

## Review of SGN-3016 Digital Image Processing Course

## 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 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.



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.

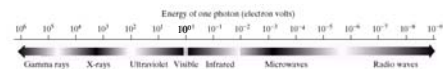
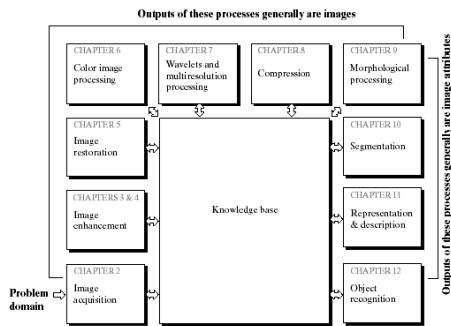


FIGURE 1.5 The electromagnetic spectrum arranged according to energy per photon.

## 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

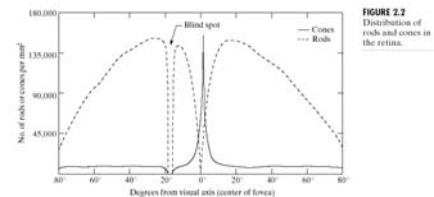


FIGURE 2.2 Distribution of rods and cones in the retina.

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.

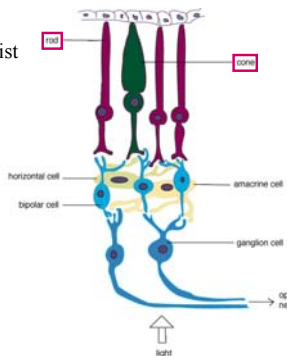
**Cones** are responsible for **photopic** (color or bright-light) vision; while **rods** are for **scotopic** (dim-light) vision.

Retina is circular with 1.5 mm in diameter with 150 000 cones/mm<sup>2</sup>, 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*.

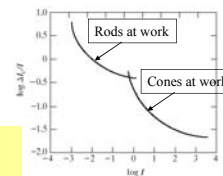
*Rods are long slender receptors, 75~150 million, and cones are shorter and thicker, 6~7 million.*



### 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

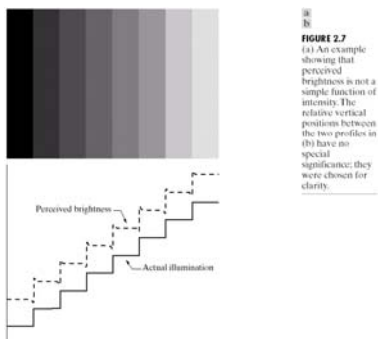
Perceived brightness is NOT a simple function of intensity.

#### 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.



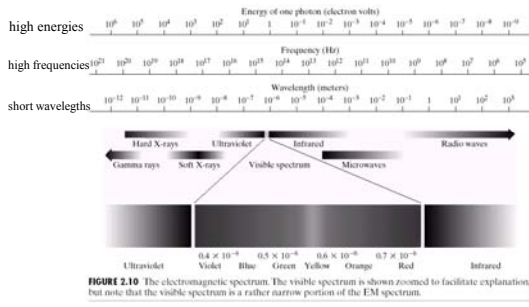
### Chapter 2: Digital Image Fundamentals

#### 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

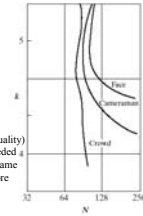


Chapter 2: Digital Image Fundamentals

**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:**

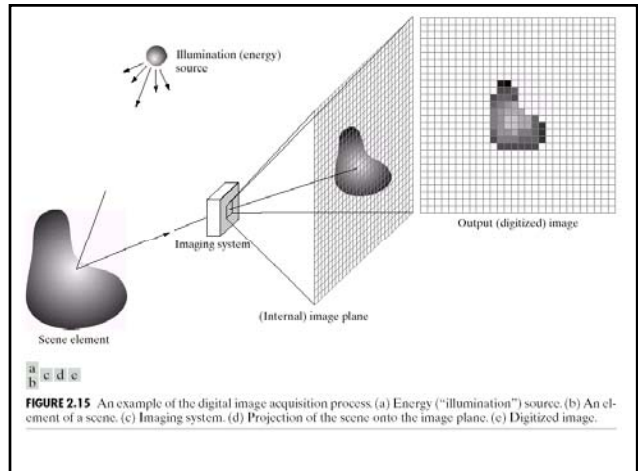
1. Isopreference curves tend to shift right and upward (i.e. better image quality)
2. In images with a large amount of details, only a few gray levels are needed
3. In the other two image categories, the perceived quality remained the same in some intervals in which  $N$  was increased but  $k$  actually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

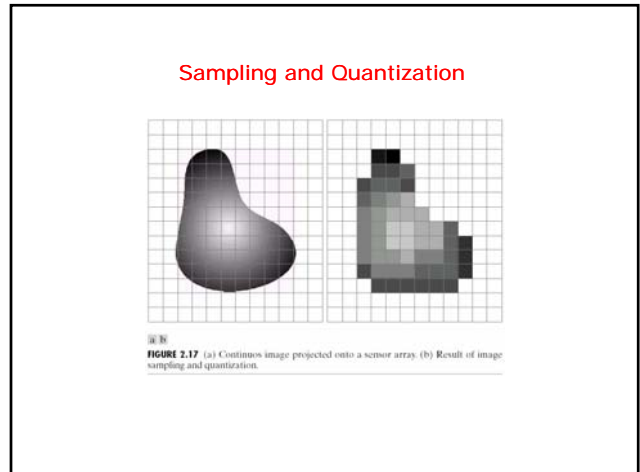
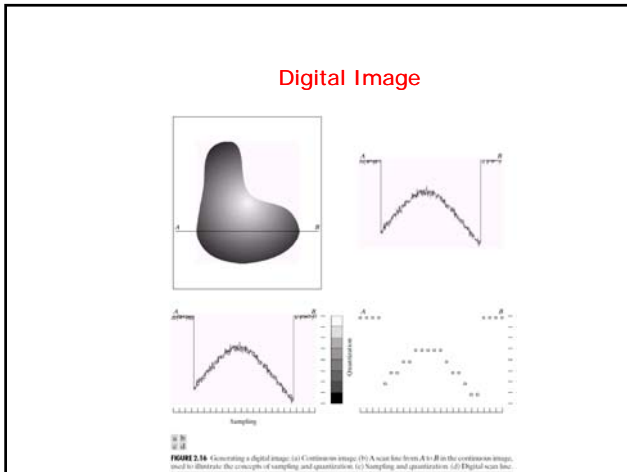
A Simple Image Formation Model

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,  
 (black)  $0 \leq f(x,y) \leq f_{max}$  (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.





### 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:

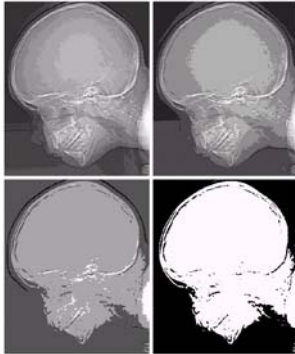
- Contrast Quantization,
- Dithering.

FIGURE 2.21 (a) 452 × 374, 256-level image. (b) 452 × 374, 64-level image. (c) 452 × 374, 32-level image. (d) 452 × 374, 16-level image. Note the appearance of very fine ridge-like structures in the areas of smooth gray levels, e.g. skull.

64 levels

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

FIGURE 2.21  
A commonly  
occurring  
artifact  
displayed in 16, 8,  
4, and 2 gray  
levels. (Original  
courtesy of  
Dr. David  
D. Parker,  
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 logical 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

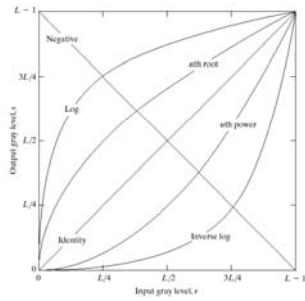
- **Goal: Image enhancement seeks**
  - to improve the visual appearance of an image, or
  - convert it to a form suited for analysis by a human or a machine.
- **Image enhancement does not, however,**
  - seek to restore the image, nor
  - increase its information contents
- **Peculiarity:**
  - actually, 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

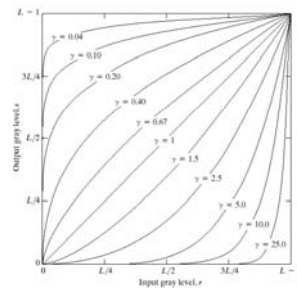
FIGURE 3.3 Some basic gray-level transformation functions used for image enhancement.



$$s = T[r]$$

### Image Enhancement in the Spatial Domain Power-Law transformations

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



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

### Image Enhancement in the Spatial Domain Piecewise-Linear Transformations

#### 2. Level slicing

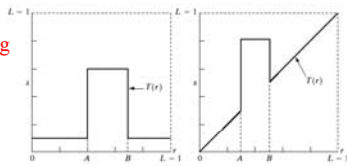
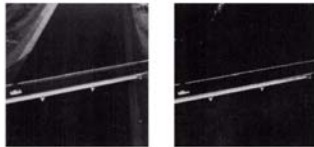


FIGURE 3.11 (a) This transformation highlights range [A, B] of gray levels and reduces all others to a constant level. (b) This transformation highlights range [A, B] but preserves all other levels. (c) An image. (d) Result of using the transformation in (a).

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

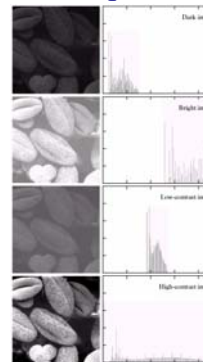
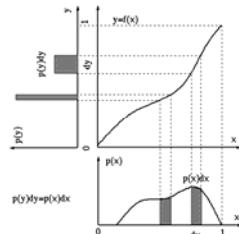


FIGURE 3.15 Four basic image types: dark, light, low contrast, high contrast, and their corresponding histograms. (Digital Image courtesy of Dr. Roger Handberg, School of Biological Sciences, Australian National University, Canberra, Australia.)

### 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:
 
$$h[i] = \text{constant, for all } i$$
- 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. In particular, if the gray levels are assumed to be in the ranges between 0 and 1 ( $0 < x < 1, 0 < y < 1$ ), then we have:

$$dy = p(x)dx, \text{ or } \frac{dy}{dx} = p(x)$$

- i.e., the mapping function for histogram equalization is:

$$y = \int_0^x p(u)du = F(x) - F(0) = F(x)$$

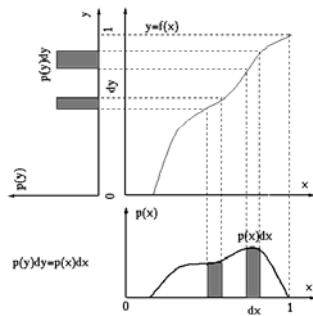
- where

$$F(x) = \int_0^x p(u)du, \quad F(0) = 0$$

- is the cumulative probability distribution of the input image, which monotonically increases.

### 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, will be wide, causing to be low to keep;
  - If  $p(x)$  is low,  $y=f(x)$  has a shallow slope, will be narrow, causing to be high.



### Image Enhancement in the Spatial Domain Histogram Equalization

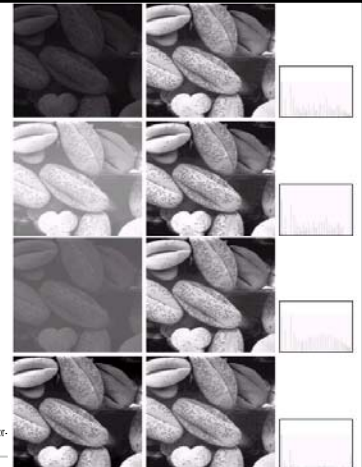
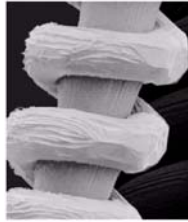


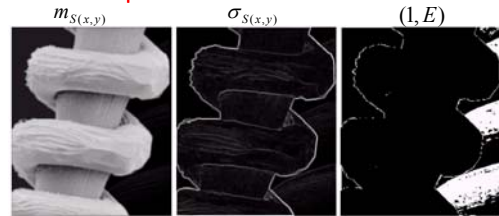
FIGURE 3.17 (a) Images from Fig. 3.15. (b) Results of histogram equalization. (c) Corresponding histograms.

### Image Enhancement in the Spatial Domain

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



### Image Enhancement in the Spatial Domain



**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.

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

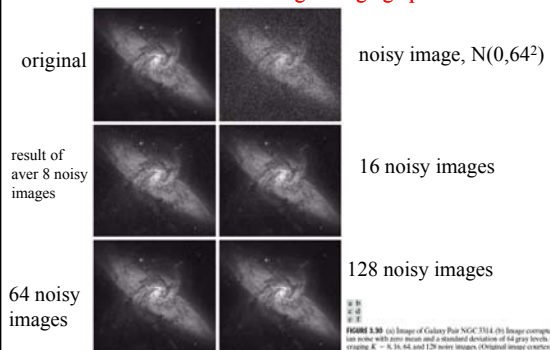
### Image Enhancement in the Spatial Domain



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

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

### Image Enhancement in the Spatial Domain Enhancement using averaging operations



**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.)

sk1

### 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

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

### Image Enhancement in the Spatial Domain

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  $3 \times 3$  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

#### Spatial 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  $K=(2M+1) \times (2N+1)$ :

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

produces an output image with

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

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

## Slide 37

---

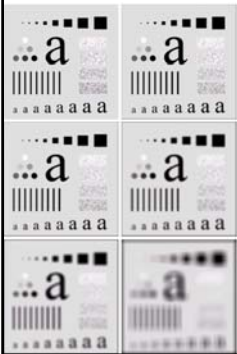
sk1 How to create (Gaussian, Uniform, ..) noise?

How to add them (scaling)?

Why adding diminishes the noise variance? (prove)

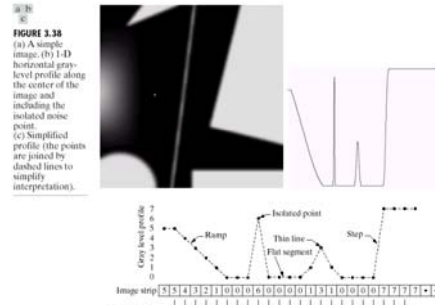
serkan; 9/16/2007

**Image Enhancement in the Spatial Domain**



**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 Spatial Filters**



**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 Spatial 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:**

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

- **this can be implemented using the mask in the next slide**

**Image Enhancement in the Spatial Domain**  
**Combining spatial enhancement methods**

**original** →

**Laplacian of original**

- 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 diseases, e.g. bone infection and tumors

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

**original+Laplacian** →

**Sobel of orig**

**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).

## 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
- **Transforms**

## Chapter 4 Image Enhancement in the Frequency Domain



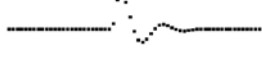
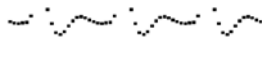
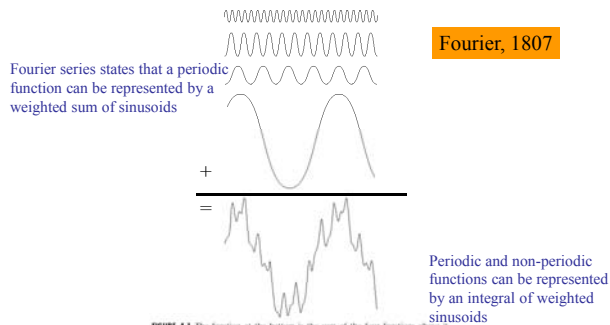
Type of Transform	Example Signal
Fourier Transform <i>signals that are continuous and aperiodic</i>	
Fourier Series <i>signals that are continuous and periodic</i>	
Discrete Time Fourier Transform <i>signals that are discrete and aperiodic</i>	
Discrete Fourier Transform <i>signals that are discrete and periodic</i>	

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



## 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

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

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

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

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

#### The Discrete-Time Fourier Transform

The F.T. can be also written as

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

Note that

$$X_s(\omega + \omega_s) = X_s(\omega)$$

By defining

$$\Omega = \omega T = 2\pi fT = 2\pi \frac{f}{f_s}$$

and omitting the symbol T and the subscript s one can write

$$X(e^{j\Omega}) = \sum_{n=-\infty}^{\infty} x(n) e^{-jn\Omega}$$

which is known as the Discrete-time Fourier Transform of  $x(n)$ .

### Chapter 4 Image Enhancement in the Frequency Domain

#### The Inverse Discrete-time Fourier Transform

The Inverse DTFT is

$$x(n) = \frac{1}{2\pi} \int_{-\pi}^{\pi} X(e^{j\Omega}) e^{jn\Omega} d\Omega$$

A DTFT pair is denoted as

$$x(n) \leftrightarrow X(e^{j\Omega})$$

Unlike the CFT the DTFT is a periodic complex function with period  $2\pi$ . The DTFT is a linear transformation and has properties similar to those of F.T.

### Chapter 4 Image Enhancement in the Frequency Domain

#### Numerical Computation of the Fourier Transform The DFT and the FFT

For numerical computation not only time has to be discrete, but also frequency. Discretizing yields a frequency spacing

Sample  $\Omega$  at regular intervals  $\Omega \Rightarrow \Omega_k = \frac{2\pi}{N} k$

and the discrete spectrum is given by

$$X(e^{j\Omega_k}) = \sum_{n=0}^{N-1} x(n) e^{-jn\Omega_k}$$

### Numerical Computation of the Fourier Transform The DFT and the FFT

For numerical computation not only time has to be discrete, but also frequency. Discretizing yields a frequency spacing

$$\text{Sample } \Omega \text{ at regular intervals} \quad \Omega \Rightarrow \Omega_k = \frac{2\pi}{N} k$$

and the discrete spectrum is given by

$$X(e^{j\Omega_k}) = \sum_{n=0}^{N-1} x(n) e^{-jn\Omega_k}$$

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N} \quad \text{and } k=0, 1, \dots, N-1$$

The inverse Discrete Fourier Transform (IDFT) of the sequence  $x(n)$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j2\pi kn/N} \quad \text{and } n=0, 1, \dots, N-1$$

### 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  $f_s/N$  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  $k f_s/N$ . 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 used to provide more resolution in the frequency components
- Zero padding is often combined with tapered windows

### Chapter 4 Image Enhancement in the Frequency Domain

#### Spectral Estimates over Finite-time Data windows

Frequency domain representations are appropriately defined by the Fourier Transform integrals over an infinite time span.

The DFT, however, estimates the spectrum over finite time

The DFT essentially applies a window to truncate the data.

The simplest data window is the rectangular (boxcar).

Truncation in time is convolution in frequency

The frequency domain characteristics of the data window, namely its bandwidth and sidelobes, affect the DFT spectral estimate.

### Chapter 4 Image Enhancement in the Frequency Domain

The Fourier transform of a discrete function of one variable,  $f(x)$ ,  $x = 0, 1, 2, \dots, M - 1$ , is given by the equation

$$F(u) = \frac{1}{M} \sum_{x=0}^{M-1} f(x) e^{-j2\pi ux/M} \quad \text{for } u = 0, 1, 2, \dots, M - 1. \quad (4.2-5)$$

This *discrete Fourier transform* (DFT) is the foundation for most of the work in this chapter. Similarly, given  $F(u)$ , we can obtain the original function back using the inverse DFT:

$$f(x) = \sum_{u=0}^{M-1} F(u) e^{j2\pi ux/M} \quad \text{for } x = 0, 1, 2, \dots, M - 1. \quad (4.2-6)$$

The  $1/M$  multiplier in front of the Fourier transform sometimes is placed in front of the inverse instead. Other times (not as often) both equations are multiplied by  $1/\sqrt{M}$ . The location of the multiplier does not matter. If two multipliers are used, the only requirement is that their product be equal to  $1/M$ .

### Chapter 4 Image Enhancement in the Frequency Domain

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  $v$ :

$$F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(ux+vy)} dx dy \quad (4.2-3)$$

and, similarly for the inverse transform,

$$f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{j2\pi(ux+vy)} du dv. \quad (4.2-4)$$

### 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

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

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

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

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

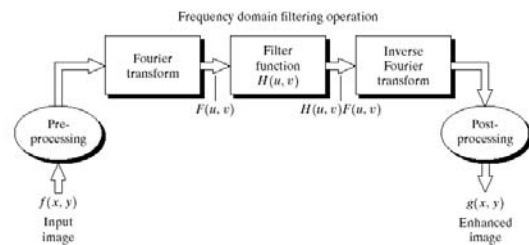


FIGURE 4.5 Basic steps for filtering in the frequency domain.

### Chapter 4 Image Enhancement in the Frequency Domain

3. Gaussian Filters

frequency domain

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

spatial domain

Low-pass      high-pass

**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 marks shown are used in Chapter 3 for lowpass and highpass filtering.

### Chapter 4 Image Enhancement in the Frequency Domain

4. Ideal low-pass filter

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

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

**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

a graylevel 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 responsible for ringing

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

notice blurring and ringing!

**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.

### Chapter 4 Image Enhancement in the Frequency Domain

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

**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  $1/2$  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

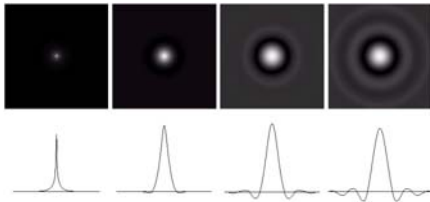


FIGURE 4.16 (a)–(d) Spatial representation of GLPFs of order 1, 2, 5, and 20, and corresponding gray-level profiles through the center of the filters (all filters have a cutoff frequency of 5). Note that ringing increases as a function of filter order.

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

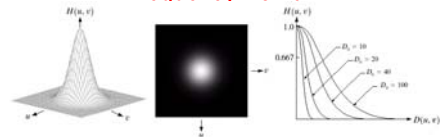


FIGURE 4.17 (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of  $D_0$ .

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

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

$\sigma$  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

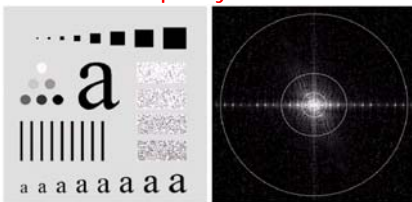


FIGURE 4.11 (a) An image of size 500 × 500 pixels and (b) its Fourier spectrum. The superimposed circles have radii values of 5, 15, 30, 80, and 230, which enclose 92.0, 94.6, 96.4, 98.0, and 99.5% of the image power, respectively.

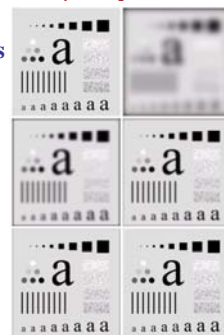
- note the concentration of image energy inside the inner circle.
- what happens if we low-pass filter it with cut-off freq. at the position of these circles? (see next slide)

Chapter 4  
Image Enhancement in the  
Frequency Domain

Results of GLPFs

Remarks:

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

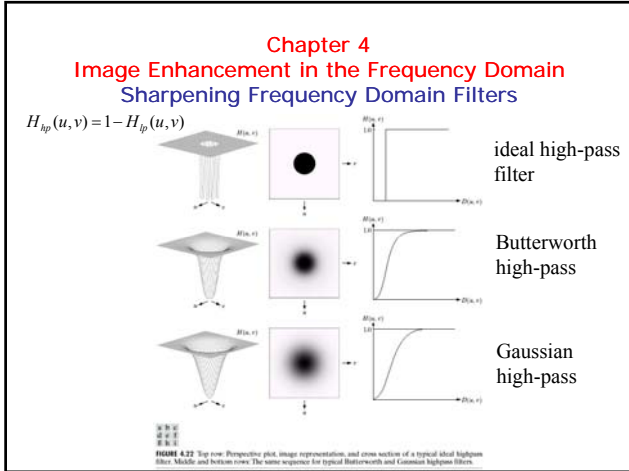


2. Less smoothing than BLPFs since the latter have tighter control over the transitions bet low and high frequencies.

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

3. No ringing!

FIGURE 4.18 (a) Original image. (b)–(f) Results of filtering with Gaussian low-pass filters. The original image is of size 500 × 500. The filters are of radii 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Figs. 4.12 and 4.13.



**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:

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

Therefore, the Laplacian can be implemented in frequency by:

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

Recall that  $F(u,v)$  is centered if  $F(u,v) = FT[(-1)^{x+y} f(x,y)]$   
and thus the center of the filter must be shifted, i.e.

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

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

$$s(x,y) = \mathcal{Z}^{-1}\{S(u,v)\} = \mathcal{Z}^{-1}\{H(u,v)F_i(u,v)\} + \mathcal{Z}^{-1}\{H(u,v)F_r(u,v)\} \quad (4.5-6)$$

By letting

$$\tilde{i}(x,y) = \mathcal{Z}^{-1}\{H(u,v)F_i(u,v)\} \quad (4.5-7)$$

and

$$\tilde{r}(x,y) = \mathcal{Z}^{-1}\{H(u,v)F_r(u,v)\} \quad (4.5-8)$$

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

$$s(x,y) = \tilde{i}(x,y) + \tilde{r}(x,y) \quad (4.5-9)$$

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,

$$g(x,y) = e^{s(x,y)} = e^{\tilde{i}(x,y)} \cdot e^{\tilde{r}(x,y)} = i_0(x,y)r_0(x,y) \quad (4.5-10)$$

where

$$r_0(x,y) = e^{\tilde{i}(x,y)} \quad i_0(x,y) = e^{\tilde{r}(x,y)} \quad (4.5-11)$$

Chapter 4  
Image Enhancement in the  
Frequency Domain: Homomorphic filtering

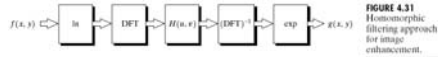


FIGURE 4.31 Homomorphic filtering approach for image enhancement.

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

$$\gamma_L < 1 \text{ and } \gamma_H > 1$$

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

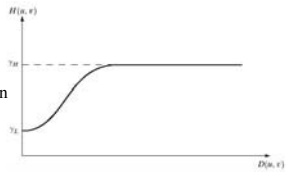


FIGURE 4.32 Cross section of a circularly symmetric filter function,  $D(u, v)$  is the distance from the origin of the centered transform.

Net result: simultaneous dynamic range compression and contrast enhancement

Image Transforms

Unitary Transforms

Recall that

- a matrix  $A$  is **orthogonal** if  $A^{-1} = A^T$
- a matrix is called **unitary** if  $A^{-1} = A^{*T}$

( $A^*$  is the complex conjugate of  $A$ ).

A **unitary** transformation:

$$v = Au, \\ u = A^{-1}v = A^{*T}v$$

is a series representation of  $u$  where  $v$  is the vector of the series coefficients which can be used in various signal/image processing tasks.

4.74

Image Transforms

In image processing, we deal with 2-D transforms.

Consider an  $N \times N$  image  $u(m,n)$ .

An orthonormal (orthogonal and normalized) series expansion for image  $u(m,n)$  is a pair of transforms of the form:

$$v(k,l) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} u(m,n) a_{k,l}(m,n)$$

$$u(m,n) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} v(k,l) a_{k,l}^*(m,n)$$

where the image transform  $\{a_{kl}(m,n)\}$  is a set of complete orthonormal discrete basis functions satisfying the following two properties: 4.75

Image Transforms

Property 1: **orthonormality**:

$$\sum_{m=0}^{N-1} \sum_{n=0}^{N-1} a_{k,l}(m,n) a_{k',l'}^*(m,n) = \delta(k-k', l-l')$$

Property 2: **completeness**:

$$\sum_{k=0}^{N-1} \sum_{l=0}^{N-1} a_{k,l}(m,n) a_{k,l}^*(m',n') = \delta(m-m', n-n')$$

$v(k,l)$ 's are called the *transform coefficients*,  $V = [v(k,l)]$  is the *transformed image*, and  $\{a_{kl}(m,n)\}$  is the *image transform*.

4.76

## Image Transforms

### Remark:

- Property 1 minimizes the sum of square errors for any truncated series expansion.
- Property 2 makes this error vanish in case no truncation is used.

4.77

## Image Transforms

### Separable Transforms

- The computational complexity is reduced if the transform is *separable*, that is,

$$a_{k,l}(m,n) = a_k(m) b_l(n) = a(k,m) b(l,n)$$

where  $\{a_k(m), k=0, \dots, N-1\}$  and  $\{b_l(n), n=0, \dots, N-1\}$  are 1-D complete orthonormal sets of basis vectors.

4.78

## Image Transforms

### Properties of Unitary Transforms

#### 1. Energy conservation:

if  $v=Au$  and  $A$  is unitary, then

$$\|v\|^2 = \|u\|^2$$

Therefore, a unitary transformation is simply a rotation!

4.79

## Image Transforms

### 2. Energy compaction:

*Example:* A zero-mean vector  $u=[u(0), u(1)]$  with covariance matrix:

$$R_u = \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} \quad 0 < \rho < 1$$

is transformed as

$$v = \frac{1}{2} \begin{bmatrix} \sqrt{3} & 1 \\ -1 & \sqrt{3} \end{bmatrix} u$$

4.80

## Image Transforms

The covariance of  $v$  is:

$$R_v = \begin{bmatrix} 1 + \sqrt{3}(\rho/2) & \rho/2 \\ \rho/2 & 1 - \sqrt{3}(\rho/2) \end{bmatrix} u$$

The total average energy in  $u$  is 2 and it is equally distributed:

$$\sigma_u^2(0) = \sigma_u^2(1) = 1$$

whereas in  $v$ :

$$\sigma_v^2(0) = 1 + \sqrt{3}(\rho/2) \quad \text{and} \quad \sigma_v^2(1) = 1 - \sqrt{3}(\rho/2)$$

The sum is still 2 (energy conservation), but if  $\rho=0.95$ , then

$$\sigma_v^2(0) = 1.82 \quad \text{and} \quad \sigma_v^2(1) = 0.18$$

Therefore, 91.1% of the total energy has been packed in  $v(0)$ .

Note also that the correlation in  $v$  has decreased to 0.83!

## Image Transforms

### Conclusions:

- In general, most unitary transforms tend to pack the image energy into few transform coefficients.
- This can be verified by evaluating the following quantities: If  $\mu_u = E[u]$  and  $R_u = \text{cov}[u]$ , then  $\mu_v = E[v] = A\mu_u$  and

$$R_v = E[(v - \mu_v)(v - \mu_v)^*] = AR_u A^*{}^T$$

- Furthermore, if inputs are highly correlated, the transform coefficients are less correlated.

### Remark:

Entropy, which is a measure of average information, is preserved under unitary transformation.

4.82

## Image Transforms:

### 1-D Discrete Fourier Transform (DFT)

Definition: the DFT of a sequence  $\{u(n), n=0, 1, \dots, N-1\}$  is defined as

$$v(k) = \sum_{n=0}^{N-1} u(n) W_N^{kn} \quad k = 0, 1, \dots, N-1$$

where

$$W_N = \exp\left\{\frac{-j2\pi}{N}\right\}$$

The inverse transform is given by:

$$u(n) = \frac{1}{N} \sum_{k=0}^{N-1} v(k) W_N^{-kn} \quad n = 0, 1, \dots, N-1$$

## Image Transforms:

### 1-D Discrete Fourier Transform (DFT)

To make the transform unitary, just scale both  $u$  and  $v$  as

$$v(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} u(n) W_N^{kn} \quad k = 0, 1, \dots, N-1$$

and

$$u(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} v(k) W_N^{-kn} \quad n = 0, 1, \dots, N-1$$

## Image Transforms: 1-D Discrete Fourier Transform (DFT)

### Properties of the DFT

- a) The  $N$ -point DFT can be implemented via FFT in  $O(N \log_2 N)$ .
- b) The DFT of an  $N$ -point sequence has  $N$  degrees of freedom and requires the same storage capacity as the sequence itself (even though the DFT has  $2N$  coefficients, half of them are redundant because of the conjugate symmetry property of the DFT about  $N/2$ ).
- c) Circular convolution can be implemented via DFT; the circular convolution of two sequences is equal to the product of their DFTs ( $O(N \log_2 N)$  compared with  $O(N^2)$ ).
- d) Linear convolution can also be implemented via DFT (by appending zeros to the sequences).

4.85

## Image Transforms: 2-D Discrete Fourier Transform (DFT)

Definition: The 2-D unitary DFT is a separable transform given by

$$v(k, l) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} u(m, n) W_N^{km} W_N^{ln} \quad k, l = 0, 1, \dots, N-1$$

and the inverse transform is given by:

$$u(m, n) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} v(k, l) W_N^{-km} W_N^{-ln} \quad m, n = 0, 1, \dots, N-1$$

Same properties extended to 2-D as in the 1-D case.

4.86

## Chapter 4 Image Enhancement in the Frequency Domain

**TABLE 4.1**  
Summary of some important properties of the 2-D Fourier transforms.

Property	Expression(s)
Fourier transform	$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)}$
Inverse Fourier transform	$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M + vy/N)}$
Polar representation	$F(u, v) =  F(u, v)  e^{j\phi(u, v)}$
Spectrum	$ F(u, v)  = [R^2(u, v) + I^2(u, v)]^{1/2}$ , $R = \text{Real}(F)$ and $I = \text{Imag}(F)$
Phase angle	$\phi(u, v) = \tan^{-1} \left[ \frac{I(u, v)}{R(u, v)} \right]$
Power spectrum	$P(u, v) =  F(u, v) ^2$
Average value	$\bar{f}(x, y) = F(0, 0) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)$
Translation	$f(x, y) e^{j2\pi(u_0 x/M + v_0 y/N)} \Leftrightarrow F(u - u_0, v - v_0)$ $f(x - x_0, y - y_0) \Leftrightarrow F(u, v) e^{-j2\pi(u x_0/M + v y_0/N)}$ When $x_0 = u_0 = M/2$ and $y_0 = v_0 = N/2$ , then $f(x, y) (-1)^{x+y} \Leftrightarrow F(u - M/2, v - N/2)$ $f(x - M/2, y - N/2) \Leftrightarrow F(u, v) (-1)^{x+y}$

4.87

## Chapter 4 Image Enhancement in the Frequency Domain

Conjugate symmetry	$F(u, v) = F^*(-u, -v)$ $ F(u, v)  =  F(-u, -v) $
Differentiation	$\frac{\partial f(x, y)}{\partial x} \Leftrightarrow (ju)F(u, v)$ $(-jv)^2 f(x, y) \Leftrightarrow \frac{\partial^2 F(u, v)}{\partial v^2}$
Laplacian	$\nabla^2 f(x, y) \Leftrightarrow -(u^2 + v^2)F(u, v)$
Distributivity	$\nabla^2 [f_1(x, y) + f_2(x, y)] = \nabla^2 [f_1(x, y)] + \nabla^2 [f_2(x, y)]$ $\nabla^2 [f_1(x, y) \cdot f_2(x, y)] = \nabla^2 [f_1(x, y)] \cdot \nabla^2 [f_2(x, y)]$
Scaling	$af(x, y) \Leftrightarrow aF(u, v)$ , $f(ax, by) \Leftrightarrow \frac{1}{ ab } F(u/a, v/b)$
Rotation	$x = r \cos \theta$ , $y = r \sin \theta$ , $u = u' \cos \theta$ , $v = u' \sin \theta$ $f(x, y) \Leftrightarrow F(u', v')$
Periodicity	$F(u, v) = F(u + M, v) = F(u, v + N) = F(u + M, v + N)$ $f(x, y) = f(x + M, y) = f(x, y + N) = f(x + M, y + N)$
Separability	See Eqs. (4.6-14) and (4.6-15). Separability implies that we can compute the 2-D transform of an image by first computing 1-D transforms along each row of the image, and then computing a 1-D transform along each column of this intermediate result. The reverse, column and then row, yields the same result.

**TABLE 4.1**  
(continued)

4.88

**Chapter 4**  
**Image Enhancement in the**  
**Frequency Domain**

Property	Expression(s)
Computation of the inverse Fourier transform using a forward transform algorithm	$\frac{1}{MN} F^{-1}(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j\pi(u(x-M/2) + v(y-N/2))}$ This equation indicates that inputting the function $F(u, v)$ into an algorithm designed to compute the forward transform (right side of the preceding equation) yields $F^{-1}(x, y)/MN$ . Taking the complex conjugate and multiplying this result by $MN$ gives the desired inverse.
Convolution*	$f(x, y) * h(x, y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) h(x - m, y - n)$
Correlation*	$f(x, y) \cdot h(x, y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f^*(m, n) h(x + m, y + n)$
Convolution theorem*	$f(x, y) * h(x, y) \Leftrightarrow F(u, v) H(u, v);$ $f(x, y) h(x, y) \Leftrightarrow F(u, v) \cdot H(u, v)$
Correlation theorem*	$f(x, y) \cdot h(x, y) \Leftrightarrow F^*(u, v) H(u, v);$ $F^*(x, y) h(x, y) \Leftrightarrow F(u, v) \cdot H(u, v)$

TABLE 4.1  
(continued)

4.89

**Image Transforms:**  
**Discrete Fourier Transform (DFT)**

**Drawbacks of FT**

- Complex number computations are necessary,
- Low convergence rate due mainly to sharp discontinuities between the right and left side and between top and bottom of the image which result in large magnitude, high spatial frequency components.

4.90

**Image Transforms:**  
**Cosine and Sine Transforms**

- Both are unitary transforms that use sinusoidal basis functions as does the FT.
- Cosine and sine transforms are NOT simply the cosine and sine terms in the FT!

**Cosine Transform**

Recall that *if a function is continuous, real and symmetric, then its Fourier series contains only real coefficients, i.e. cosine terms of the series.*

This result can be extended to DFT of an image by forcing symmetry.

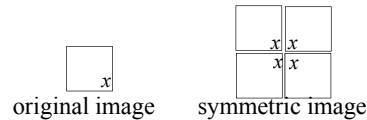
Q: How?

A: Easy!

4.91

**Image Transforms:**  
**Discrete Fourier Transform (DFT)**

Form a symmetrical image by reflection of the original image about its edges, e.g.,



- Because of symmetry, the FT contains only cosine (real) terms:

$$c(n, k) = \begin{cases} \frac{1}{\sqrt{N}} & k = 0, 0 \leq n \leq N-1 \\ \frac{1}{\sqrt{N}} \cos \frac{\pi(2n+1)k}{2N} & 1 \leq k \leq N-1, 0 \leq n \leq N-1 \end{cases}$$

4.92

## Image Transforms

### Remarks

- the cosine transform is real,
- it is a fast transform,
- it is very close to the KL transform,
- it has excellent energy compaction for highly correlated data.

### Sine Transform

- Introduced by Jain as a fast algorithm substitute for the KL transform.

### Properties:

- same as the DCT.

4.93

## Image Transforms

### Hadamard, Haar and Slant Transforms:

all are related members of a family of non-sinusoidal transforms.

### Hadamard Transform

Based on the Hadamard matrix - a square array of  $\pm 1$  whose rows and columns are orthogonal (very suitable for DSP).

Example:

$$H_2 = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Note that

$$H_2 H_2^T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

4.94

## Image Transforms

### How to construct Hadamard matrices?

**A: simple!**

$$H_{2N} = \frac{1}{\sqrt{2}} \begin{bmatrix} H_N & H_N \\ H_N & -H_N \end{bmatrix}$$

Example:

$$H_4 = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}$$

The Hadamard matrix performs the decomposition of a function by a set of rectangular waveforms.

4.95

## Image Transforms

Note: some Hadamard matrices can be obtained by sampling the Walsh functions.

Hadamard Transform Pairs:

$$v(k) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} u(n) (-1)^{b(k,n)} \quad k = 0, 1, \dots, N-1$$

$$u(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} v(k) (-1)^{b(k,n)} \quad n = 0, 1, \dots, N-1$$

where

$$b(k,n) = \sum_{i=0}^{m-1} k_i n_i \quad k_i, n_i = 0, 1$$

and  $\{k_i\}$   $\{n_i\}$  are the binary representations of  $k$  and  $n$ , respectively, i.e.,

$$k = k_0 + 2k_1 + \dots + 2^{m-1} k_{m-1} \quad n = n_0 + 2n_1 + \dots + 2^{m-1} n_{m-1}$$

4.96

## Image Transforms

### Properties of Hadamard Transform:

- it is real, symmetric and orthogonal,
- it is a fast transform, and
- it has good energy compaction for highly correlated images.

4.97

## Image Transforms

### Haar Transform

is also derived from the (Haar) matrix:

ex:

$$H_4 = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ \sqrt{2} & 1 & -\sqrt{2} & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix}$$

It acts like several “edge extractors” since it takes differences along rows and columns of the local pixel averages in the image.

4.98

## Image Transforms

### Properties of the Haar Transform:

- it's real and orthogonal,
- very fast,  $O(N)$  for  $N$ -point sequence!
- it has very poor energy compaction.

4.99

## Image Transforms

### The Slant Transform

is an orthogonal transform designed to possess these properties:

- slant basis functions (monotonically decreasing in constant size steps from maximum to minimum amplitudes),
- fast, and
- to have high energy compaction.
- Slant matrix of order 4:

$$S_2 = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 3a & a & -a & -3a \\ 1 & -1 & -1 & 1 \\ a & -3a & 3a & -a \end{bmatrix} \quad \text{where } a = \frac{1}{\sqrt{5}}$$

4.100

## Image Transforms

### The Karhunen-Loeve Transform (KL)

Originated from the series expansions for random processes developed by Karhunen and Loeve in 1947 and 1949 based on the work of Hotelling in 1933 (the discrete version of the KL transform). Also known as Hotelling transform or method of principal component.

The idea is to transform a signal into a set of uncorrelated coefficients.

#### General form:

$$v(m, n) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} u(k, l) \Psi(k, l; m, n) \quad m, n = 0, 1, \dots, N-1$$

4.101

## Image Transforms

where the kernel

$$\Psi(k, l; m, n)$$

is given by the orthonormalized eigenvectors of the correlation matrix, i.e. it satisfies

$$\lambda_i \Psi_i = R \Psi_i \quad i = 0, \dots, N^2 - 1$$

where  $R$  is the  $(N^2 \times N^2)$  covariance matrix of the image mapped into an  $(N^2 \times 1)$  vector and  $\Psi_i$  is the  $i$ 'th column of  $\Psi$

If  $R$  is separable, i.e.  $R = R_1 \otimes R_2$

Then the KL kernel  $\Psi$  is also separable, i.e.,

$$\Psi(k, l; m, n) = \Psi_1(m, k) \Psi_2(n, l) \quad \text{or} \quad \Psi = \Psi_1 \otimes \Psi_2 \quad 4.102$$

## Image Transforms

### Advantage of separability:

- reduce the computational complexity from  $O(N^6)$  to  $O(N^3)$ !
- Recall that an  $N \times N$  eigenvalue problem requires  $O(N^3)$  computations.

### Properties of the KL Transform

1. Decorrelation: the KL transform coefficients are uncorrelated and have zero mean, i.e.,  
 $E[v(k, l)] = 0$  for all  $k, l$ ; and  $E[v(k, l)v^*(m, n)] = \lambda(k, l)\delta(k - m, l - n)$
2. It minimizes the mse for any truncated series expansion. Error vanishes in case there is no truncation.
3. Among all unitary transformations, KL packs the maximum average energy in the first few samples of  $v$ .

4.103

## Image Transforms

### Drawbacks of KL:

- a) unlike other transforms, the KL is image-dependent, in fact, it depends on the second order moments of the data,
- b) it is very computationally intensive.

4.104

## Image Transforms

### Singular Value Decomposition (SVD):

SVD does for one image exactly what KL does for a set of images.

Consider an  $N \times N$  image  $U$ . Let the image be real and  $M \leq N$ .

The matrix  $UU^T$  and  $U^T U$  are nonnegative, symmetric and have identical eigenvalues  $\{\lambda_i\}$ . There are at most  $r \leq M$  nonzero eigenvalues.

It is possible to find  $r$  orthogonal  $M \times 1$  eigenvectors  $\{\Phi_m\}$  of  $U^T U$  and  $r$  orthogonal  $N \times 1$  eigenvectors  $\{\Psi_m\}$  of  $UU^T$ , i.e.

$$U^T U \Phi_m = \lambda_m \Phi_m, \quad m = 1, \dots, r$$

and

$$UU^T \Psi_m = \lambda_m \Psi_m, \quad m = 1, \dots, r$$

4.105

## Image Transforms: SVD Cont'd

The matrix  $U$  has the representation:

$$U = \Psi \Lambda^{1/2} \Phi^T = \sum_{m=1}^r \sqrt{\lambda_m} \Psi_m \Phi_m^T$$

where  $\Psi$  and  $\Phi$  are  $N \times r$  and  $M \times r$  matrices whose  $m$ th columns are the vectors  $\Psi_m$  and  $\Phi_m$ , respectively.

This is the singular value decomposition (SVD) of image  $U$ , i.e.

$$U = \sum_{l=1}^{MN} v_l a_l b_l^T$$

where  $v_l$  are the transform coefficients.

4.106

## Image Transforms: SVD Cont'd

The energy concentrated in the transform coefficients  $v_1, \dots, v_k$  is maximized by the SVD transformation for the **given image**.

While the KL transformation maximizes the average energy in a given number of transform coefficients  $v_1, \dots, v_k$ , where the average is taken over an **ensemble of images** for which the autocorrelation function is constant.

The usefulness of SVD is severely limited due to the large computational effort required to compute the eigenvalues and eigenvectors of large image matrices.

### Sub-Conclusions:

1. KL is computed for a set of images, while SVD is for a single image
2. There may be fast transformation approximating KLT but not for SVD
3. SVD is more useful elsewhere, e.g. to find generalized inverses for singular matrices
4. SVD could also be useful in data compression.

4.107

## Image Transforms

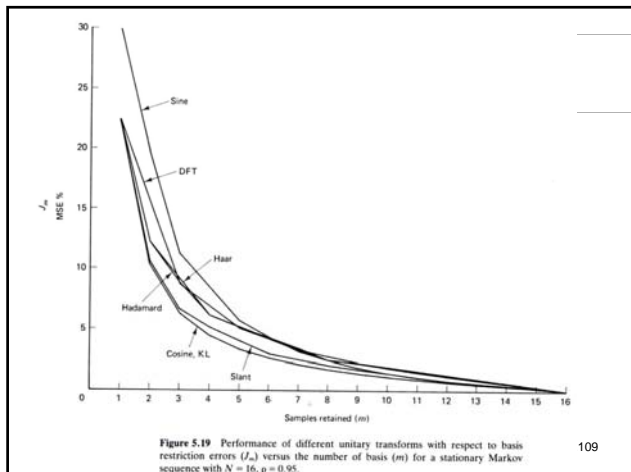
### Evaluation and Comparison of Different Transforms

Performance of different unitary transforms with respect to basis restriction errors ( $J_m$ ) versus the number of basis ( $m$ ) for a stationary Markov sequence with  $N=16$  and correlation coefficient 0.95.

$$J_m = \frac{\sum_{k=m}^{N-1} \sigma_k^2}{\sum_{k=0}^{N-1} \sigma_k^2}, \quad m = 0, \dots, N-1$$

where the variances have been arranged in decreasing order.

4.108



## Image Transforms

### Evaluation and Comparison of Different Transforms

#### Zonal Filtering

Zonal Mask:

Define the normalized MSE:

$$J_s = \frac{\sum_{k,j \in \text{stopband}} |v_{k,j}|^2}{\sum_{k,j=0}^{N-1} |v_{k,j}|^2} = \frac{\text{energy in stopband}}{\text{total energy}}$$

Figure 5.20 Zonal filters for 2:1, 4:1, 8:1, 16:1 sample reduction. White areas are passbands, dark areas are stopbands. .110

#### Zonal Filtering with DCT transform

(a) Original image; (b) 4:1 sample reduction; (c) 8:1 sample reduction; (d) 16:1 sample reduction.

Figure Basis restriction zonal filtered images in cosine transform domain.

#### Basis restriction: Zonal Filtering with different transforms

(a) Cosine; (b) sine; (c) military DFT; (d) Hadamard; (e) Haar; (f) Slant.

Figure Basis restriction zonal filtering using different transforms with 4:1 sample reduction.

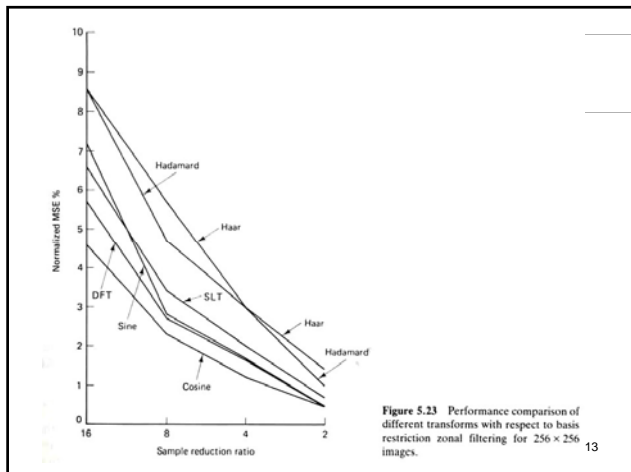


TABLE - Summary of Image Transforms	
DFT/unitary DFT	Fast transform, most useful in digital signal processing, convolution, digital filtering, analysis of circulant and Toeplitz systems. Requires complex arithmetic. Has very good energy compaction for images.
Cosine	Fast transform, requires real operations, near optimal substitute for the KL transform of highly correlated images. Useful in designing transform coders and Wiener filters for images. Has excellent energy compaction for images.
Sine	About twice as fast as the fast cosine transform, symmetric, requires real operations; yields fast KL transform algorithm which yields recursive block processing algorithms, for coding, filtering, and so on; useful in estimating performance bounds of many image processing problems. Energy compaction for images is very good.

4.114

Hadamard	Faster than sinusoidal transforms, since no multiplications are required; useful in digital hardware implementations of image processing algorithms. Easy to simulate but difficult to analyze. Applications in image data compression, filtering, and design of codes. Has good energy compaction for images.
Haar	Very fast transform. Useful in feature extraction, image coding, and image analysis problems. Energy compaction is fair.
Slant	Fast transform. Has "image-like basis"; useful in image coding. Has very good energy compaction for images

4.115

Karhunen-Loeve	Is optimal in many ways, has no fast algorithm; useful in performance evaluation and for finding performance bounds. Useful for small size vectors e.g., color multispectral or other feature vectors. Has the best energy compaction in the mean square sense over an ensemble.
Fast KL	Useful for designing fast, recursive-block processing techniques, including adaptive techniques. Its performance is better than independent block-by-block processing techniques.
SVD transform	Best energy-packing efficiency for any given image. Varies drastically from image to image; has no fast algorithm or a reasonable fast transform substitute; useful in design of separable FIR filters, finding least squares and minimum norm solutions of linear equations, finding rank of large matrices, and so on. Potential image processing applications are in image restoration, power spectrum estimation and data compression.

4.116

## Image Transforms

### Conclusions

1. It should often be possible to find a sinusoidal transform as a good substitute for the KL transform
2. Cosine Transform always performs best!
3. All transforms can only be appreciated if individually experimented with
4. Singular Value Decomposition (SVD) is a transform which locally (per image) achieves pretty much what the KL does for an ensemble of images (i.e., decorrelation).

4.117

## 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, conharmonic, median, min, max, mid-point, alpha-trimmed mean and simple adaptive filters)
- Linear position invariant degradations
- Inverse filtering
- Wiener filtering

## Chapter 5 Image Restoration

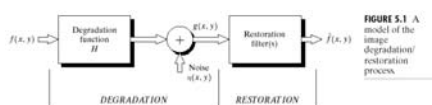


FIGURE 5.1 A model of the image degradation/restoration process.

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

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

Goal of restoration:

Find the restoration filter such that

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

## 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  $m \times n$ , 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

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

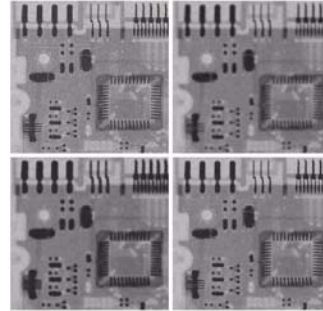
- the filter output is the median value of the input data inside the filter window.
- **Effects:** for certain types of noises, it produces excellent results with considerably less blurring than linear filters.

### Chapter 5 Image Restoration: adaptive filtering

$$f_i(x, y) = g_i(x, y) - \frac{\sigma_n^2}{\sigma_L^2} [g(x, y) - m_L]$$

$\sigma_n^2$  is the local variance of the pixels  
 $\sigma_n^2$  is the variance of the noise corrupting  $f(x, y)$   
 $m_L$  is the local mean

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  $7 \times 7$ .



#### 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:

- $z_{\min}$  = minimum gray level value in  $S_{xy}$
- $z_{\max}$  = maximum gray level value in  $S_{xy}$
- $z_{\text{med}}$  = median of gray levels in  $S_{xy}$
- $z_{xy}$  = gray level at coordinates  $(x, y)$
- $S_{\max}$  = 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:
- $A1 = z_{\text{med}} - z_{\min}$
  - $A2 = z_{\max} - z_{\text{med}}$
  - 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:
- $B1 = z_{xy} - z_{\min}$
  - $B2 = z_{\max} - z_{xy}$
  - If  $B1 > 0$  AND  $B2 < 0$ , output  $z_{xy}$
  - Else output  $z_{\text{med}}$ .

### 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 :

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

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: Optimum notch filtering**

**Optimization Problem:**

Select/find  $w(x,y)$  to minimize  $Var\{\hat{f}\}$  over a specific neighborhood of each pixel, i.e.,  
 $\min E[(\hat{f} - E[\hat{f}])^2]$  over  $S(x,y)$

where  $S(x,y)$  is fixed neighborhood at pixel  $(x,y)$ .

**Solution:**

Let  $S(x,y)$  be of size  $N=(2a+1)(2b+1)$  pixels around  $(x,y)$ . Then

$$Var\{\hat{f}\} = \sigma^2(x,y) = \frac{1}{N} \sum_{s=-a}^a \sum_{t=-b}^b [\hat{f}(x+s,y+t) - \bar{f}(x,y)]^2$$

Where  $\bar{f}(x,y)$  is the average of  $\hat{f}(x,y)$  in  $S(x,y)$ , i.e.,

$$\bar{f}(x,y) = \frac{1}{N} \sum_{s=-a}^a \sum_{t=-b}^b \hat{f}(x+s,y+t)$$

**Chapter 5**  
**Image Restoration: Optimum notch filtering**

Rewriting  $Var\{\hat{f}\}$

$$\sigma^2(x,y) = \frac{1}{N} \sum_{s=-a}^a \sum_{t=-b}^b \{ [g(x+s,y+t) - w(x+s,y+t)\eta(x+s,y+t)] - [\bar{g}(x,y) - \overline{w(x,y)\eta(x,y)}] \}^2$$

Assume that  $w(x,y)$  remains constant over  $S(x,y)$ , then

$$w(x+s,y+t) = w(x,y) \text{ and } \overline{w(x,y)\eta(x,y)} = w(x,y)\bar{\eta}(x,y) \text{ inside } S$$

Then,

$$\sigma^2(x,y) = \frac{1}{N} \sum_{s=-a}^a \sum_{t=-b}^b \{ [g(x+s,y+t) - w(x,y)\eta(x+s,y+t)] - [\bar{g}(x,y) - w(x,y)\bar{\eta}(x,y)] \}^2$$

To minimize  $\sigma^2(x,y)$ , we solve  $\frac{\partial \sigma^2(x,y)}{\partial w(x,y)} = 0$  for  $w(x,y)$ .

Result:

$$w(x,y) = \frac{\bar{g}(x,y)\bar{\eta}(x,y) - \overline{g(x,y)\eta(x,y)}}{\bar{\eta}^2(x,y) - \overline{\eta^2(x,y)}}$$

Finally,

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

**Chapter 5**  
**Image Restoration: Linear, Position-invariant Degrations**

Recall the image degraation/restoration process

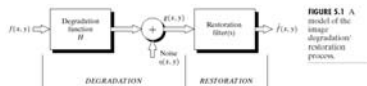


FIGURE 5.1 A model of the image degradation/restoration process.

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  $N(\dots)$ !

**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.

**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:

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

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:

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

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

Here,  $1/H(f)$  is the inverse of the original system, and  $\text{SNR}(f) = S(f)/N(f)$  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.

- (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)



**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 with noise variance reduced by five orders of magnitude from (a). Note in (d) how the deblurred image is quite similar to the original image.

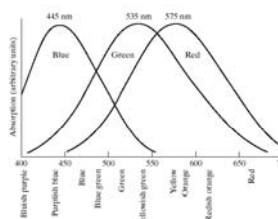
## Chapter 6: Color Image Processing

- 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:



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

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

- 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 produce secondary colors of light: e.g. magenta, cyan and yellow. Mixing the three primaries produces white color.



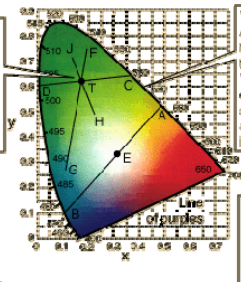
A primary color of pigments or colorants is defined as one that subtracts or absorbs a primary color of light and reflects the other two.

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

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

## 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. AFB or any pair for which the connecting line passes through E can form a complementary color pair.

### Chapter 6 Color Image Processing

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 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

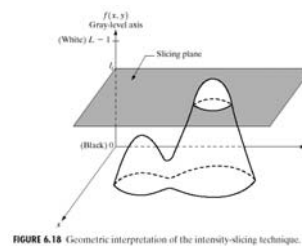
#### Three Perceptual Measures

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



assign different colors to levels above and below the slicing plane

usually, several levels are used.

FIGURE 6.18 Geometric interpretation of the intensity-slicing technique.

Chapter 6  
Color Image Processing: Gray level to color transformation

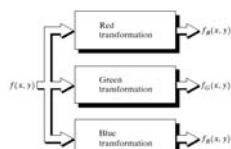


FIGURE 6.23 Functional block diagram for pseudocolor image processing.  $f_R$ ,  $f_G$ , and  $f_B$  are fed into the corresponding red, green, and blue inputs of an RGB color monitor.

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:

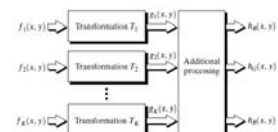
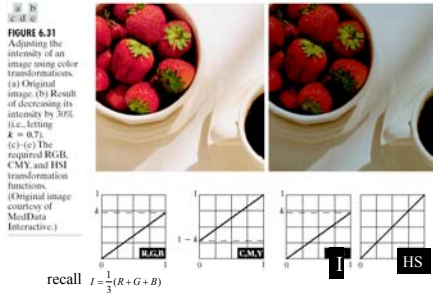


FIGURE 6.26 A pseudocolor coding approach used when several monochrome images are available.

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 1 component is transformed.



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

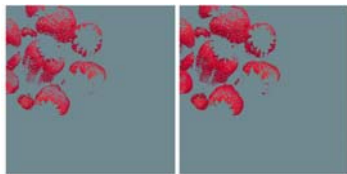
How can this be done? Can map the colors outside some range of interest to some neutral color and leave the rest as they are. Let  $w = (a_1, a_2, a_3)$  be the average of the color region of interest and  $W$  the width of this region, then

$$s_i = \begin{cases} 0.5 & \text{if } |r_j - a_j| > \frac{W}{2} \\ r_j & \text{for } i=1,2,3 \end{cases}$$

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

$$s_i = \begin{cases} 0.5 & \text{if } \sum_{j=1}^3 (r_j - a_j)^2 > R_0^2 \\ r_j & \end{cases}$$

Chapter 6  
Color Image Processing: color slicing example



**FIGURE 6.34** Color slicing transformations that detect (a) reds within an RGB cube of width  $W = 0.2540$  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  $(85, 85, 85)$ .

Chapter 6  
Color Image Processing: Tone and color corrections

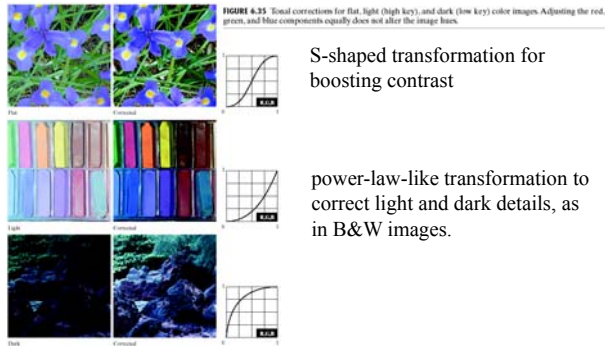
Goal: correct color image through pixel transformations to get a better visualization and / or print out.

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

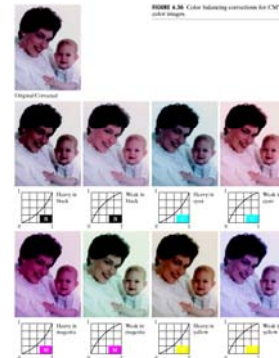
Like HSI,  $L^*a^*b^*$  decouples intensity from color

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

Chapter 6  
Color Image Processing: Tonal transformations



Chapter 6  
Color Image Processing: color balancing

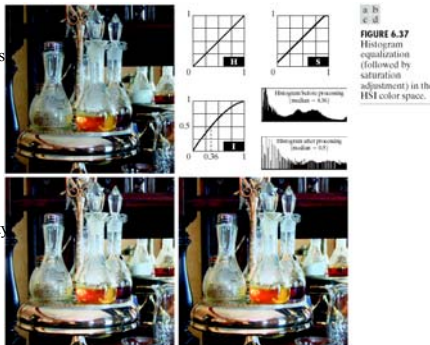


Chapter 6  
Color Image Processing: Histogram equalization

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.



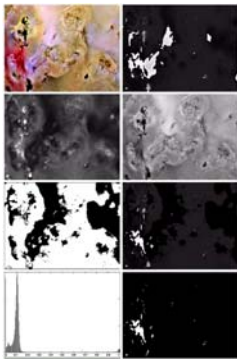
Chapter 6  
Color Image Processing: color image sharpening



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)

**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**

**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!