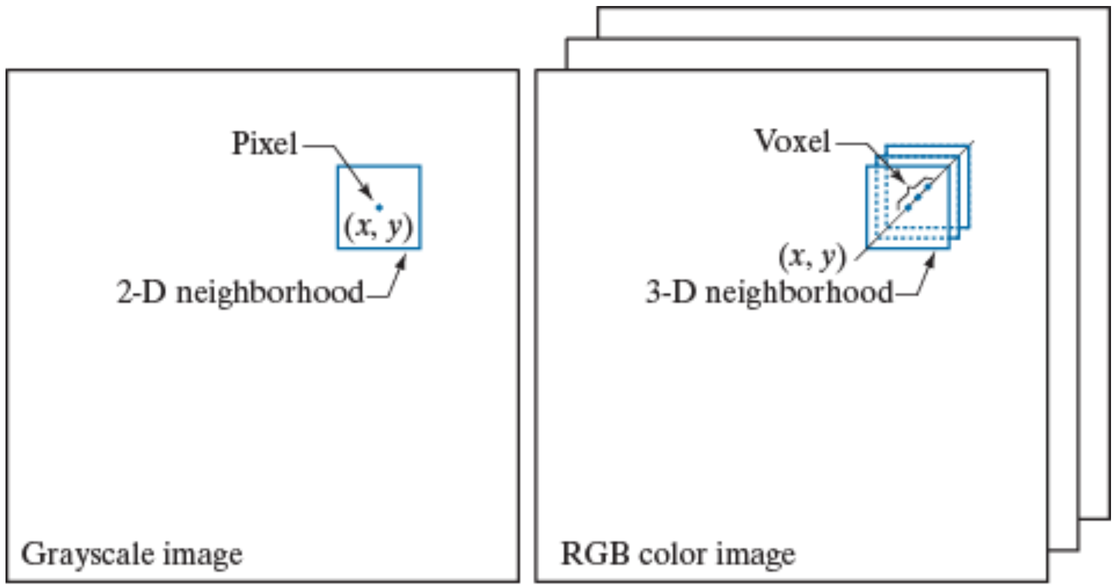
- Multispectral images have been acquired by different sensors at different wavelengths.
- Combining them to obtain a color image can be achieved as follows:

FIGURE 7.24 A pseudocolor coding approach using multiple grayscale images. The inputs are grayscale images. The outputs are the three components of an RGB composite image.



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





a b
FIGURE 7.27 Spatial neighborhoods for grayscale and RGB color images. Observe in (b) that a *single* pair of spatial coordinates, (x, y), addresses the same spatial location in all three images.

a color image is multi-valued, i.e. in RGB, each pixel has 3 values

Consider the following color transformation:

$$g(x,y) = T[f(x,y)]$$

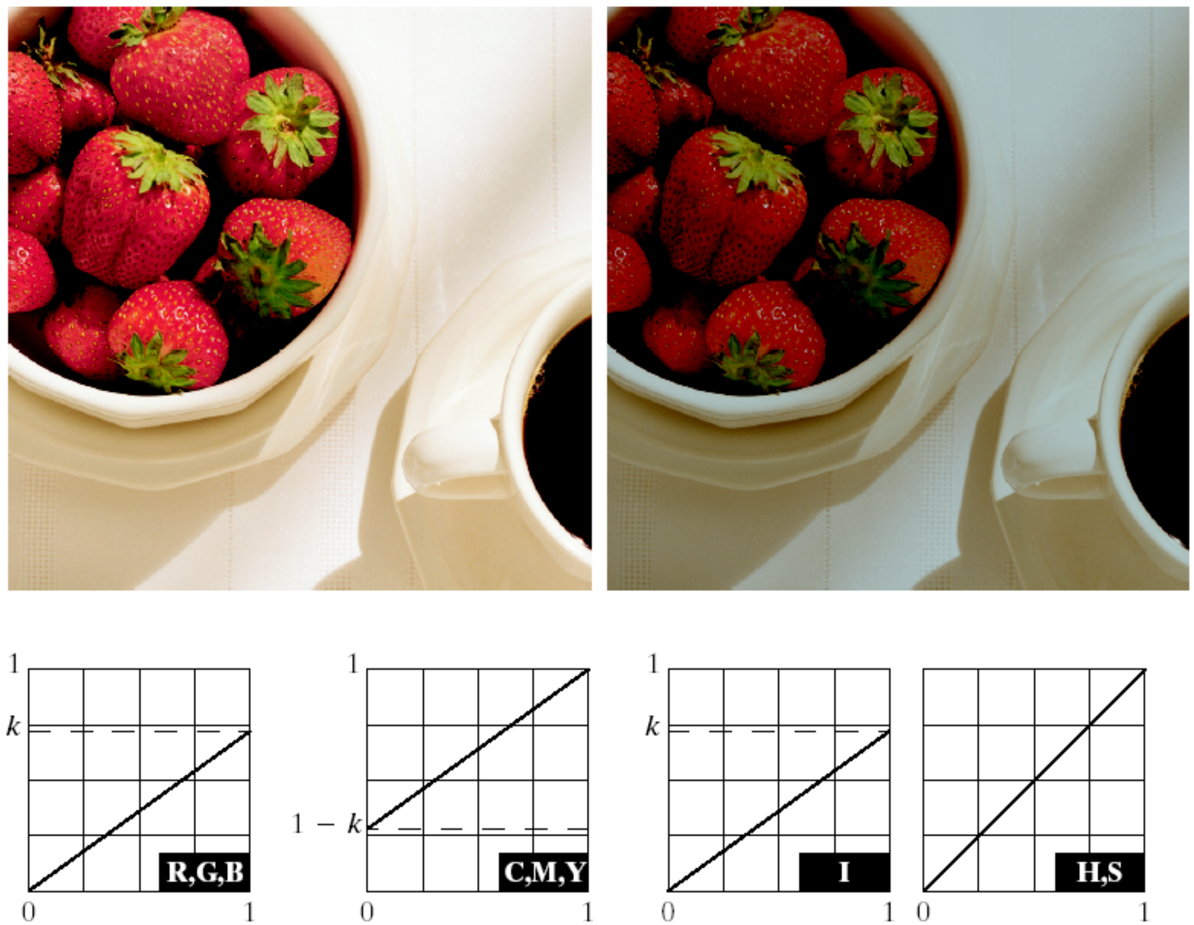
Let r_i and s_i denote the color components of f and g, respectively,

$$s_i = T_i(r_1, r_2, \ldots, r_n) \quad where\ n = 3\ or\ 4$$

How can you decrease the image intensity by 30%?

a b
c d e

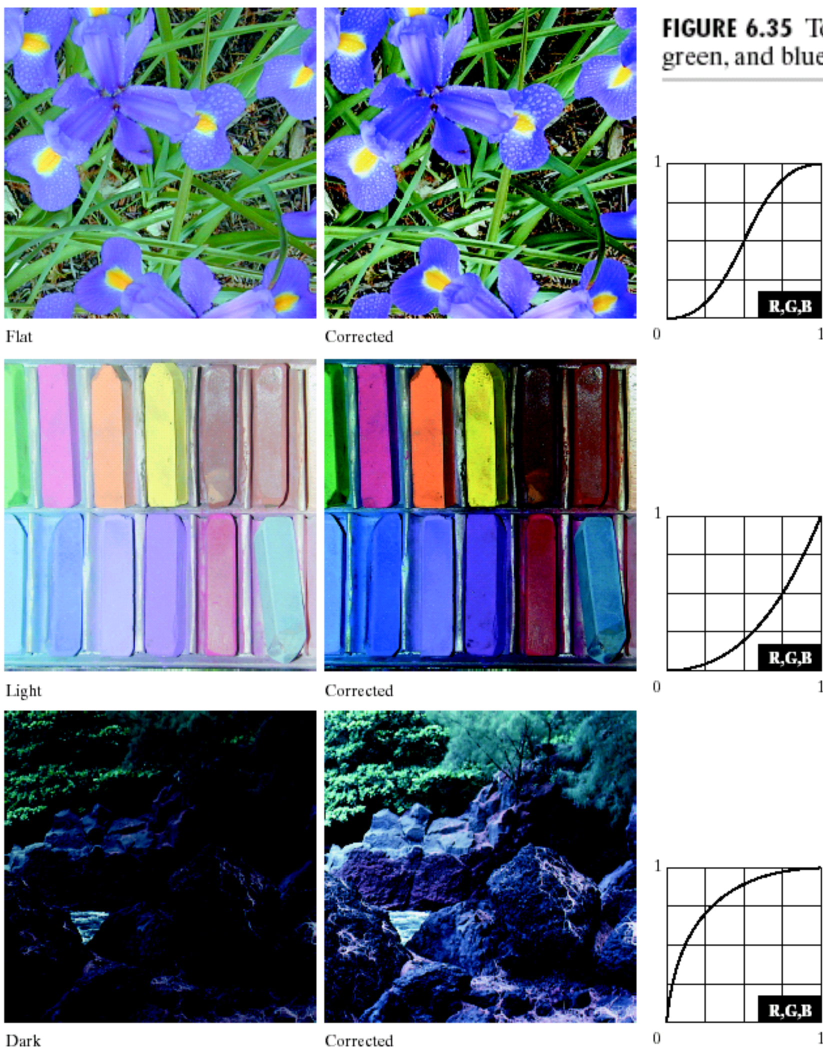
FIGURE 6.31 Adjusting the intensity of an image using color transformations. (a) Original image. (b) Result of decreasing its intensity by 30% (i.e., letting $k = 0.7$). (c)–(e) The required RGB, CMY, and HSI transformation functions. (Original image courtesy of MedData Interactive.)



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

Color image histogram equalization for tonal correction

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.

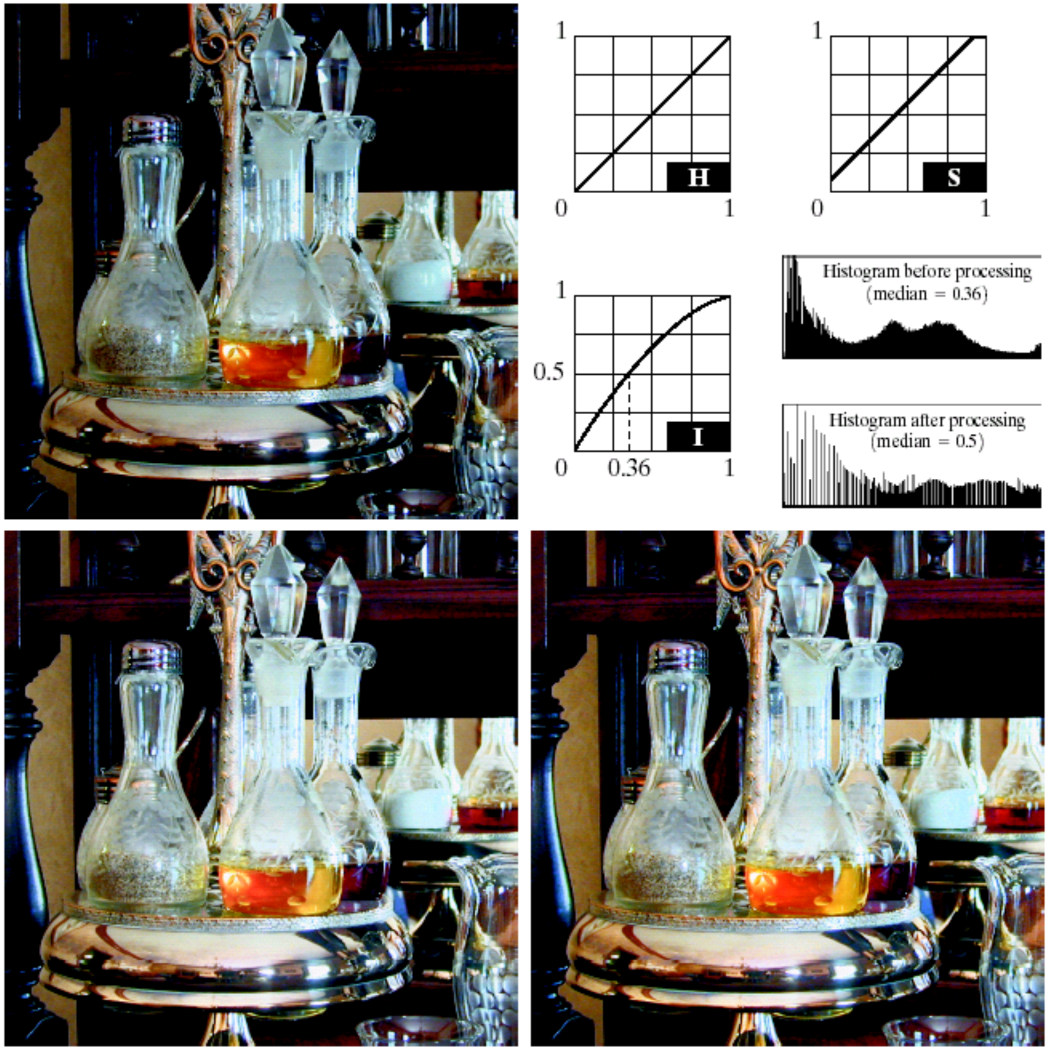


S-shaped transformation for boosting contrast

Power-law-like transformation to correct light and dark details, as in grey-scale images.

Histogram Equalization for Color Images

(a) original



a b
c d

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

(c) Equalized image by modifying only I.

(d) Equalized image by modifying I and slightly S.

Two remarks:

1) Notice the effect on the histogram,
2) Notice the liquid color, it's not as crisp as in the original.

Color Image Processing

a	b
c	d

FIGURE 7.36 (a) RGB image. (b) Red component image. (c)Green component. (d) Blue component.

Color Image Smoothing

Smoothing all color components versus smoothing only intensity component:

a b c

FIGURE 7.38 Image smoothing with a 5×5 averaging kernel. (a) Result of processing each RGB component image. (b) Result of processing the intensity component of the HSI image and converting to RGB. (c) Difference between the two results.

Note that (a) is smoother and lost some of its original colors, in contrast to (b) which preserved its hue and saturation.

Color Image Sharpening

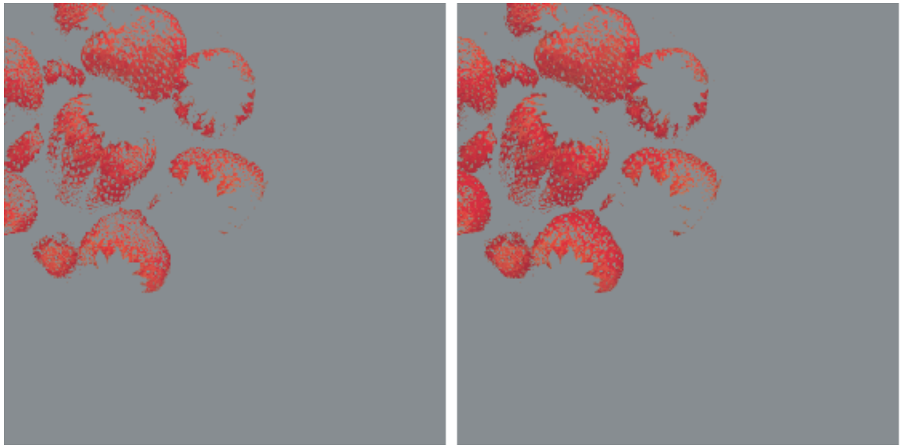
a b c

FIGURE 7.39 Image sharpening using the Laplacian. (a) Result of processing each RGB channel. (b) Result of processing the HSI intensity component and converting to RGB. (c) Difference between the two results.

Color Segmentation

How to keep red colors only and make everything else grey?

Remember color slicing!

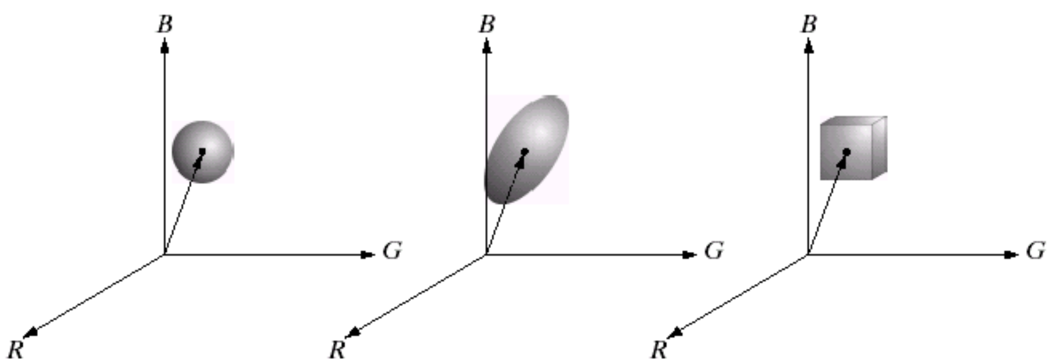


a b

FIGURE 7.32 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 $(0.5, 0.5, 0.5)$.

Segmentation in RGB Color Space
An extension of color slicing

Suppose that regions of specific color range are to be segmented. The specific color is specified by an average color **a** and a neighborhood around it, defined by a suitable distance measure: we say that **z** is similar to **a** if D(**a**,**z**) is smaller than a threshold D_0.



a b c

~~FIGURE 6.43~~
Three approaches for enclosing data regions for RGB vector segmentation.

(a) Euclidean distance (most general)
(b) Malahanobis distance (take into account properties of the data)
(c) Bounding box (reduce computational complexity)

Segmentation in HSI Color Space

a b
c d
e f
g h

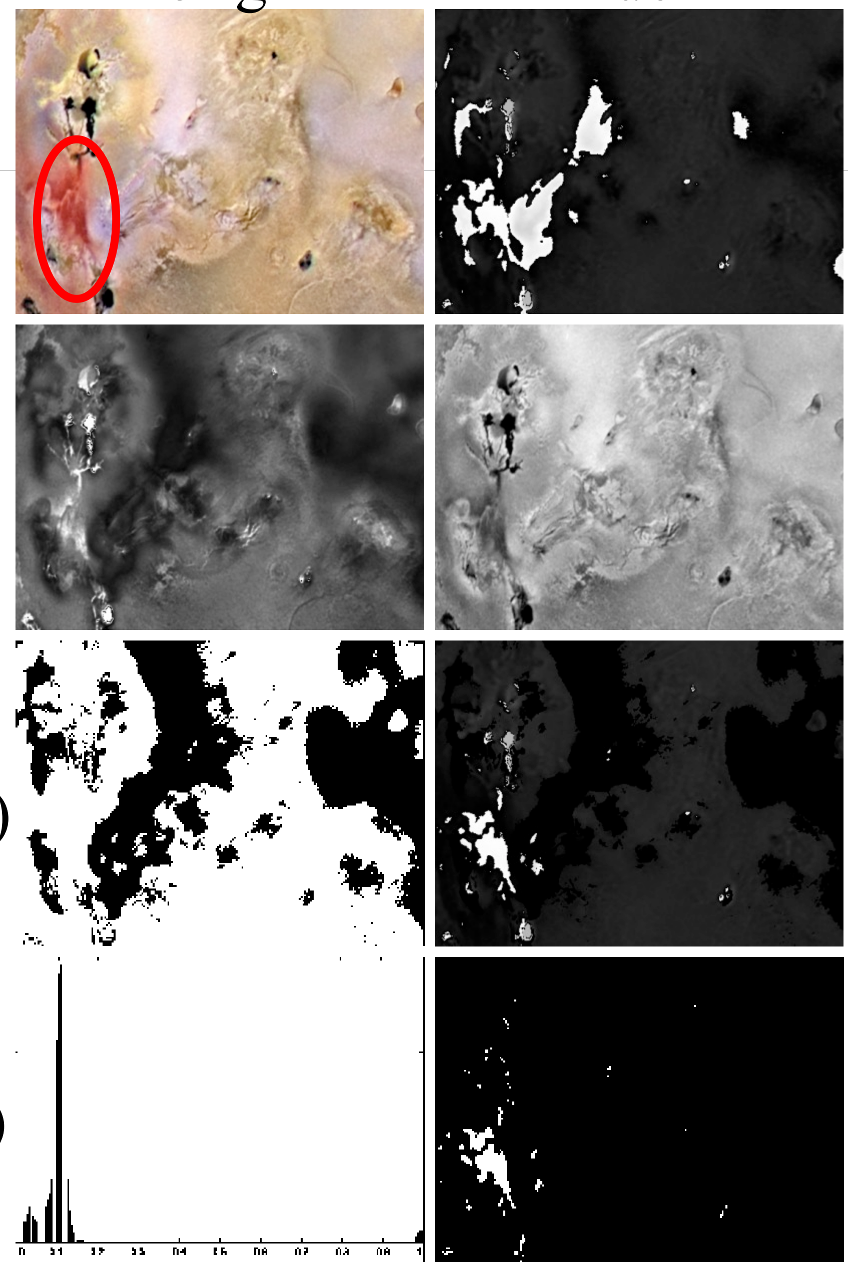
FIGURE 7.40 Image segmentation in HSI space. (a) Original. (b) Hue. (c) Saturation. (d) Intensity. (e) Binary saturation mask (black = 0). (f) Product of (b) and (e). (g) Histogram of (f). (h) Segmentation of red components from (a).

Problem:
Segment the reddish region in the lower left side of the image.

Proposed solution:
The region of interest has high **hue** values in the blue-magenta side of Red.
Let us prepare a mask by thresholding the saturation at 10% of the max value (e), and
Multiply the hue and thresholded saturation to enhance the areas of high saturation in the hue.
Then, we take the histogram of (f) and notice the high values of the reddish pixels near 1 (g).
Finally, segment in pixel values in (a) to obtain the result in (h).

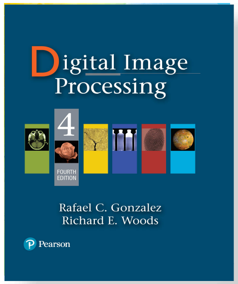
original hue

saturation intensity



(e) (f)

(g) (h)



Noise Filtering in Color Images

a	b
c	d

FIGURE 7.46 (a)–(c) Red, green, and blue 8-bit component images corrupted by additive Gaussian noise of mean 0 and standard deviation of 28 intensity levels. (d) Resulting RGB image. [Compare (d) with Fig. 7.44(a).]

Consider the RGB components, each was corrupted with Gaussian noise (0,28).

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

Noise Filtering in Color Images

Now, convert the same image to HSI and look at the components!

a b c

FIGURE 7.47 HSI components of the noisy color image in Fig. 7.46(d). (a) Hue. (b) Saturation. (c) Intensity.

This is due to the nonlinearities in the conversion between RGB and HSI. The intensity component I does not look bad, why?

Noise Filtering in Color Images

a b
c d

FIGURE 7.48
(a) RGB image with green plane corrupted by salt-and-pepper noise.
(b) Hue component of HSI image.
(c) Saturation component.
(d) Intensity component.

Suppose that only one of the RGB channel is corrupted with noise, say the green channel, converting this to HSI yields a high degradation in the hue and saturation components.

Basics of Digital Video

References:

1. Chapters 1 and 6 (specifically sections 6.1, 6.2, 6.4.1, and 6.4.4): Video Formation, Perception, and Representation, in Video Processing and Communications, by Wang, Ostermann and Zhang

2. Ostermann et al. Video coding with H.264/AVC: Tools, performance, and complexity, IEEE Circuits and Systems Magazine, 1st Quarter, 2004

Outline

- Video capture and display
- Raster scan

- Analog Color TV broadcasting
- RGB to YIQ and YC1C2 conversions
- Digital Video
- Three types of redundancies and where they are exploited in image and hybrid video compression
- Temporal redundancy via prediction
- Motion estimation and motion compensation
- Block Matching Algorithm (BMA)
- Motion compensated prediction
- Reducing artifacts (blocking and ringing)
- Video encoders

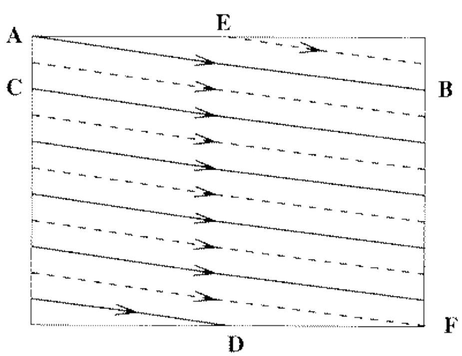
Video Capture and Display

Involves the following components:

- Light reflection physics
- Imaging operator
- Color capture
- Color display
- Component vs. composite video

Raster Scan

- Real-world scene is a continuous monochrome 3-D signal (temporal, horizontal, vertical)
- Analog video is stored in the raster format
  - Sampling in time: consecutive sets of frames
    - To render motion properly, >=30 frame/s is needed
  - Sampling in vertical direction: a frame is represented by a set of scan lines
    - Number of lines depends on maximum vertical frequency and viewing distance, 525 lines in the NTSC system
  - Video-raster = 1-D signal consisting of scan lines from successive frames



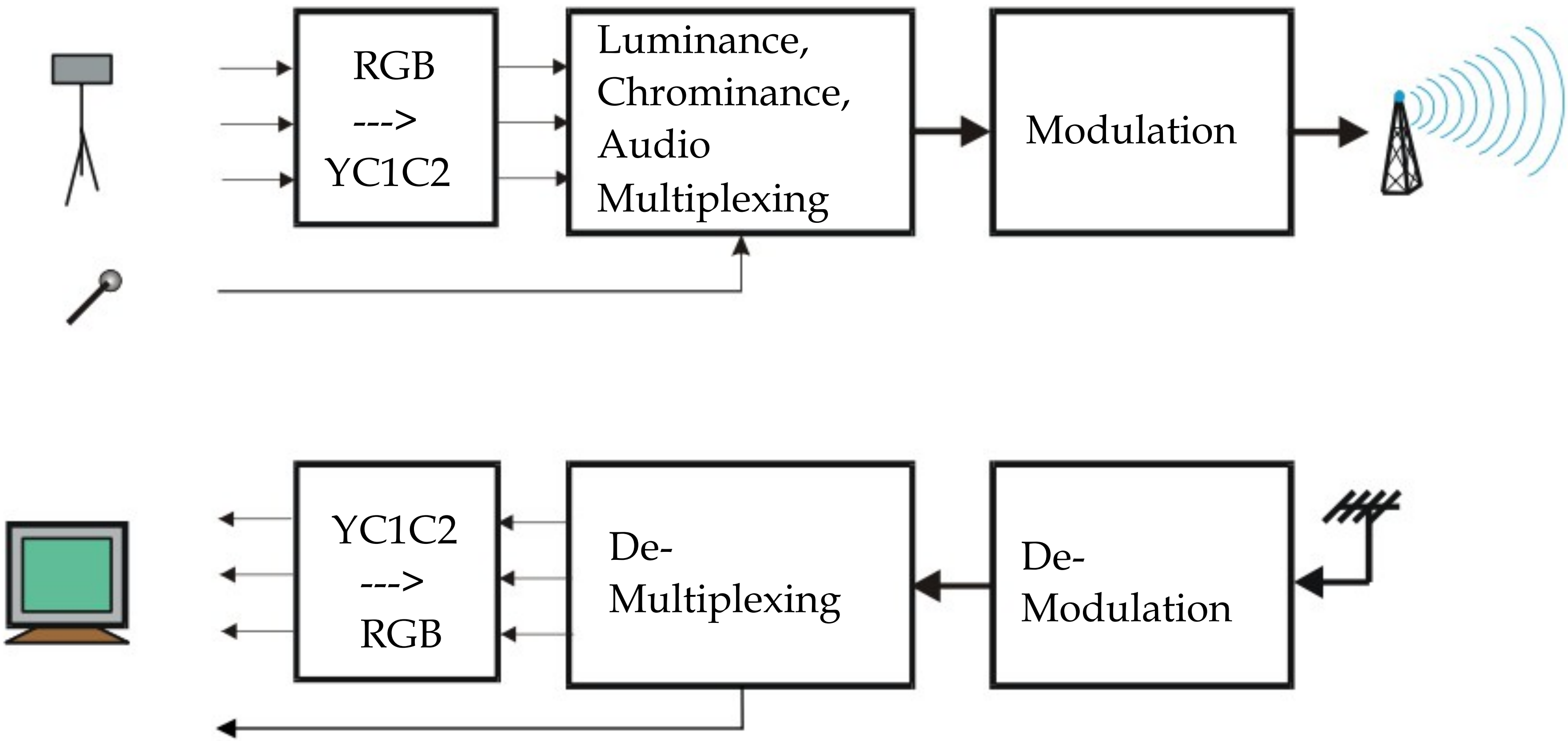
NTSC: National Television System Committee

Progressive versus Interlaced Videos

- Video raster: Progressive vs. interlaced raster
- Analog TV systems

- **Progressive**
  - Every pixel on the screen is either refreshed in a sequential order (monitors) or simultaneously (films)
- **Interlaced**
  - Each frame is refreshed twice: the little gun at the back of the CRT shoots all the correct phosphors on the even numbered rows of pixels first and then odd numbered rows
  - NTSC frame-rate of 29.97 means the screen is redrawn 59.94 times a second
  - In other words, 59.94 half-frames per second or 59.94 fields per second

Color TV Broadcasting and Receiving



Why not using RGB directly?

- R,G,B components are correlated (How to verify this?)
  - Transmitting R,G,B components separately is redundant
  - More efficient use of bandwidth is desired
- RGB->YC1C2 transformation (recall YUV for analog)
  - Decorrelating: Y,C1,C2 are uncorrelated
  - C1 and C2 require lower bandwidth (WHY? See s.33)
  - Y (luminance) component can be received by B/W TV sets
- YIQ in NTSC
  - I: orange-to-cyan
  - Q: green-to-purple (human eye is less sensitive)
    - Q can be further band-limited than I
  - Hue = Arctan(Q/I) (Phase); Saturation =SQRT(I^2+Q^2) (Magnitude)
  - Hue is better retained compared to Saturation

Y: luminance
C1 (or Cb): blue difference
C2 (or Cr): red difference
YCbCr is a rotated and scaled version of RGB

Conversion between RGB and YIQ, YCbCr

RGB -> YIQ

Y = 0.299 R + 0.587 G + 0.114 B

I = 0.596 R - 0.275 G - 0.321 B

Q = 0.212 R - 0.523 G + 0.311 B

YIQ -> RGB

R = 1.0 Y + 0.956 I + 0.620 Q

G = 1.0 Y - 0.272 I - 0.647 Q

B = 1.0 Y - 1.108 I + 1.700 Q

RGB -> YCbCr

Y= 16 + 65.738*R/256 + 129.057*G/256 + 25.064*B/256

Cb = 128-37.945*R/256 - 74.494*G/256 + 112.439*B/256

Cr = 128+112.439*R - 94.154*G/256 - 18.285*B/256

YCbCr -> RGB

R = 298.082*Y/256 + 408.583 * Cr/256 - 222.921

G = 298.082*Y/256 - 100.291*Cb/256 - 208.120*Cr/256 + 135.576

B = 298.082*Y/256 + 516.412*Cb/256 - 276.836

Why Digital Video

"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 after proper coding/compression

- Earlier, it was quite impressive how a DVD can store a three-hour movie, but now mobile phones come with 1TB storage)
- Transmission of high-quality video through network in reasonable time

Digital Video Coding Principles

Still-Image Compression

- Still-image compression
  - Still-image techniques provide a basis for video compression
  - Video can be compressed using still-image compression individually on each frame
  - E.g., "Motion JPEG" or MJPEG
- But modern video codecs go well beyond this
  - Start with still-image compression techniques
  - Add motion estimation/compensation
    - Takes advantage of similarities between frames in a video sequence

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

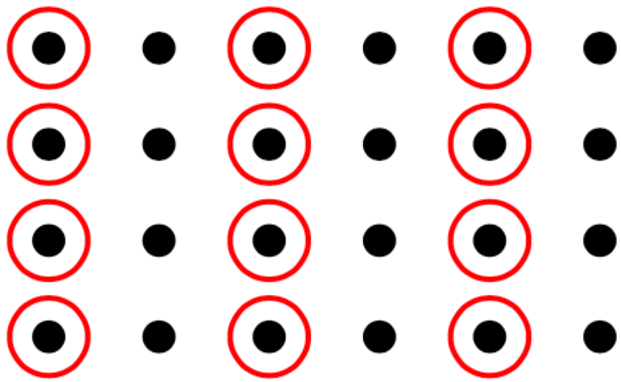
Issues to be handled

- 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 used values are encoded with fewer bits!)
  - constant and variable bit-rate requirement

MPEG combines all of these features !!!

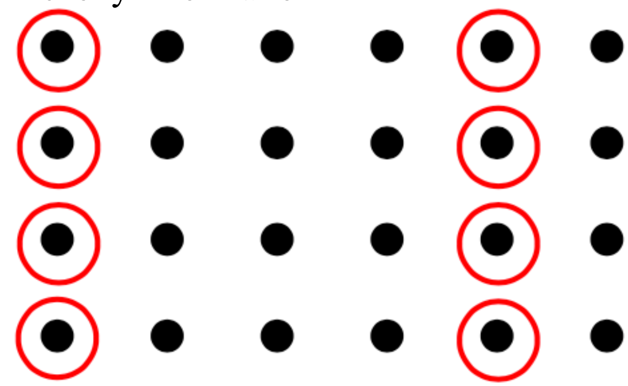
Chrominance Subsampling Formats

- ❖ Humans have a subjectively lower sensitivity to color,
- ❖ Color information is typically sampled at a lower rate than the intensity information



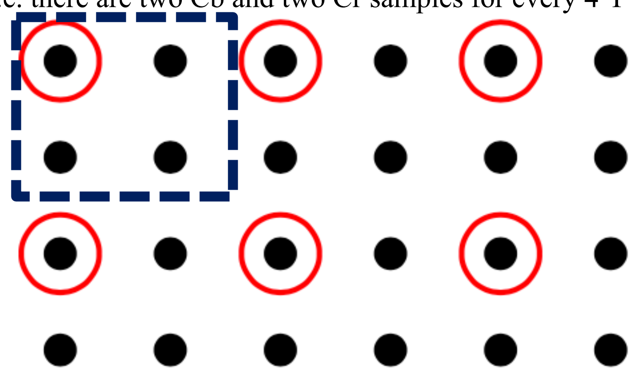
4:2:2

The color information is down-sampled by a factor of 2 horizontally from the full resolution intensity image
i.e. there are two Cb and two Cr samples for every 4 Y samples

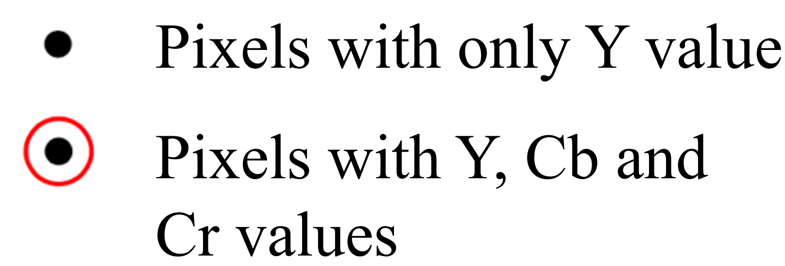


4:1:1

4:1:1 sampling yields 1 Cb and 1 Cr sample for every 4 horizontal Y samples, notice the asymmetric resolution

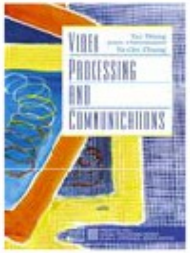


4:2:0



The color information is down-sampled by a factor of 2 horizontally and vertically from the full resolution intensity image, again 1 Cb and 1 Cr for every 4 Y samples, notice the more symmetric resolution

The **4:4:4** sampling structure represents video in which the color components of the signal are sampled at the same rate as the luminance signal



Key Ideas in Video Compression

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

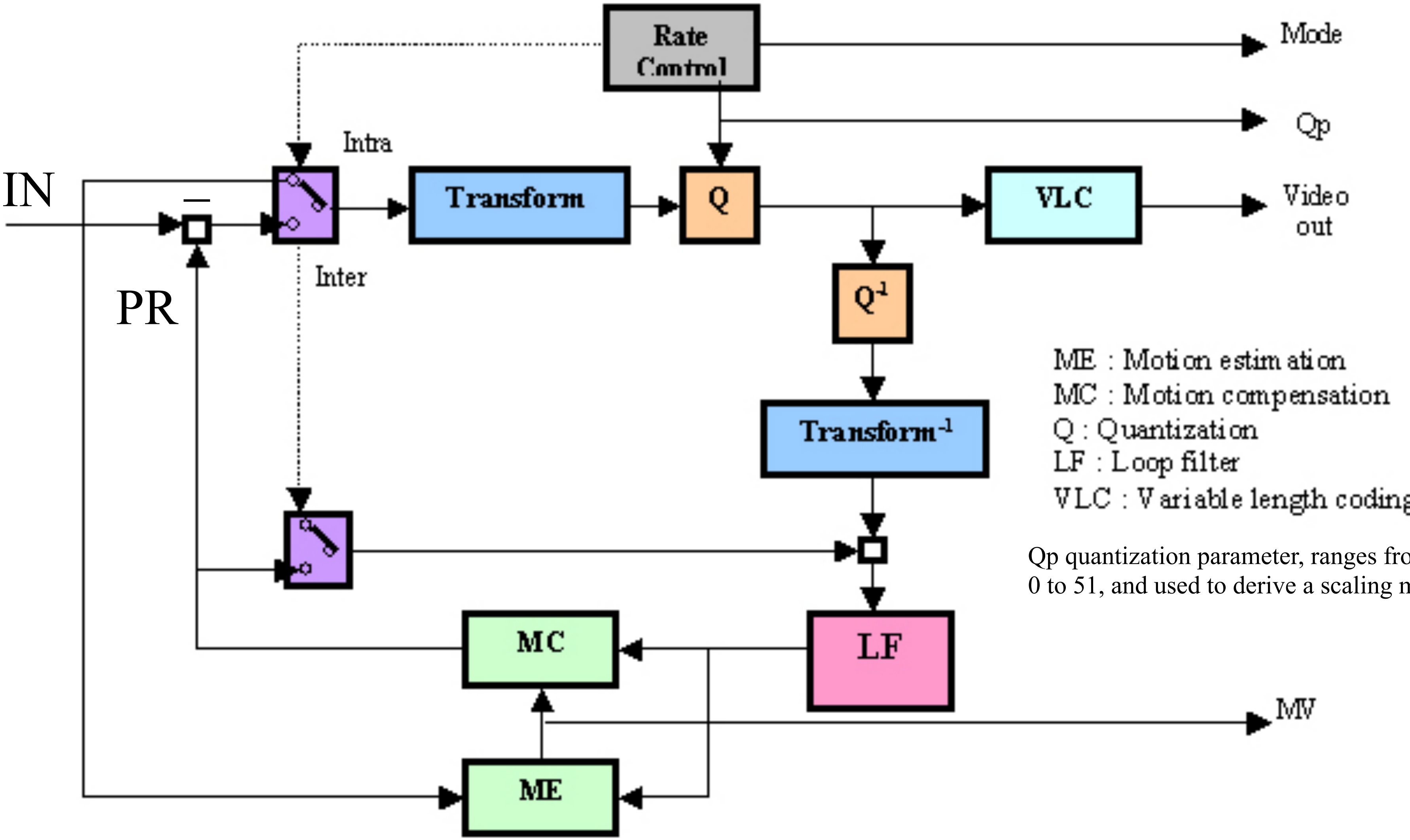
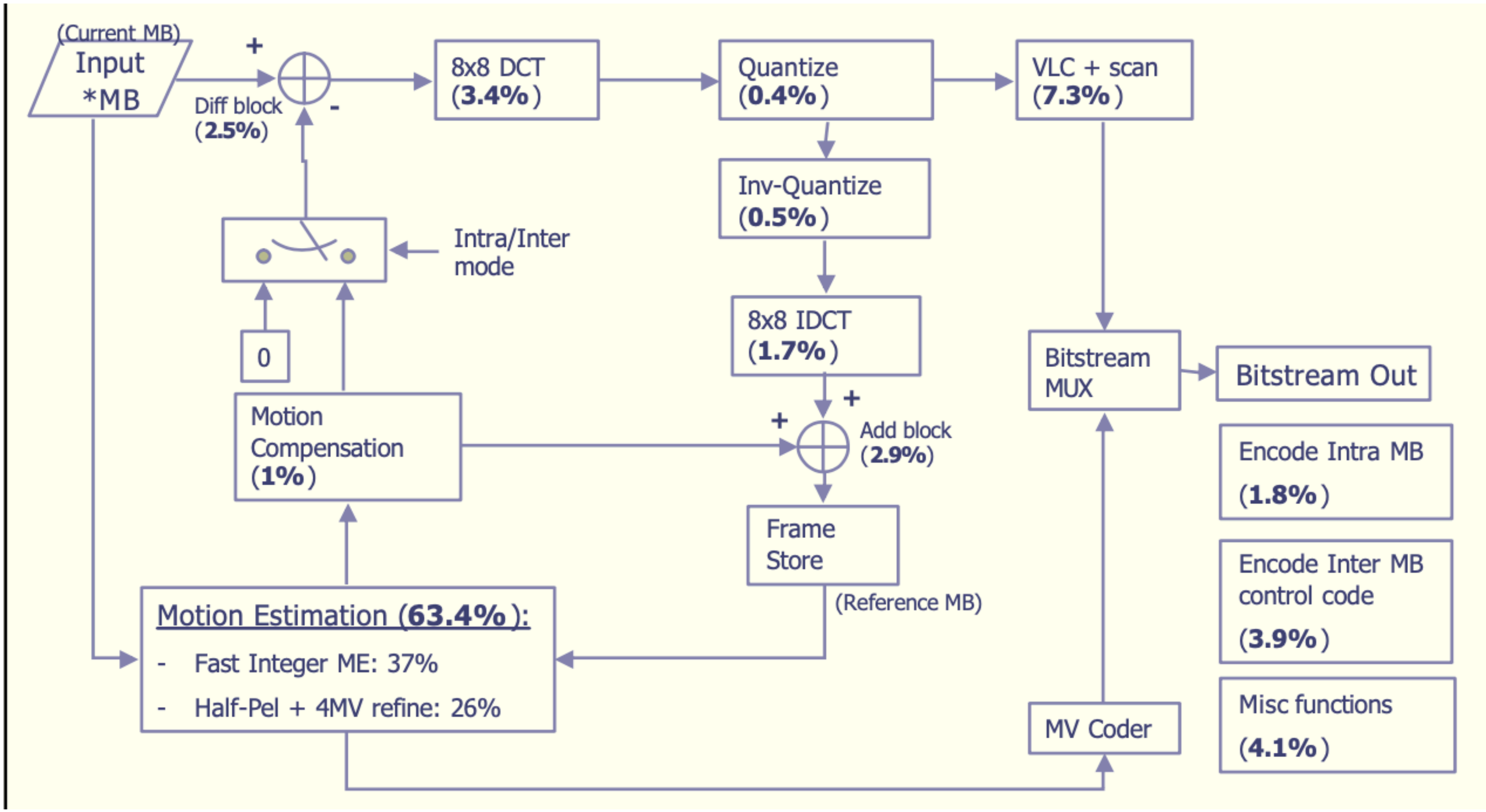


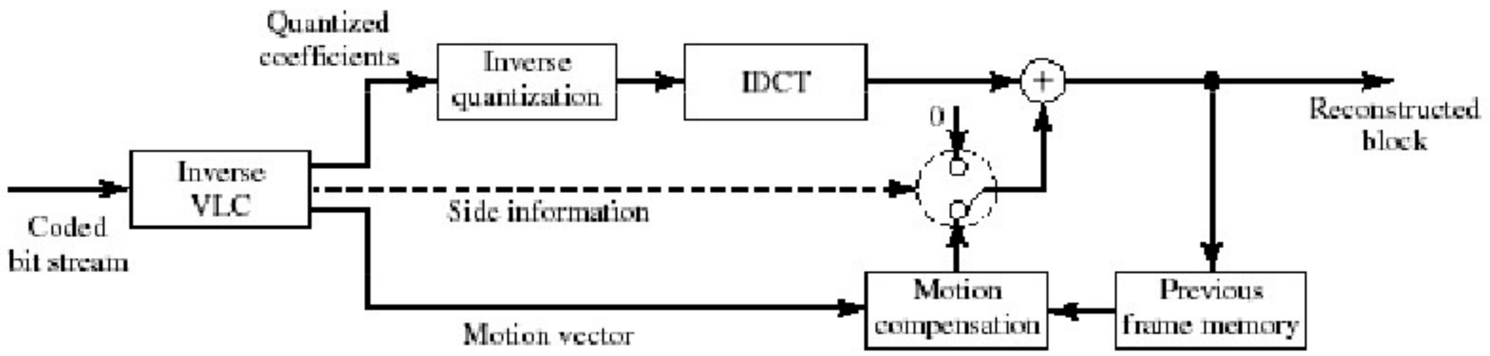
Figure 1: Generic Video encoder's block diagram

Qp quantization parameter, ranges from 0 to 51, and used to derive a scaling matrix

Video Encoder Block Diagram



Decoder Block Diagram



Decoder

Redundancies in video compression

Perceptual redundancy

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

Spatial redundancy

- Neighboring pixels have close luminance levels
- Intra-frame compression making use of
  - Correlation/compression within a frame

Temporal redundancy

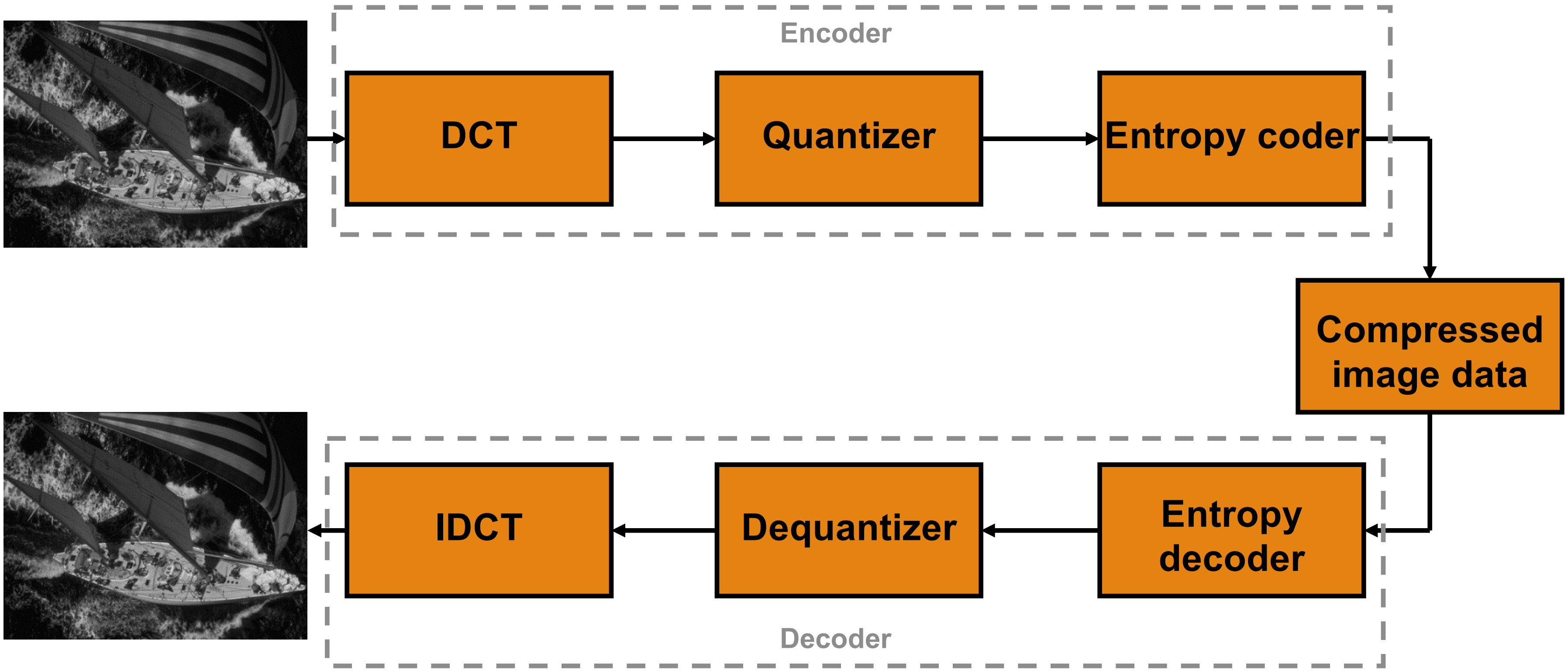
- Differences between adjacent frames can be small
- Inter-frame compression exploiting
  - Correlation/compression between frames
  - Motion estimation and motion compensation

Statistical redundancy

Context-adaptive binary arithmetic coding (**CABAC**) (entropy **encoding)** is applied to the output bitstream

Which of these redundancies can be exploited in Image Compression?

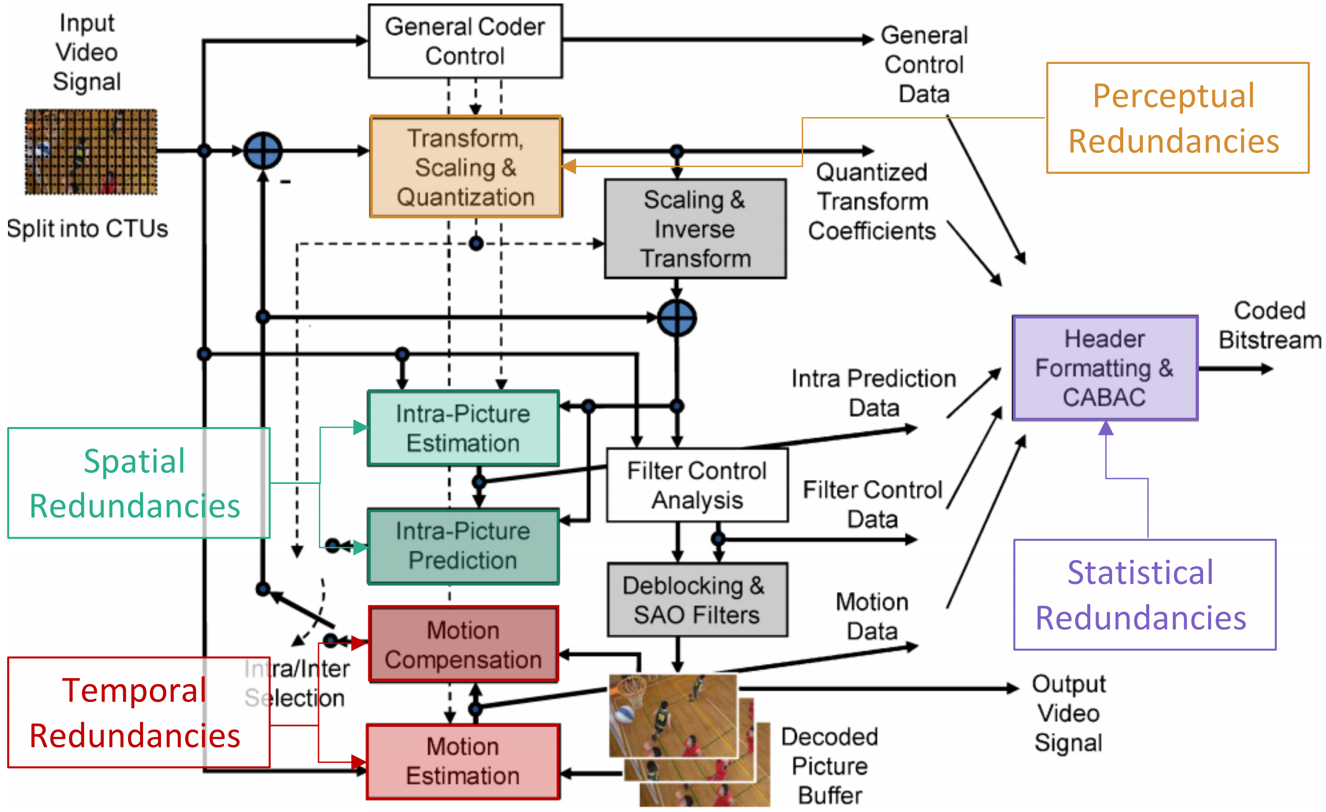
Fundamentals of JPEG



Key ideas in video compression

- Predict a new frame from a previous frame and only code the prediction error
- Prediction error will be coded using an image coding method (e.g., DCT-based as in JPEG)
- 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-based method
- Use motion-compensated prediction to account for object motion
- 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)

Redundancies in Hybrid Coding



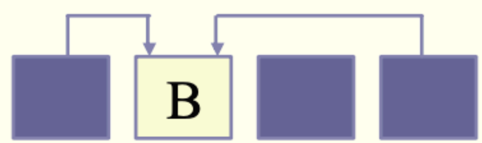
Prediction = Estimation + Compensation

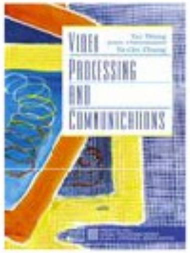
- Requires at least one "reference frame"
  - Reference frame must be encoded before the current frame
  - But, reference frame can be a future frame in the display sequence
- Three kinds of frames: I, P, and B

- I frame is encoded as a still image and doesn't depend on any reference frame

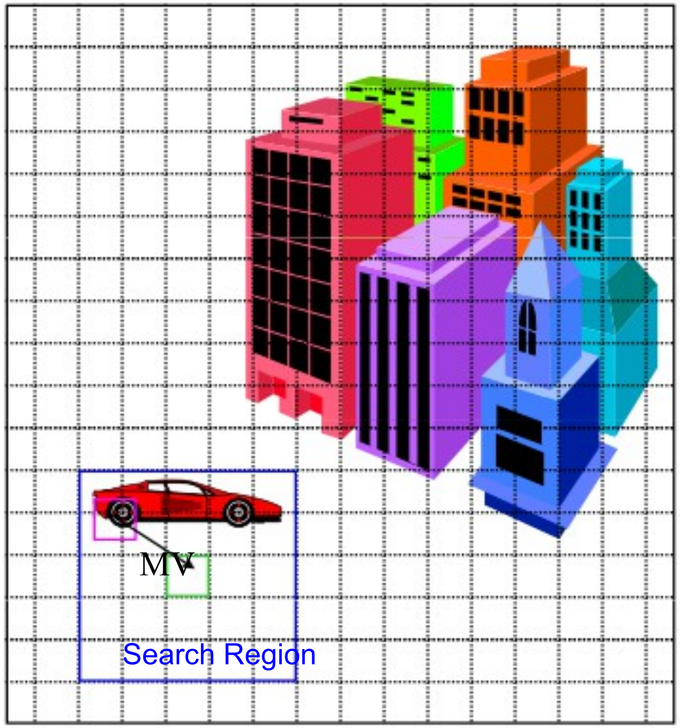
- P frame depends on previously displayed reference frame

- B frame depends on previous and future reference frames

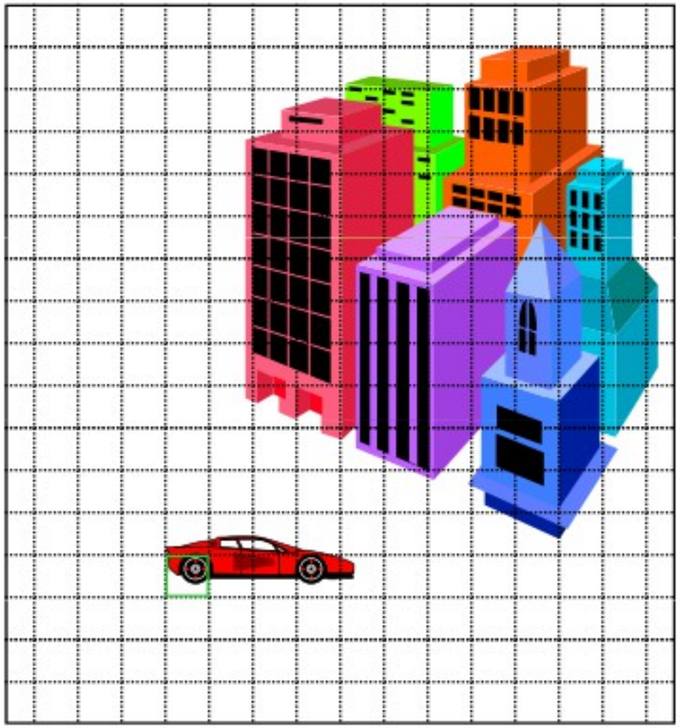




Motion Estimation –
Block Matching Algorithm (BMA)



Frame *t-1*
(Reference Frame)

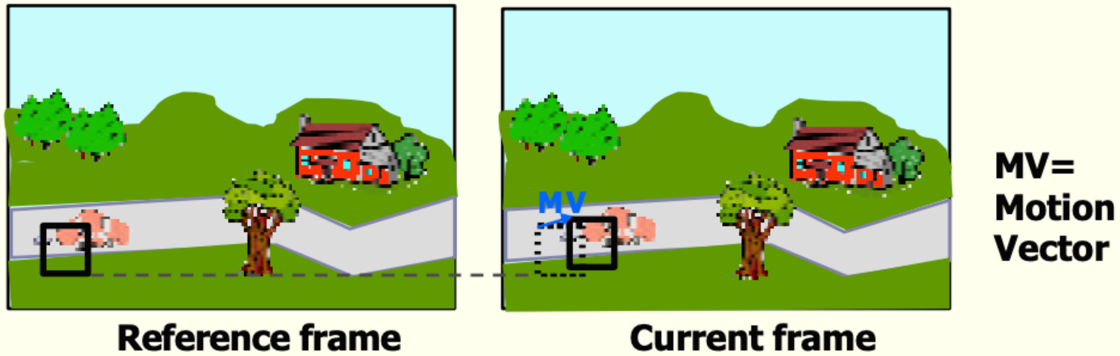


Frame *t*
(Predicted frame)

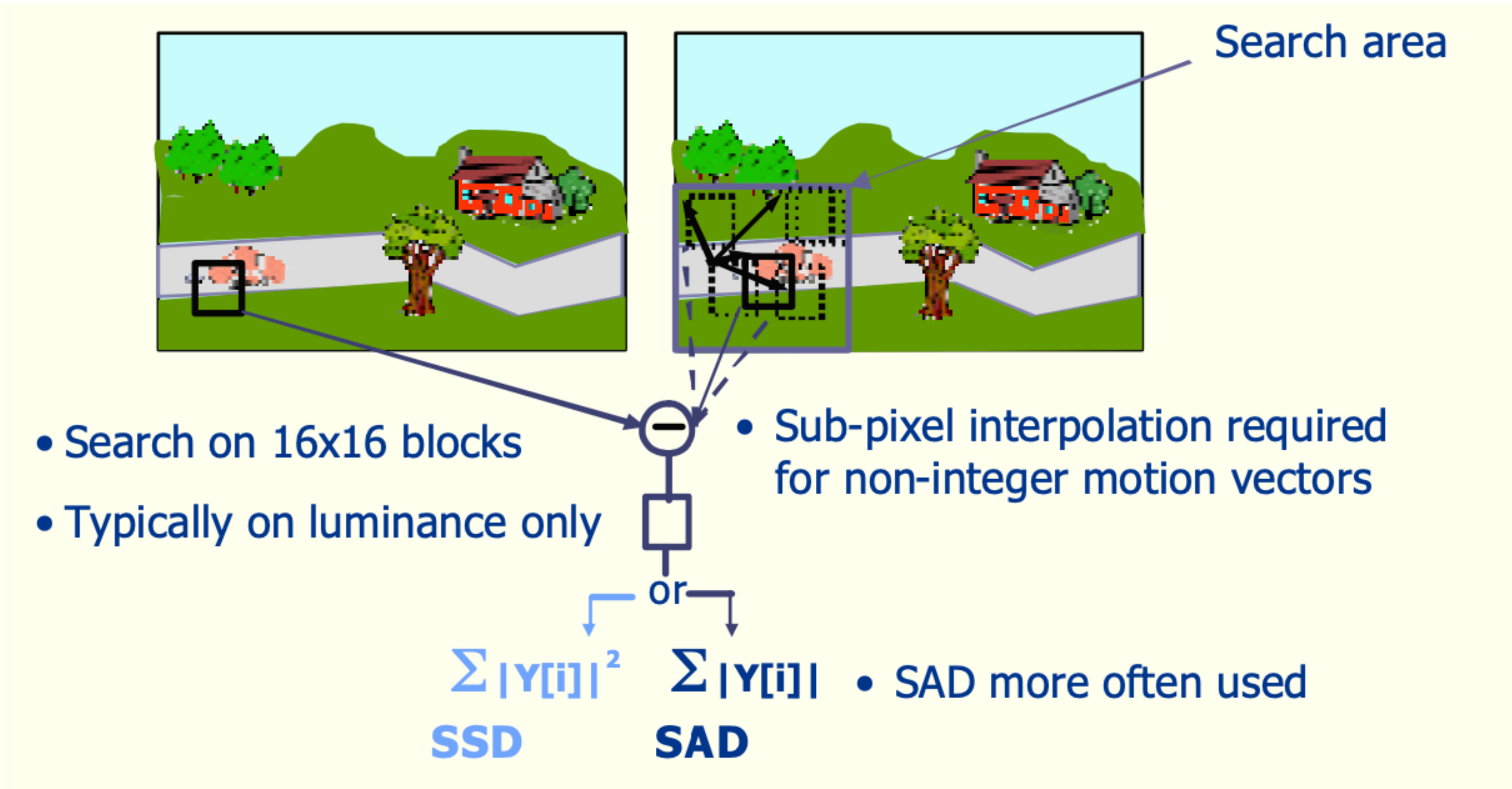
Adjacent frames are similar and changes are due to object or camera motion --- Temporal correlation

Motion estimation

- Predict the contents of each macroblock based on motion relative to reference frame
  - Search reference frame for a 16x16 region that matches the macroblock
  - Encode motion vectors
  - Encode difference between predicted and actual macroblock pixels

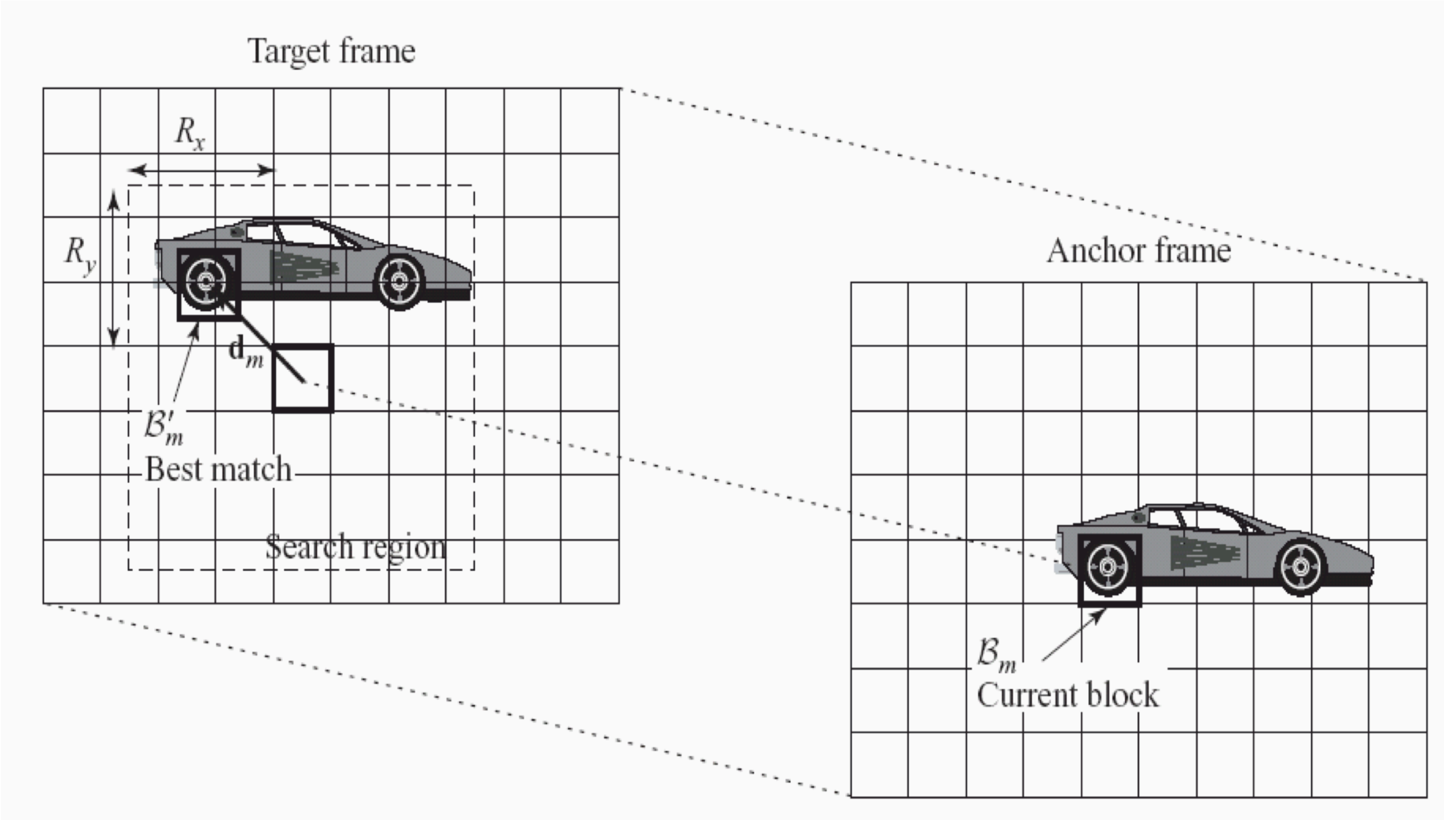


Motion estimation: The problem



SSD = Sum of Squared Differences SAD = Sum of Absolute Differences

Exhaustive Block Matching Algorithm (EBMA)



EBMA searches the full search region to find the best match

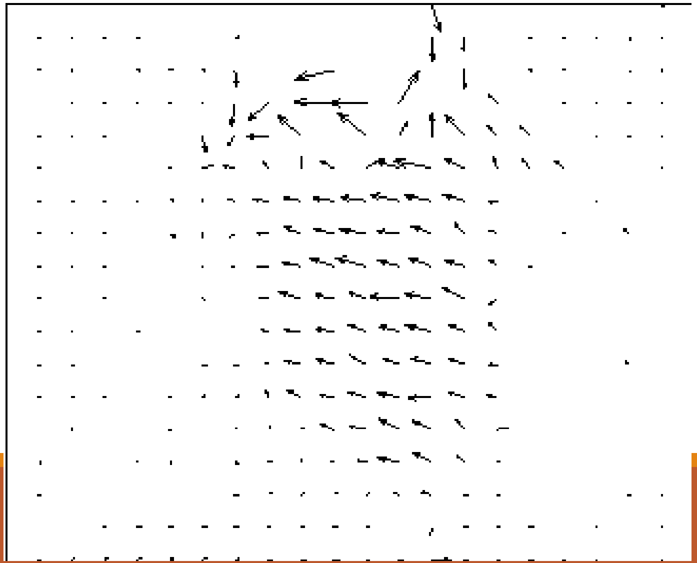
Advantages and disadvantages of Block Matching Algorithms

- 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 one block to another
  - Fix 1: Mesh-based motion estimation
  - Fix 2: Imposing smoothness constraint explicitly
- Wrong MV in flat regions
  - because motion is indeterminate when spatial gradient is near zero
- Nonetheless, it is widely used for motion compensated prediction in video coding
  - Because of its simplicity and optimality in minimizing the prediction error

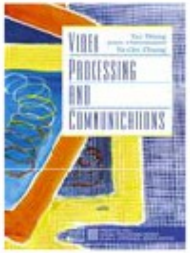
Sample (2D) Motion Field

Anchor Frame

Target Frame

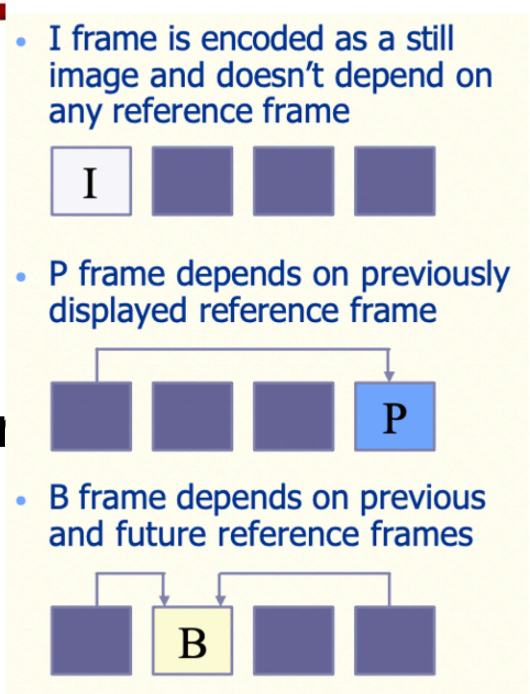


Motion Field



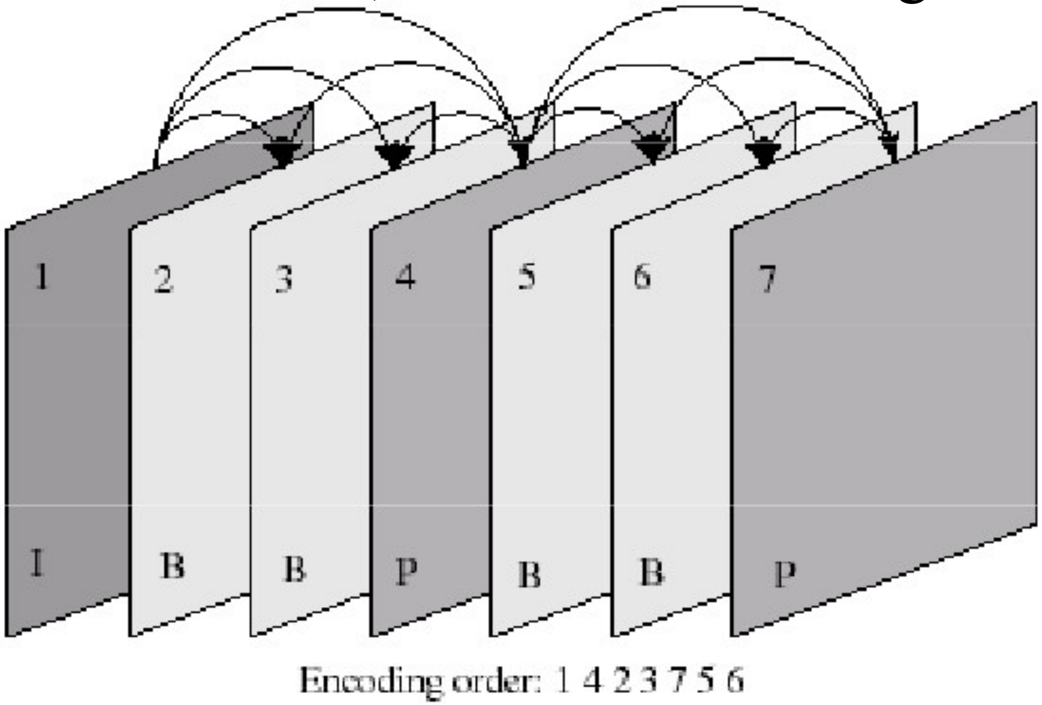
Group-of-Picture Structure (GOP)

- 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 (bi-directional prediction)
  - Require extra delay!
  - Enable frame skip at receiver (temporal scalability)
- *Typically*, an I-frame every 15 frames (0.5 seconds), and
- two or more B-frames before and after each P frame
  - Compromise between compression and delay



Group of Picture Structure

Given a GOP of IBBPBBP, what is the encoding order?



Encoding order:

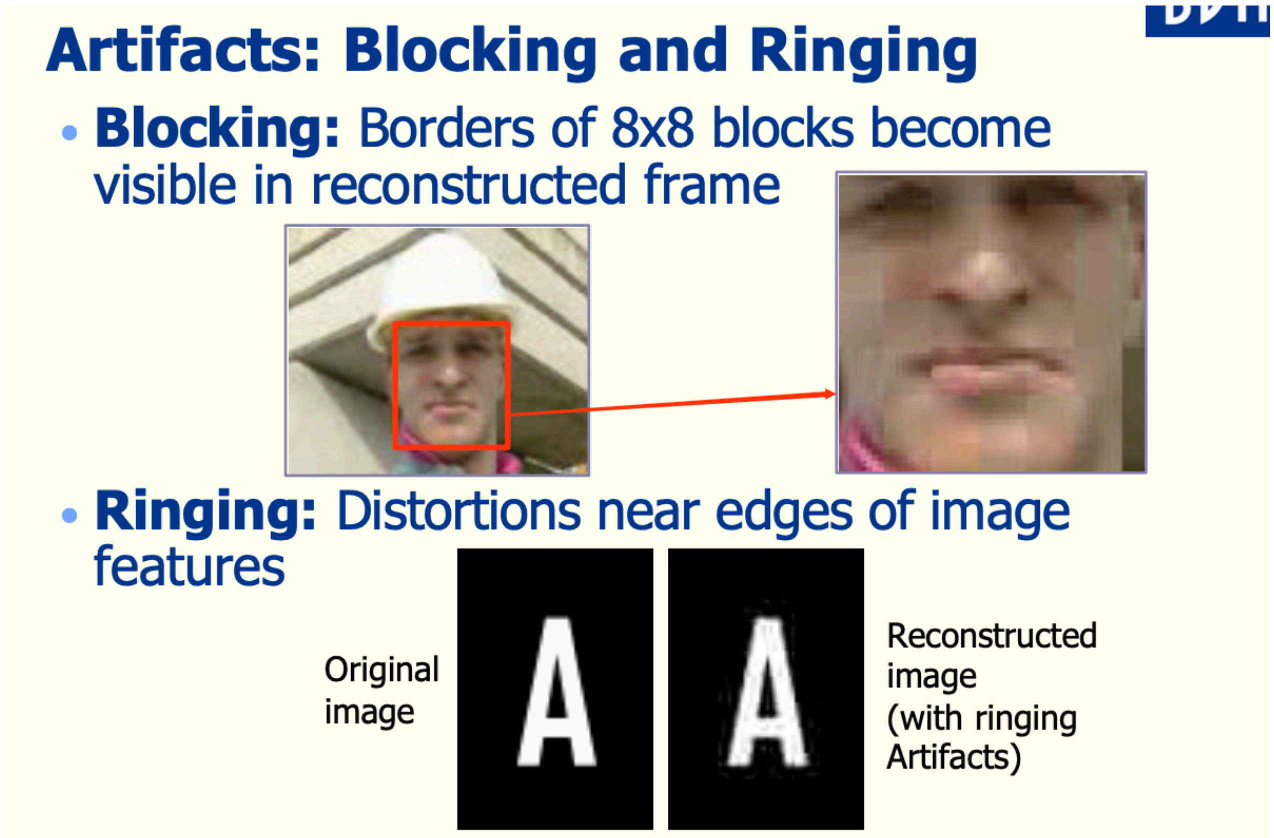
1^{st}, 4^{th} (using 1^{st}), 2^{nd} (using 1^{st}, 4^{th}), 3^{rd} (using 1^{st}, 4^{th}), 7^{th} (using 4^{th}), 5^{th} (using 4^{th}, 7^{th}), 6^{th} (using 4^{th}, 7^{th})

Motion Compensated Prediction

- Divide current frame into disjoint 16×16 macroblocks (MB)

- For each MB, search a window in the reference frame(s) for closest match (e.g. min Sum of Absolute Difference, SAD)
- Motion compensation copies pixels from the reference frame to predict the current macroblock (calculate the prediction error from min SAD)
- For each of the four 8×8 blocks in the macroblock, perform DCT-based coding
- Transmit motion vector (displacement between the current MB and the best matching MB) + entropy coded prediction error (lossy coding)
- Computational load
  - Varies with video content
  - can require 5-40% of the total decoder processor cycles
- Memory usage
  - Require reference frame buffers

Reducing artifacts



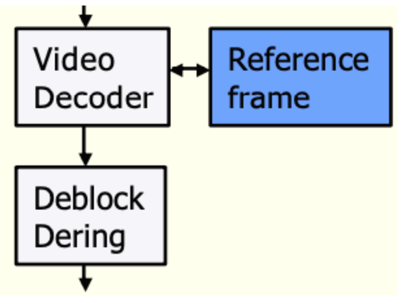
Deblocking and deringing filters

Low-pass filters are used to smooth the image where artifacts occur

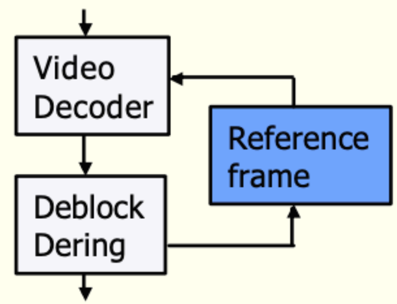
- **Deblocking:**
  - Low-pass filter the pixels at borders of 8x8 blocks
  - One-dimensional filter applied perpendicular to 8x8 block borders
- **Deringing:**
  - Detect edges of image features
  - Adaptively apply 2D filter to smooth out areas near edges
  - Little or no filtering applied to edge pixels in order to avoid blurring

Artifact reduction: Post-processing versus In-loop filtering

- Deblocking/deringing often applied after the decoder (post-processing)
  - Reference frames are not filtered
  - Developers free to select best filters for the application or not filter at all



- Deblocking/deringing can be incorporated in the compression algorithm (in-loop filtering)
  - Reference frames are filtered
  - Same filters must be applied in encoder and decoder
  - Better image quality at very low bit-rates



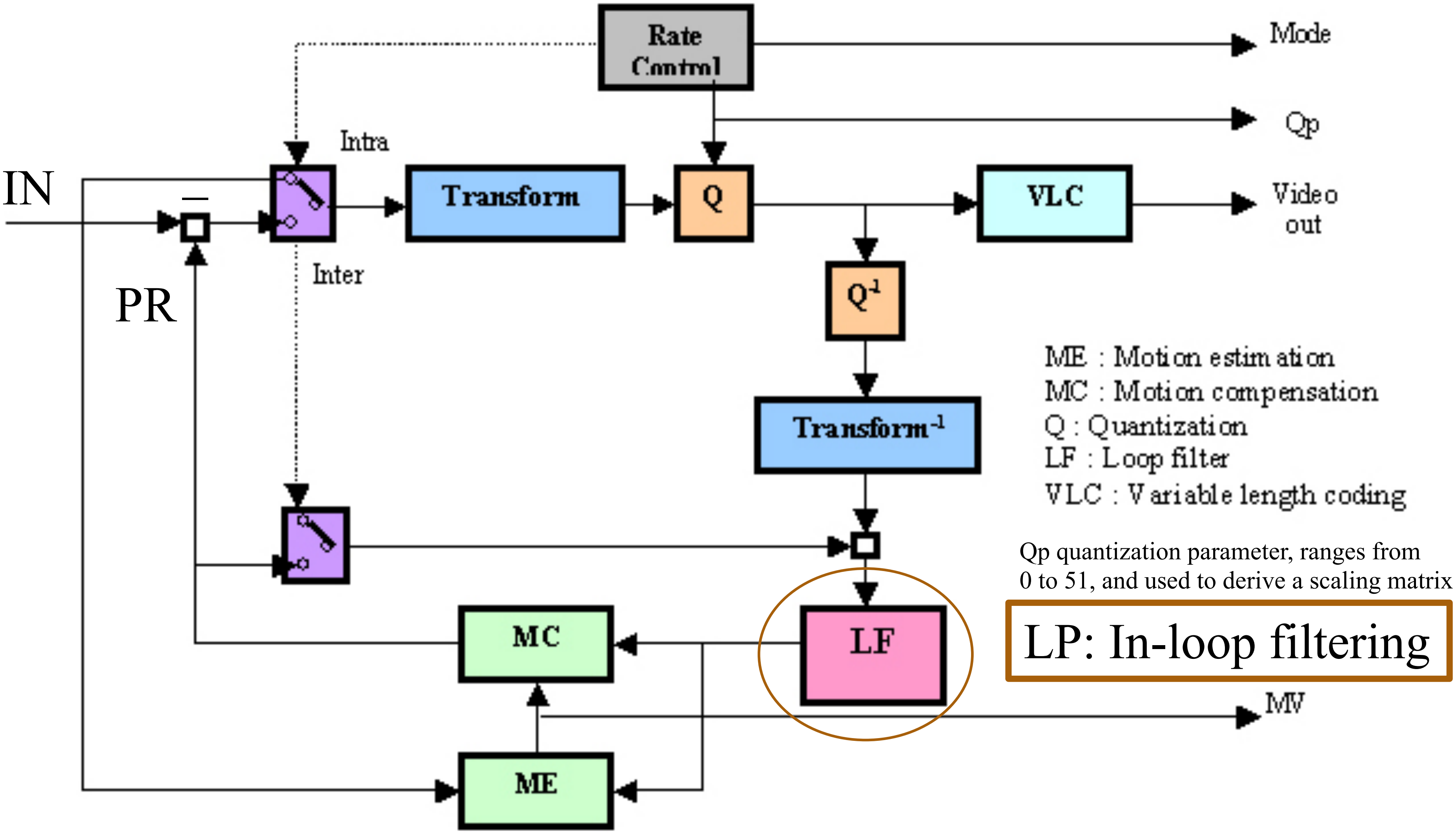


Figure 1: Generic Video encoder's block diagram

Current Image and Video Compression Standards

Standard	Application	Bit Rate
JPEG	Continuous-tone still-image compression	Variable
H.261	Video telephony and teleconferencing over ISDN	p x 64 kb/s
MPEG-1	Video on digital storage media (CD-ROM)	1.5 Mb/s
MPEG-2	Digital Television	2-20 Mb/s
H.263	Video telephony over PSTN	33.6-? kb/s
MPEG-4	Object-based coding, synthetic content, interactivity	Variable
JPEG-2000	Improved still image compression	Variable
H.264 / MPEG-4 AVC	Improved video compression	10's kb/s to Mb/s

MPEG and JPEG: International Standards Organization (ISO)
H.26x family: International Telecommunications Union (ITU)