EL5123 --- Image Processing

Midterm Summary
Fall 2011

Yao Wang
Polytechnic University, Brooklyn, NY 11201

Topics Covered

- Image representation
- Color representation
- Quantization
- Contrast enhancement
- Spatial Filtering: noise removal, sharpening, edge detection
- Frequency domain representations
  - FT, DTFT, DFT, unitary transforms
  - Implementation of linear filtering using DTFT and DFT

Color Image Representation

- Light is the visible band of the EM wave
  - Color hue of a light depends on the wavelength of the EM wave
  - Illuminating light vs. reflecting light
- Color attributes:
  - Brightness (luminance)
  - Hue (depends on the wavelength)
  - Saturation (purity)
- Any color can be reproduced by mixing three primary colors
  - Illuminating and reflecting light sources follow different mixing rules
- Color can be represented in many different coordinates (or models) with 3 components

Tri-chromatic Color Mixing

- Tri-chromatic color mixing theory
  - Any color can be obtained by mixing three primary colors with a right proportion
- Primary colors for illuminating sources:
  - Red, Green, Blue (RGB)
  - Color monitor works by exciting red, green, blue phosphors using separate electronic guns
  - follows additive rule: R+G+B=White
- Primary colors for reflecting sources (also known as secondary colors):
  - Cyan, Magenta, Yellow (CMY)
  - Color printer works by using cyan, magenta, yellow and black (CMYK) dyes
  - follows subtractive rule: R+G+B=Black

Color Models

- Specify three primary or secondary colors
  - Red, Green, Blue.
  - Cyan, Magenta, Yellow.
- Specify the luminance and chrominance
  - HSB or HSI (Hue, saturation, and brightness or intensity)
  - YIQ (used in NTSC color TV)
  - YCbCr (used in digital color TV)
- Amplitude specification:
  - 8 bits per color component, or 24 bits per pixel
  - Total of 16 million colors
  - A 1kx1k true RGB color requires 3 MB memory

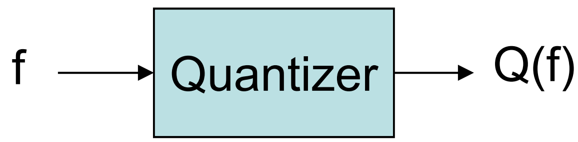
Color Image Processing

- Apply contrast enhancement or filtering (linear or non-linear) to each primary color component independently using the techniques for monochrome images
  - May change the color hue of the original image
- Convert the tri-stimulus representation into a luminance / chrominance representation, and modify the luminance component only.
- For certain applications, different operations may be applied to different color components (either in RGB domain or in YC1C2 domain) to obtain special desired effects

Pseudo Color Processing

- Given a gray-scale image, we may display it in pseudo colors to better reveal certain features
  - Color depends on image pixel value
    - Often used in medical image display, weather maps, vegetation maps
  - Decompose an image into different frequency bands and represent each band with different colors

Quantization



Decision Levels $\{t_k, k = 1, \ldots, L+1\}$

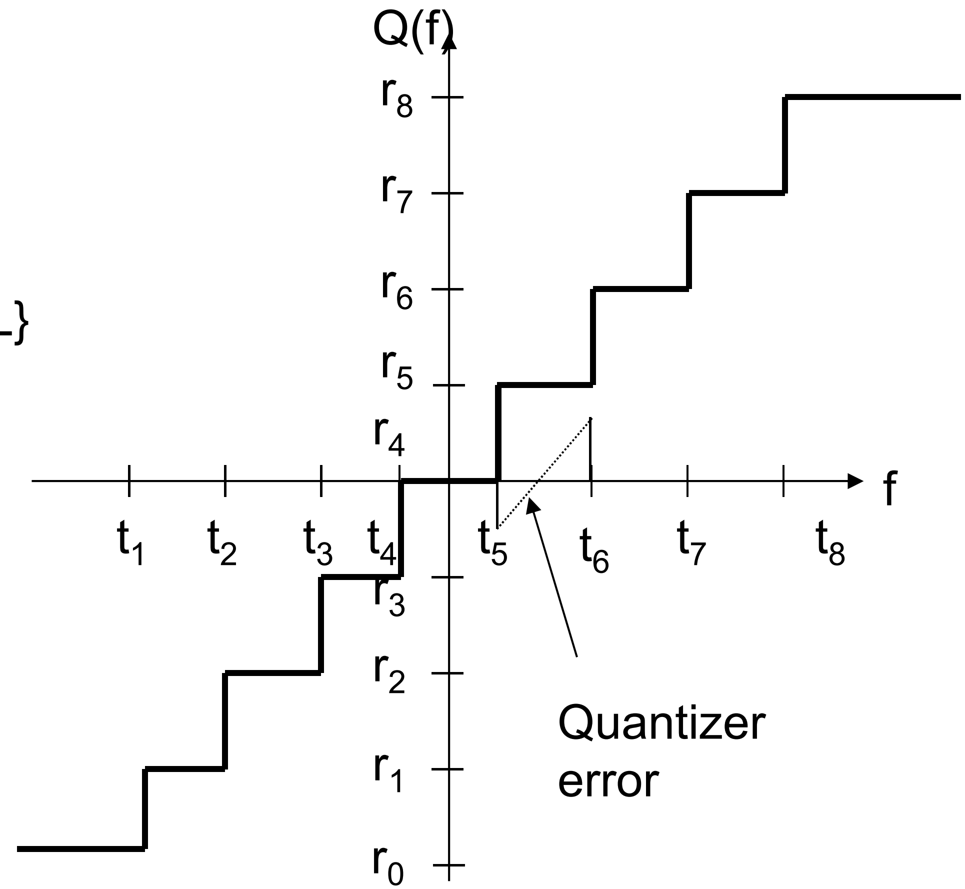
Reconstruction Levels $\{r_k, k = 1, \ldots, L\}$

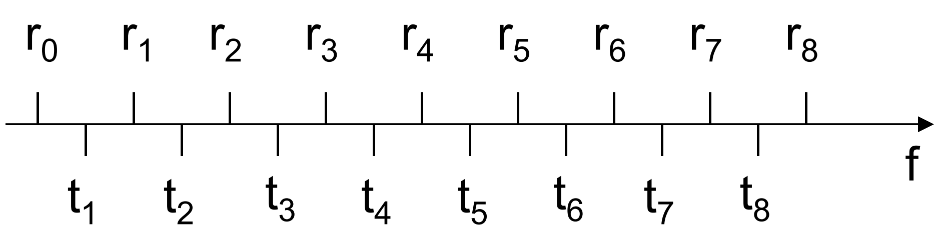
If $f \in [t_k, t_{k+1})$

Then $Q(f) = r_k$

L levels need $R = \lceil \log_2 L \rceil$ bits

$\lceil x \rceil$ returns the smallest integer that is bigger than or equal to x





Uniform Quantization

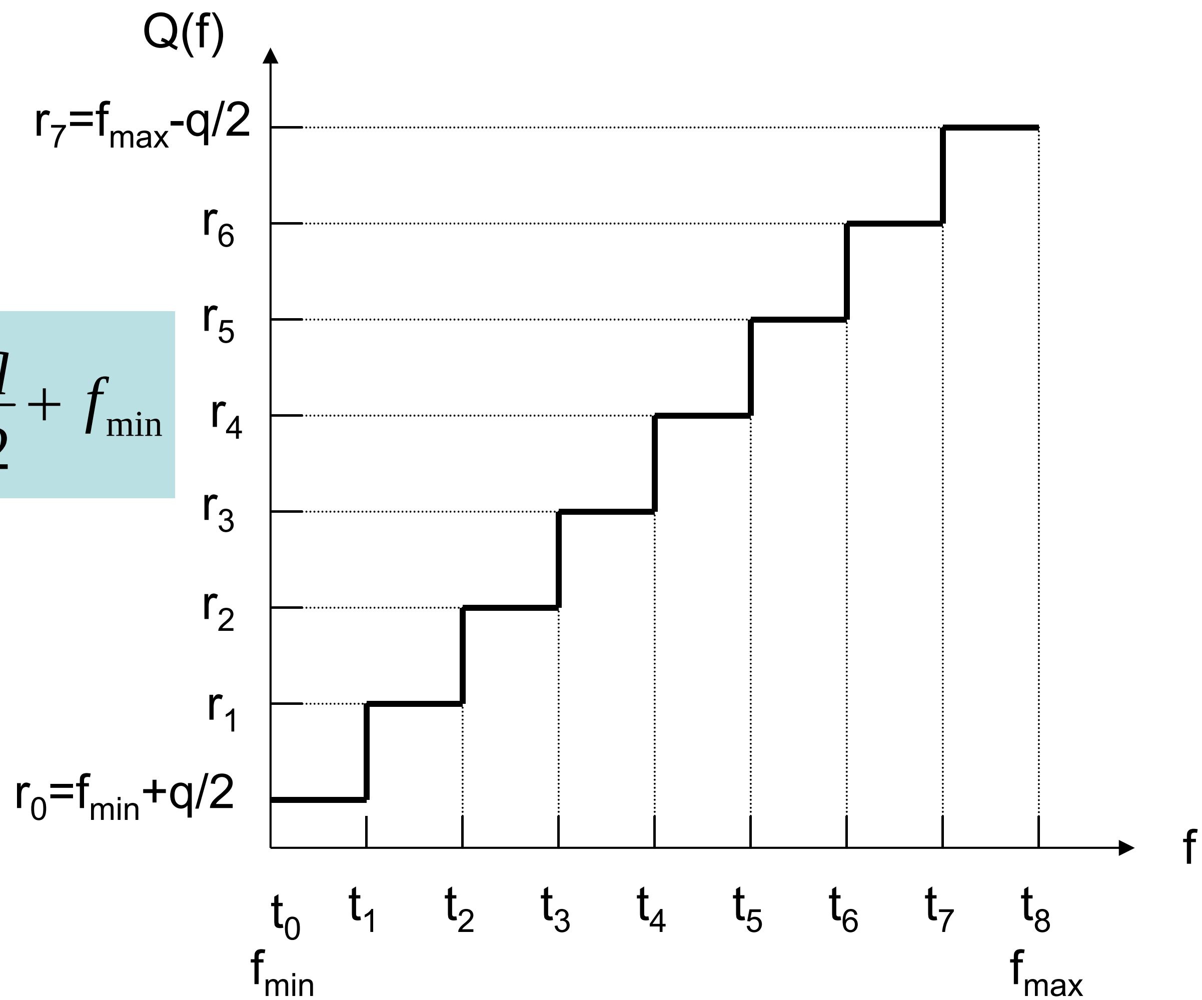
- Equal distances between adjacent decision levels and between adjacent reconstruction levels
  - $t_l - t_{l-1} = r_l - r_{l-1} = q$
- Parameters of Uniform Quantization
  - L: levels ($L = 2^R$)
  - B: dynamic range $B = f_{max} - f_{min}$
  - q: quantization interval (step size)
  - $q = B/L = B2^{-R}$

Uniform Quantization: Functional Representation

stepsize $q=(f_{max}-f_{min})/L$

$$Q(f)=\left\lfloor \frac{f-f_{\min}}{q} \right\rfloor * q+\frac{q}{2}+f_{\min}$$

$\lfloor x \rfloor$ returns the biggest integer that is smaller than or equal to x



$I(f)=\left\lfloor \frac{f-f_{\min}}{q} \right\rfloor$ is called the reconstruction level index, which indicates which reconstruction level is used for f.

MSE of a Quantizer

- Mean square error (MSE) of a quantizer for a continuous valued signal

$$MSE = \sigma_q^2 = E\{(Q(f) - f)^2\} =$$
$$\int_{t_0}^{t_L} (f - Q(f))^2 p(f) df = \sum_{l=0}^{L-1} \int_{t_l}^{t_{l+1}} (f - r_l)^2 p(f) df$$

  – Where p(f) is the probability density function of f

- MSE for a specific image

$$MSE = \sigma_q^2 = \frac{1}{MN} \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} (f(i,j) - Q(f(i,j)))^2$$

MSE of a Uniform Quantizer for A Uniform Source

$$p(f)=\begin{cases} 1/(f_{\max}-f_{\min})=1/B, & f\in(f_{\min},f_{\max}) \\ 0, & otherwise \end{cases}$$

$$\Rightarrow \quad \sigma_f^2=\int_{f_{\min}}^{f_{\max}}(f-\eta)^2\frac{1}{B}df=\frac{B^2}{12}$$

Uniform quantization into L levels: $q=B/L=B/L$

Error in each bin is the same, and is uniformly distributed in (-q/2, q/2)

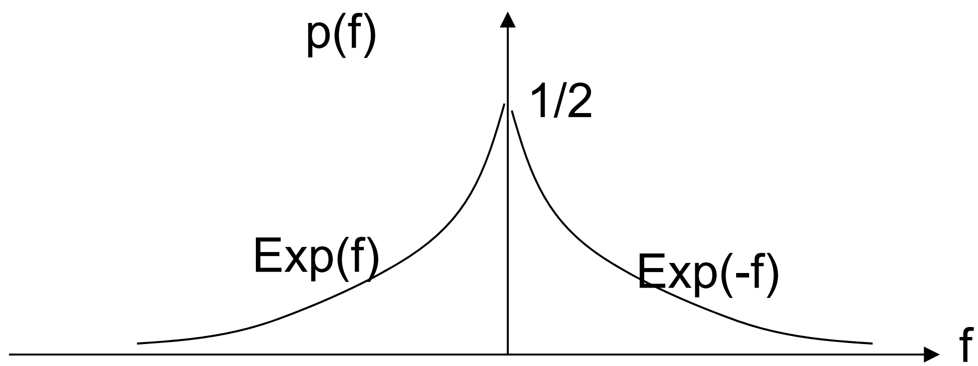
$$\Rightarrow \quad \sigma_q^2=\int_{-q/2}^{q/2}e^2\frac{1}{q}de=\frac{q^2}{12}=\frac{\left(B/L\right)^2}{12}=\frac{B^2}{12}2^{-2R}=\sigma_f^2 2^{-2R}$$

$$\Rightarrow \quad SNR=10\log_{10}\frac{\sigma_f^2}{\sigma_q^2}=10\log_{10}2^{2R}=20R\log_{10}2\approx 6R(dB)$$

$\Rightarrow$ Every additional bit increases the SNR by 6dB!

Example: Nonuniform Source

- The pdf of a signal is shown below, we want to quantize it to 2 levels. Determine the partition and reconstruction levels that minimizes the quantization error (in terms of MSE). Also compute the MSE and SNR.
- Go through in class.



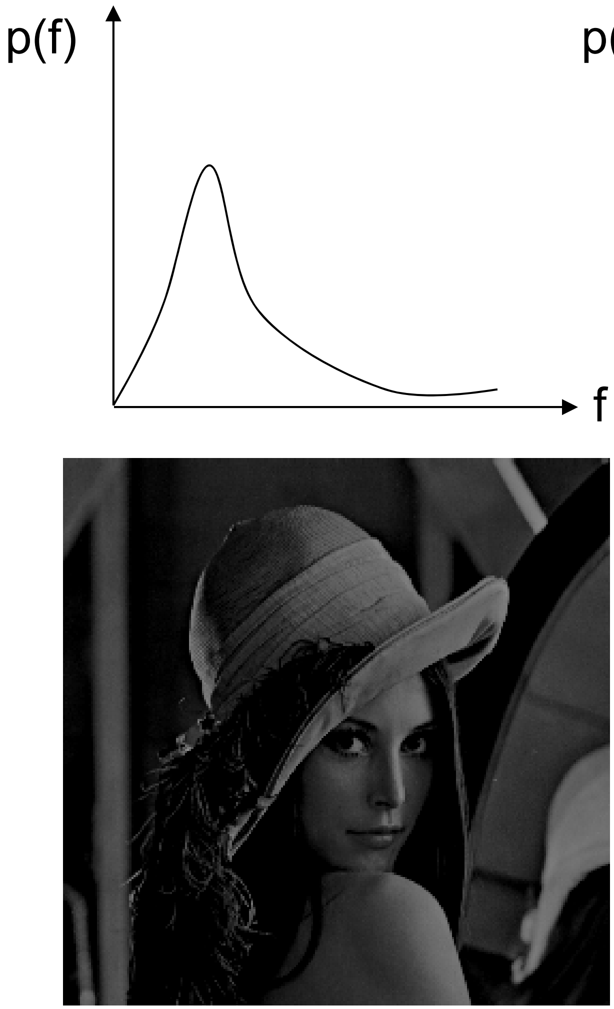
MMSE Quantizer

- For any pdf, the quantizer that minimizes MSE is known as Minimal MSE (MMSE) quantizer.
- Given simple pdf, should know how to determine the decision and reconstruction levels to minimize MSE
- Given a quantizer and signal pdf, should know how to calculate MSE

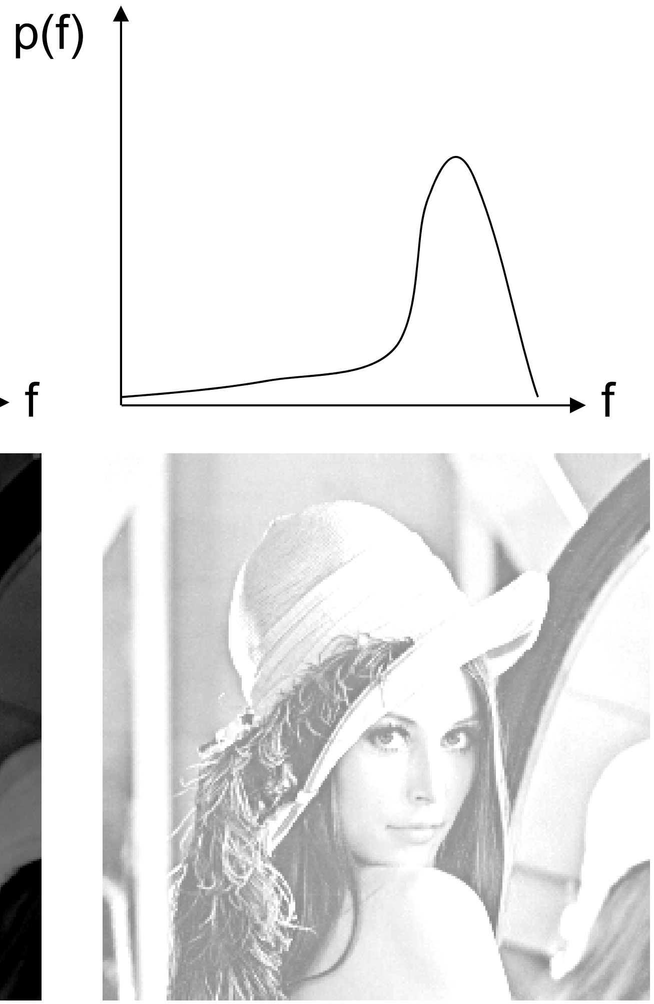
Contrast Enhancement

- An frequently used, important operation
- How to tell the contrast of an image from its histogram?
- Given a histogram, can sketch a transformation that will enhance the contrast
- Histogram equalization
- Histogram specification

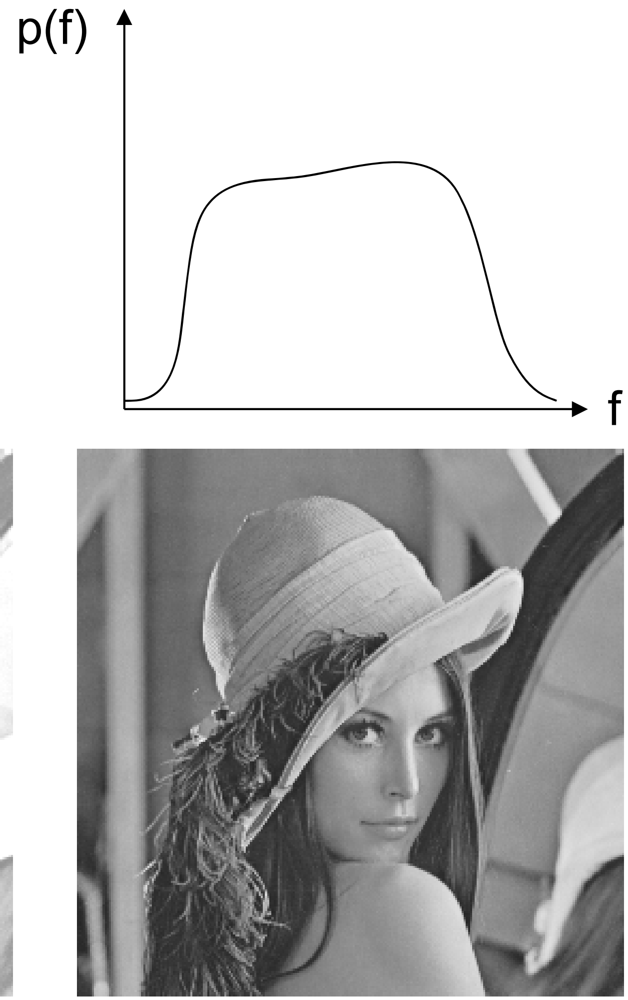
Histogram vs. Contrast



(a) Too dark

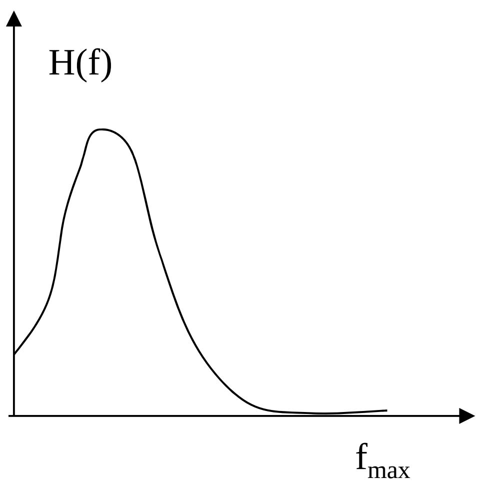


(b) Too bright

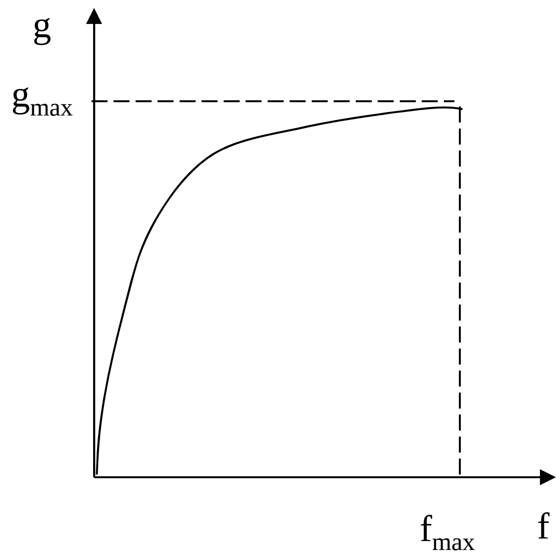


(c) Well balanced

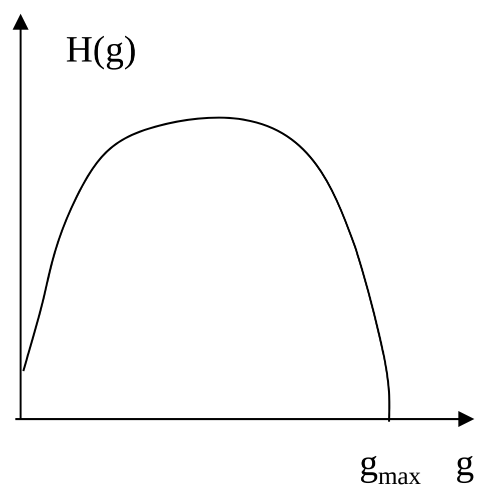
Enhancement of Too-Dark Images



Original histogram

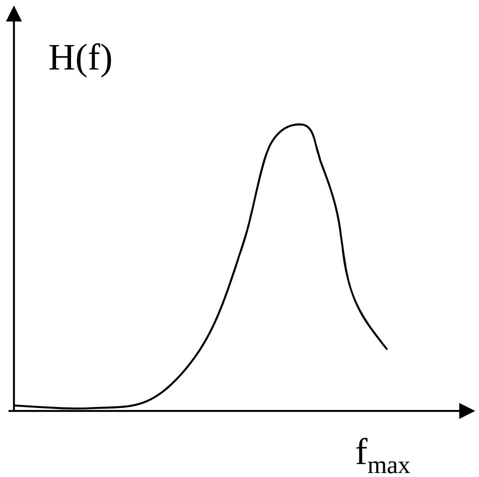


New histogram

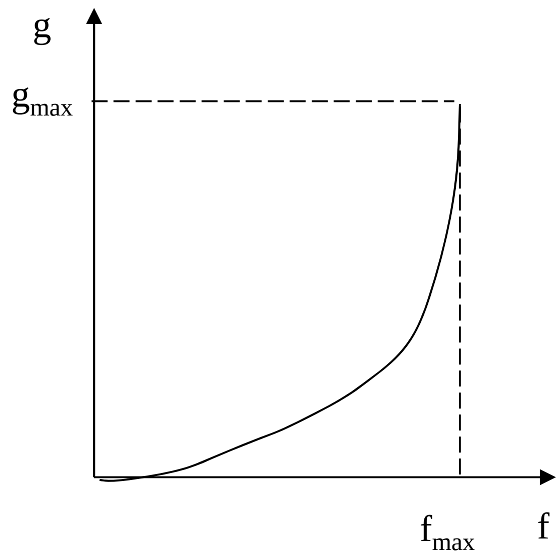


Transformation function:
"log" function: g=c log (1+f)
Or
Power law: g= c f^r, 0<r<1

Enhancement of Too-Bright Images

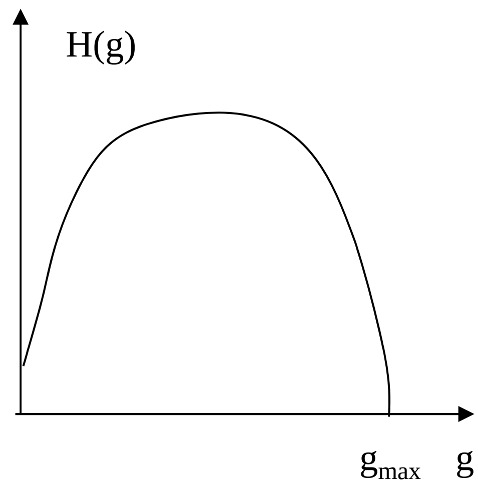


Original histogram

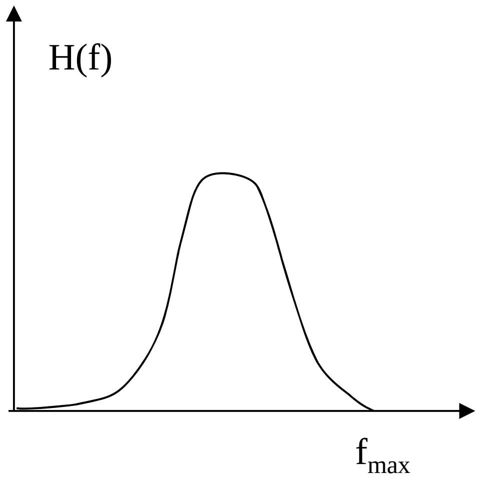


Transformation function:
Power law: g=c f^r, r>1

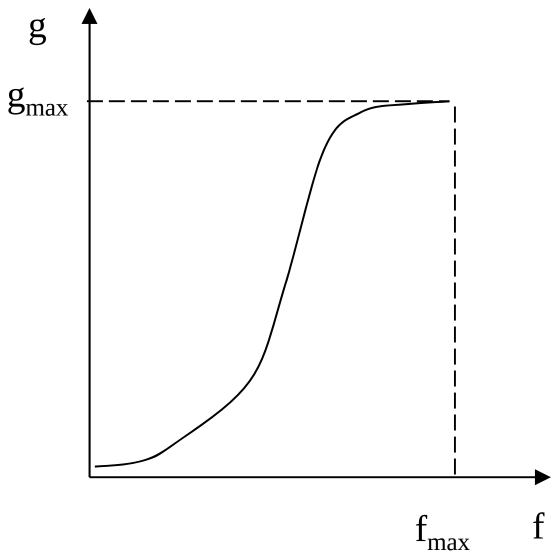
New histogram



Enhancement of Images Centered near the Middle Range

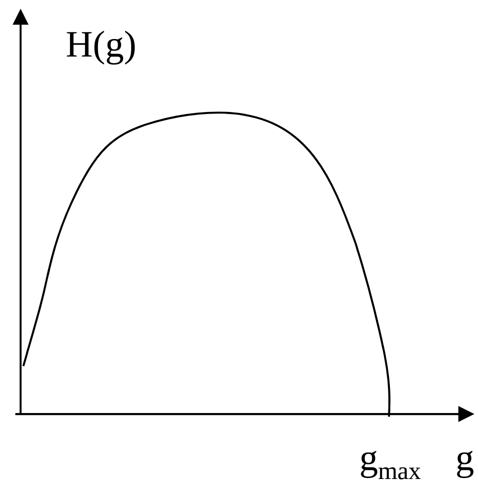


Original histogram



Transformation function

New histogram



Histogram Equalization

- Transforms an image with an arbitrary histogram to one with a flat histogram
  – Suppose f has PDF $p_F(f)$, $0 \le f \le 1$
  – Transform function (continuous version)

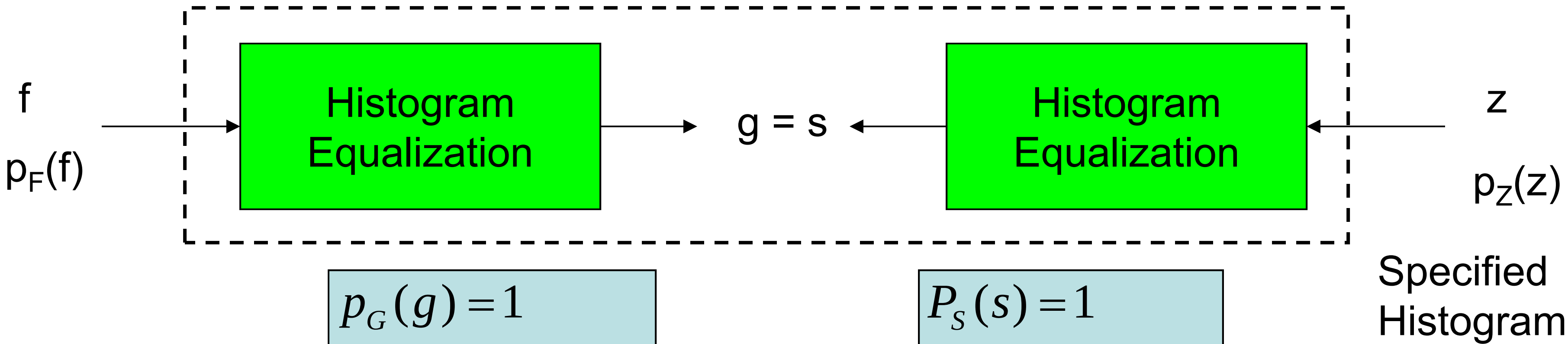
$$g(f) = \int_0^f p_F(t)dt$$

  – g is uniformly distributed in (0, 1)

Histogram Equalization

Histogram Specification

- What if the desired histogram is not flat?



$$p_G(g) = 1$$
$$g = g(f) = \int_0^f p_F(f)\,df$$

$$P_S(s) = 1$$
$$s = s(z) = \int_0^z p_Z(z)\,dz$$

$$z = s^{-1}(g) = s^{-1}(g(f))$$

Image Filtering

- Applications:
  - Noise removal (image smoothing, low-pass filtering)
  - Edge enhancement (deblurring, high-emphasis)
  - Edge detection (high-pass filtering)
- Contrast enhancement is accomplished by point operation, i.e., each pixel value is changed based on its original value, not its neighboring pixels (but the transformation function depends on the overall histogram of the image)
- Image filtering refers to changing the color value of one pixel based on the color values of this pixel and its neighbors

Three Ways of Implementing Linear Filtering

- Spatial domain
  - Weighted average of adjacent pixels = linear convolution
  - Weights or the filter depends on the desired filtering effect
- Frequency domain (FT, DTFT)
  - Design spatial filter based on desired frequency response
    - Low pass, high pass, high emphasis
  - Convolution theorem: h*f ⇔ H F
- Frequency domain (DFT)
  - Perform filtering in DFT domain
  - Convolution theorem:
    - circulation convolution h@f ⇔ H F
  - Relation between circulation convolution and linear convolution
  - Filter masks in the DFT domain must be designed properly so that the corresponding filter in the spatial domain is real
    - The filter mask should enjoy the symmetry property of real signals

Linear Convolution of Continuous Signals

- 1D convolution

$$f(x)*h(x)=\int_{-\infty}^{\infty} f(x-\alpha)h(\alpha)d\alpha=\int_{-\infty}^{\infty} f(\alpha)h(x-\alpha)d\alpha$$

  - Equalities

$$f(x)*\delta(x)=f(x),\quad f(x)*\delta(x-x_0)=f(x_0)$$

- 2D convolution

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

Linear Convolution of Discrete Signals

- 1D convolution

$$f(n)*h(n)=\sum_{m=-\infty}^{\infty} f(n-m)h(m)=\sum_{m=-\infty}^{\infty} f(m)h(n-m)$$

- 2D convolution

$$f(m,n)*h(m,n)=\sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} f(m-k,n-l)h(k,l)$$
$$=\sum_{k=-\infty}^{\infty}\sum_{l=-\infty}^{\infty} f(k,l)h(m-k,n-l)$$

- Separable filtering
  - Row first, then columns

Interpretation as weighted average of neighboring samples

- How should we design the weights?
  - Smoothing: sum=1, coef >=0
  - High emphasis: sum=1, some coef <0
  - Edge detection (high pass): sum=0

Ex: Smoothing by Averaging

- Replace each pixel by the average of pixels in a square window surrounding this pixel

$$g(m,n)=\frac{1}{9}\big(f(m-1,n-1)+f(m-1,n)+f(m-1,n+1)$$
$$+f(m,n-1)+f(m,n)+f(m,n+1)$$
$$+f(m+1,n-1)+f(m+1,n)+f(m+1,n+1)\big)$$

- Trade-off between noise removal and detail preserving:
  - Larger window -> can remove noise more effectively, but also blur the details/edges

Directional Edge Detector

- High pass in one direction and low pass in the orthogonal direction
- Prewitt edge detector

$$H_x = \frac{1}{3}\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}\begin{bmatrix} 1 & 1 & 1 \end{bmatrix}; \quad H_y = \frac{1}{3}\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} = \frac{1}{3}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

- Sobel edge detector

$$H_x = \frac{1}{4}\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}\begin{bmatrix} 1 & 2 & 1 \end{bmatrix}; \quad H_y = \frac{1}{4}\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} = \frac{1}{4}\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}\begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

Special Considerations in Implementation

- Boundary treatment
  - If f is MxN, h is KxL, convolved image g is (M+K-1)x(N+L-1)
  - Instead of expanding the image size, we modify filtering operations at the boundary
    - Symmetric expansion
    - Leave the boundary pixels unchanged
- Renormalization
  - Filtered values may not be integer, and may have negative values, may have a smaller or larger dynamic range than original
  - To save resulting image in an 8-bit unsigned char format, we normalize all values to 0-255
    - g'= (g-g_min)/(g_max-g_min) * 255

Fourier Transform For Discrete Time Sequence (DTFT)

- 1D
  - Forward Transform $$F(u) = \sum_{n=-\infty}^{\infty} f(n) e^{-j2\pi un}$$
  - Inverse Transform $$f(n) = \int_{-1/2}^{1/2} F(u) e^{j2\pi un} du$$
- 2D
  - Forward Transform $$F(u,v) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(m,n) e^{-j2\pi(mu+nv)}$$
  - Inverse Transform $$f(m,n) = \int_{-1/2}^{1/2} \int_{-1/2}^{1/2} F(u,v) e^{j2\pi(mu+nv)} dudv$$
- Separable implementation
  Transform each row first with 1D DTFT, then each column

Design Filters Based on Desired Frequency Response

- Convolution theorem

$$f(x,y)*h(x,y) \quad \Leftrightarrow \quad F(u,v)H(u,v)$$

- Filter h(x,y) designed based on desired frequency response H(u,v)
  - Sharp transitions in frequency domain ⇔very long filters in spatial domain
  - Apply a window function to smooth the transition band ⇔ shorter filters
- Giver a spatial filter, we can use DTFT to better understand its filtering effect (frequency response)
- Separable filters
  - Design horizontal and vertical filters separately
  - For images, we usually use very short filters

Discrete Fourier Transform (DFT): DTFT for Finite Duration Signals

If the signal is only defined for $n = 0,1,\ldots, N-1$:

Fourier transform becomes:

$$F'(f) = \sum_{n=0}^{N-1} f(n)\exp(-j2\pi fn), \quad f \in (0,1)$$

Sampling $F'(f)$ at $f = k/N$, $k = 0,1,\ldots,N\text{-}1$, and rescaling yields:

Forward transform (DFT):

$$F(k) = F'\left(\frac{k}{N}\right) = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} f(n)\exp\left(-j2\pi \frac{k}{N} n\right), \quad k = 0,1,\ldots, N-1$$

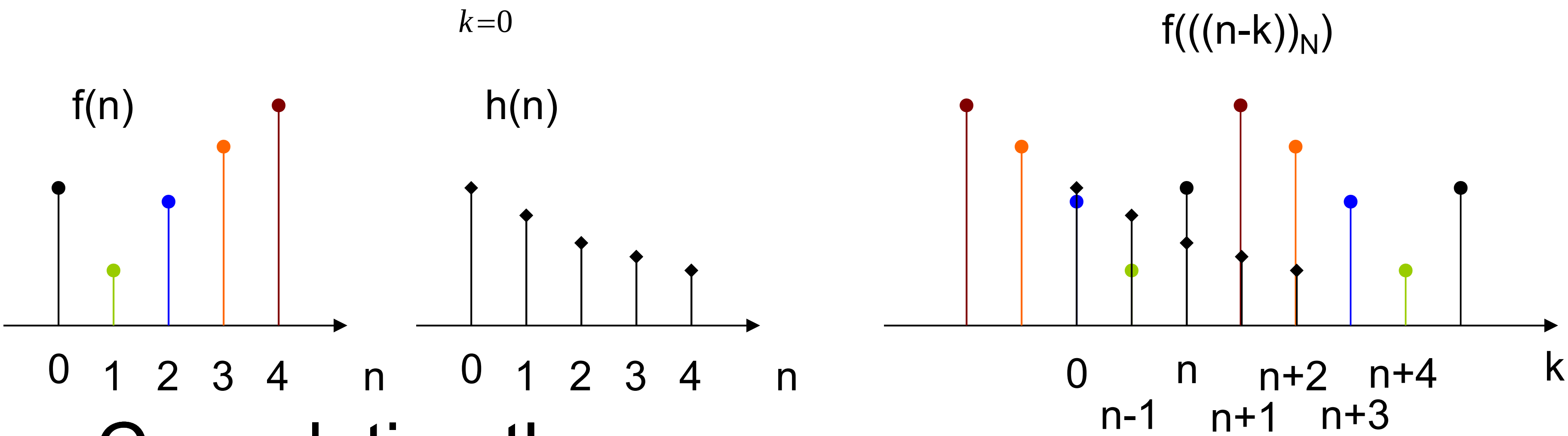
Inverse transform (IDFT):

$$f(n) = \frac{1}{\sqrt{N}} \sum_{k=0}^{N-1} F(k)\exp\left(j2\pi \frac{k}{N} n\right), \quad n = 0,1,\ldots, N-1$$

Circular Convolution Theorem

- Circular convolution

$$f(n) \otimes h(n) = \sum_{k=0}^{N-1} f(((n-k))_N)h(k)$$



- Convolution theorem

$$f(n) \otimes h(n) \quad \Leftrightarrow \quad F(k)H(k)$$

2D Discrete Fourier Transform

- Definition
  - Assuming f(m, n), m = 0, 1, …, M-1, n = 0, 1, …, N-1, is a finite length 2D sequence

$$F(k,l) = \frac{1}{\sqrt{MN}} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m,n) e^{-j2\pi\left(\frac{km}{M}+\frac{ln}{N}\right)}, \quad k = 0,1,\ldots,M-1, l = 0,1,\ldots,N-1;$$

$$f(m,n) = \frac{1}{\sqrt{MN}} \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} F(k,l) e^{j2\pi\left(\frac{km}{M}+\frac{ln}{N}\right)}, \quad m = 0,1,\ldots,M-1, n = 0,1,\ldots,N-1.$$

  - Comparing to DTFT

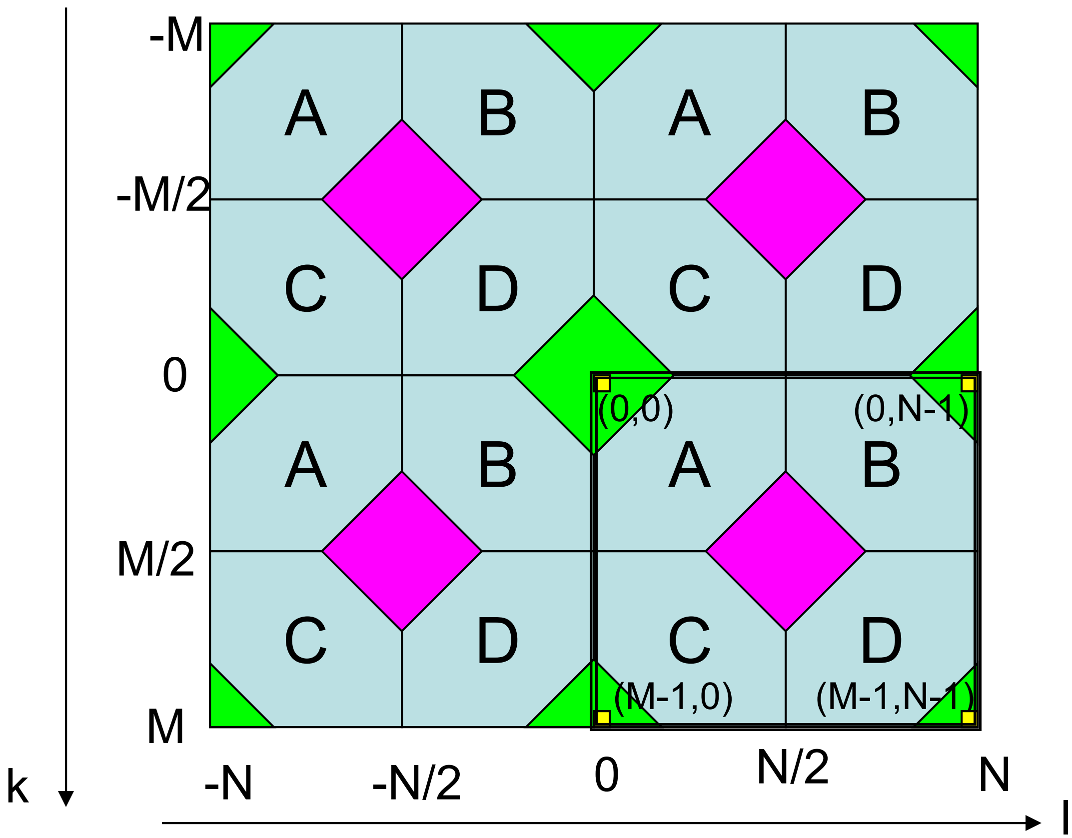
$$F(u,v) = \sum_{m=-\infty}^{\infty} \sum_{n=-\infty}^{\infty} f(m,n) e^{-j2\pi(mu+nv)}, \quad u = \frac{k}{M}, v = \frac{l}{N}$$

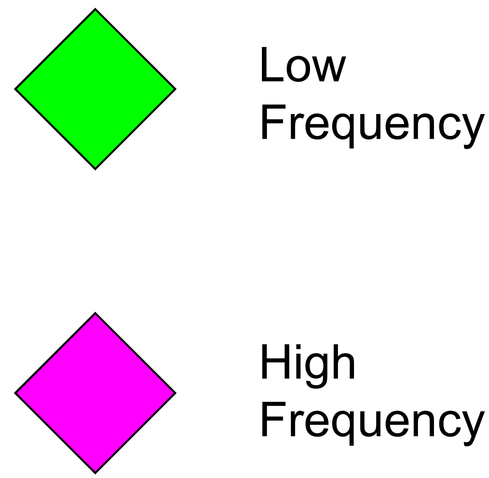
$$f(m,n) = \int_{-1/2}^{1/2} \int_{-1/2}^{1/2} F(u,v) e^{j2\pi(mu+nv)} du dv$$

Periodicity Property of 2D DFT

- Periodicity

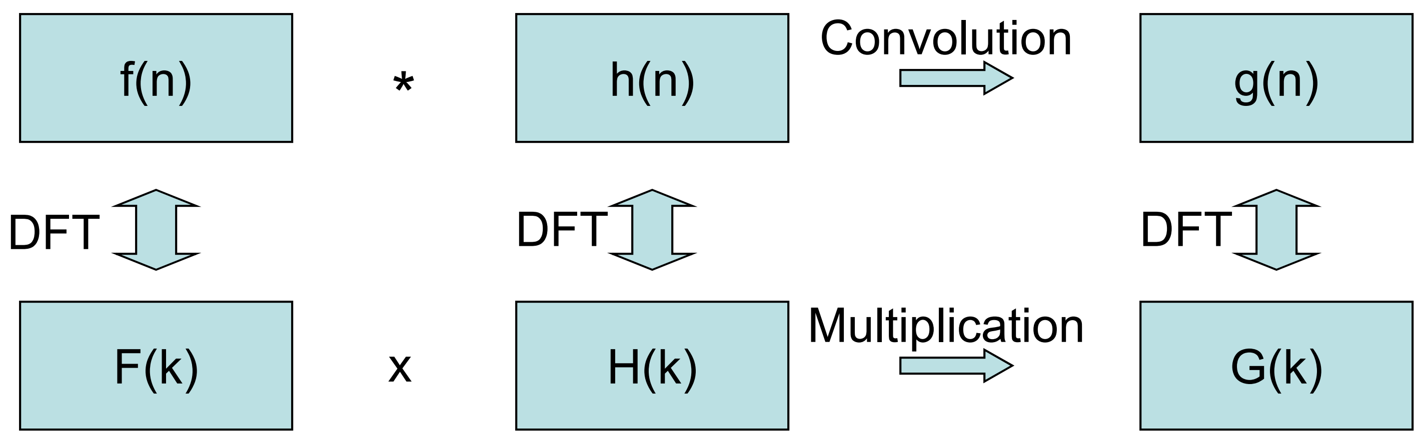
$$F(k,l) = F(((k))_M, ((l))_N), \quad k < 0 \; or \; k \geq M, l < 0 \; or \; l \geq N.$$





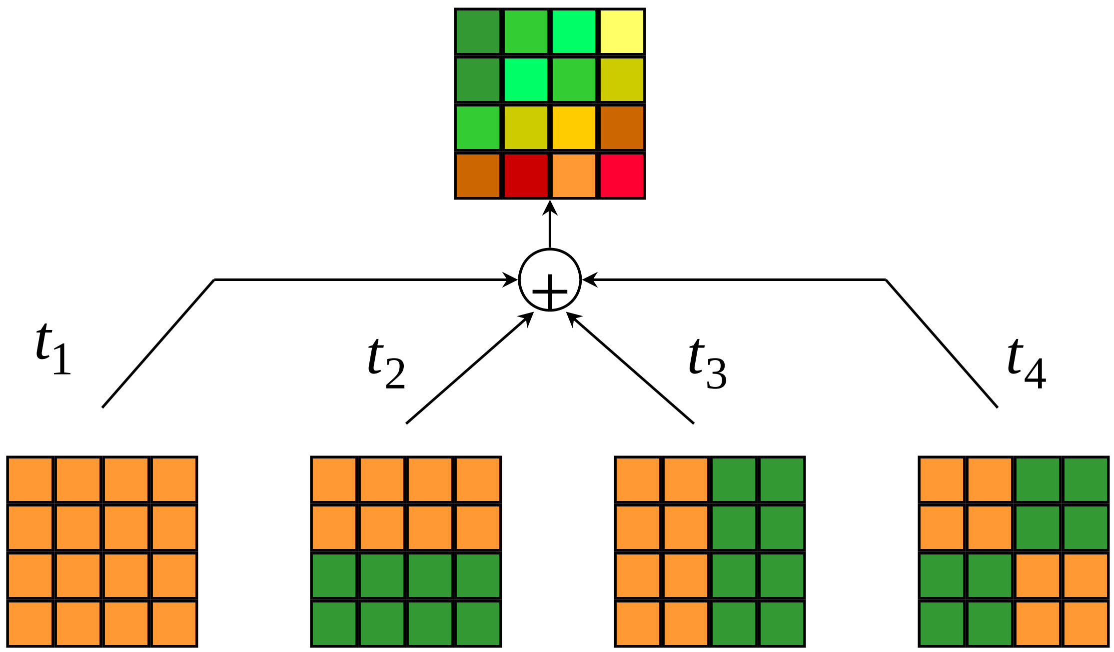
Calculate Linear Convolution Using DFT

- 1D case
  - f(n) is length N_1, h(n) is length N_2
  - g(n) = f(n)*h(h) is length $N = N_1+N_2-1$.
  - To use DFT, need to **extend** f(n) and h(n) to length N by zero padding.



What is a Linear Transform

- Represent an image (or an image block) as the linear combination of some basis images and specify the linear coefficients.



One Dimensional Linear Transform

- Let C^N represent the N dimensional complex space.
- Let $\mathbf{h}_0$, $\mathbf{h}_1$, …, $\mathbf{h}_{N-1}$ represent N linearly independent vectors in C^N.
- For any vector $\mathbf{f} \in C^N$,

$$\mathbf{f} = \sum_{k=0}^{N-1} t(k)\mathbf{h}_k = \mathbf{B}\mathbf{t},$$

$$where \quad \mathbf{B} = [\mathbf{h}_0, \mathbf{h}_1, \ldots, \mathbf{h}_{N-1}], \mathbf{t} = \begin{bmatrix} t(0) \\ t(1) \\ \vdots \\ t(N-1) \end{bmatrix}.$$

$\Rightarrow$

$$\mathbf{t} = \mathbf{B}^{-1}\mathbf{f} = \mathbf{A}\mathbf{f}$$

f and t form a transform pair

Orthonormal Basis Vectors (OBV)

- $\{\mathbf{h}_k, k=0,\ldots N-1\}$ are OBV if

$$<\mathbf{h}_k, \mathbf{h}_l> = \delta_{k,l} = \begin{cases} 1 & k = l \\ 0 & k \neq l \end{cases}$$

$$<\mathbf{h}_l, \mathbf{f}> = <\mathbf{h}_l, \sum_{k=0}^{N-1} t(k)\mathbf{h}_k> = \sum_{k=0}^{N-1} t(k) <\mathbf{h}_l, \mathbf{h}_k> = t(l) = \mathbf{h}_l{}^H \mathbf{f}$$

$$\mathbf{t} = \begin{bmatrix} \mathbf{h}_0{}^H \\ \mathbf{h}_1{}^H \\ \vdots \\ \mathbf{h}_{N-1}{}^H \end{bmatrix} \mathbf{f} = \mathbf{B}^H \mathbf{f} = \mathbf{A}\mathbf{f}$$

$$\mathbf{B}^{-1} = \mathbf{B}^H, \ or\ \mathbf{B}^H \mathbf{B} = \mathbf{B}\mathbf{B}^H = \mathbf{I}.$$

B is unitary

Definition of Unitary Transform

- Basis vectors are orthonormal
- $t(k) = \langle \mathbf{h}_k, \mathbf{f} \rangle = \sum_{n=0}^{N-1} h_k(n)^* f(n),$

$$\mathbf{t} = \begin{bmatrix} \mathbf{h}_0^H \\ \mathbf{h}_1^H \\ \vdots \\ \mathbf{h}_{N-1}^H \end{bmatrix} \mathbf{f} = \mathbf{B}^H \mathbf{f} = \mathbf{A}\mathbf{f}$$

- $f(n) = \sum_{k=0}^{N-1} t(k) h_k(n),$

$$\mathbf{f} = \sum_{k=0}^{N-1} t(k)\mathbf{h}_k = [\mathbf{h}_0 \quad \mathbf{h}_1 \quad \cdots \quad \mathbf{h}_{N-1}]\mathbf{t} = \mathbf{B}\mathbf{t} = \mathbf{A}^H \mathbf{t}$$

Two Dimensional Unitary Transform

- $\{H_{k,l}\}$ is an orthonormal set of basis images
- Forward transform

$$T(k,l) = <\mathbf{H}_{k,l}, \mathbf{F}> = \sum_{m=0}^{M-1}\sum_{n=0}^{N-1} H_{k,l}^*(m,n) F(m,n)$$

- Inverse transform

$$F(m,n) = \sum_{k=0}^{M-1}\sum_{l=0}^{N-1} T(k,l) H_{k,l}(m,n), \quad or$$

$$\mathbf{F} = \sum_{k=0}^{M-1}\sum_{l=0}^{N-1} T(k,l)\mathbf{H}_{k,l}$$

Example of 2D Unitary Transform

$$\mathbf{H}_{00} = \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix}, \mathbf{H}_{01} = \begin{bmatrix} 1/2 & 1/2 \\ -1/2 & -1/2 \end{bmatrix}, \mathbf{H}_{10} = \begin{bmatrix} 1/2 & -1/2 \\ 1/2 & -1/2 \end{bmatrix}, \mathbf{H}_{11} = \begin{bmatrix} 1/2 & -1/2 \\ -1/2 & 1/2 \end{bmatrix},$$

$$\mathbf{F} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \Rightarrow \begin{cases} T(0,0) = 5 \\ T(0,1) = -2 \\ T(1,0) = -1 \\ T(1,1) = 0 \end{cases}$$

Separable Unitary Transform

- Let $\mathbf{h}_k$, k=0, 1, …, M-1 represent a set of orthonormal basis vectors in C^M,
- Let $\mathbf{g}_l$, l=0, 1, …, N-1 represent another set of orthonormal basis vectors in C^N,
- Let $\mathbf{H}_{k,l}=\mathbf{h}_k\mathbf{g}_l^T$, or $H_{k,l}(m,n)=h_k(m)g_l(n)$.
- Then $\mathbf{H}_{k,l}$ will form an orthonormal basis set in C^{MxN}.
- Transform can be performed separately, first row wise, then column wise

Example of Separable Unitary Transform

- Example 1

$$\mathbf{h}_0 = \begin{bmatrix} 1/\sqrt{2} \\ 1/\sqrt{2} \end{bmatrix}, \mathbf{h}_1 = \begin{bmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \end{bmatrix}.$$

$$\mathbf{H}_{00} = \mathbf{h}_0 \mathbf{h}_0^T = \begin{bmatrix} 1/2 & 1/2 \\ 1/2 & 1/2 \end{bmatrix} \qquad \mathbf{H}_{01} = \mathbf{h}_0 \mathbf{h}_1^T = \begin{bmatrix} 1/2 & -1/2 \\ 1/2 & -1/2 \end{bmatrix}$$

$$\mathbf{H}_{10} = \mathbf{h}_1 \mathbf{h}_0^T = \begin{bmatrix} 1/2 & 1/2 \\ -1/2 & -1/2 \end{bmatrix} \quad \mathbf{H}_{11} = \mathbf{h}_1 \mathbf{h}_1^T = \begin{bmatrix} 1/2 & -1/2 \\ -1/2 & 1/2 \end{bmatrix}$$

- 2D DFT

$$H_{k,l}(m,n) = \frac{1}{\sqrt{MN}} e^{j2\pi\left(\frac{km}{M}+\frac{ln}{N}\right)},$$

$$h_k(m) = \frac{1}{\sqrt{M}} e^{j2\pi\frac{km}{M}}, \quad g_l(n) = \frac{1}{\sqrt{N}} e^{j2\pi\frac{ln}{N}}$$

Why Using Transform?

- When the transform basis is chosen properly
  - Many coefficients have small values and can be quantized to 0 w/o causing noticeable artifacts
  - The coefficients are uncorrelated, and hence can be coded independently w/o losing efficiency.

Midterm Exam Logistics

- Scheduled time: 11/2 3-5:40, RH615
- Closed-book, 1 sheet of notes allowed (double sided OK)
- Special Office hour
  - Monday 10/31 3-5PM or by appointment