EE421/521
Image Processing

Lecture 5
FREQUENCY DOMAIN PROCESSING

1

Spatial Frequency & HVS

2

Spatial Frequency

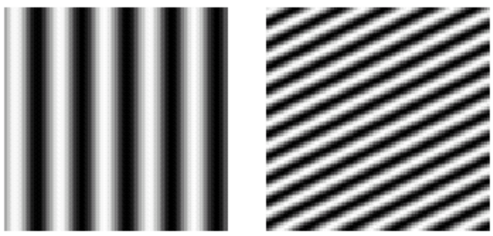
- Spatial frequency measures how fast the image intensity changes in the image plane
- Spatial frequency can be completely characterized by the variation frequencies in two orthogonal directions (e.g., horizontal and vertical)
  - f_x : cycles/horizontal unit distance
  - f_y : cycles/vertical unit distance
- Horizontal and vertical frequency can be combined and expressed in terms of magnitude and angle:

$$f_m = \sqrt{\left(f_x^2 + f_y^2\right)}$$

$$\theta = \arctan\left(\frac{f_y}{f_x}\right)$$

3

Spatial Frequency



(a) (b)

Figure 2.1 Two-dimensional sinusoidal signals: (a) $(f_x, f_y) = (5, 0)$; (b) $(f_x, f_y) = (5, 10)$. The horizontal and vertical units are the width and height of the image, respectively. Therefore, $f_x = 5$ means that there are five cycles along each row.

$s(x, y) = \sin(10\pi x)$ $\quad$ $s(x, y) = \sin(10\pi x + 20\pi y)$

$$f_m = \sqrt{f_x^2 + f_y^2} \cong 11 \text{cycles/unit length}, \theta = \arctan(f_y / f_x) \approx 64^\circ$$

4

2D Sinusoidal

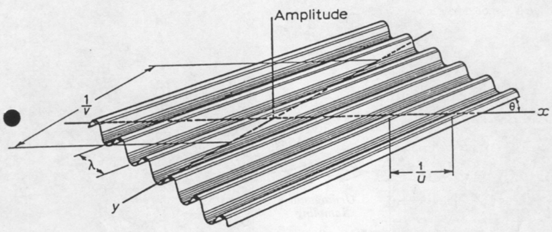
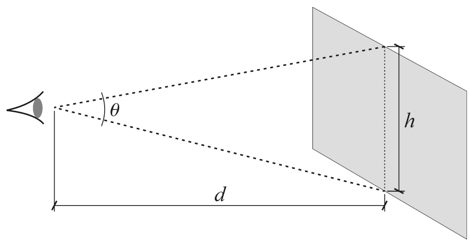


Fig. 2.3 Perspective view of a 2-D cosine wave, $\cos[2\pi(ux+vy)]$

(D.E. Pearson, *Transmission and Display of Pictorial Information*, John Wiley & Sons, 1975, p.5)

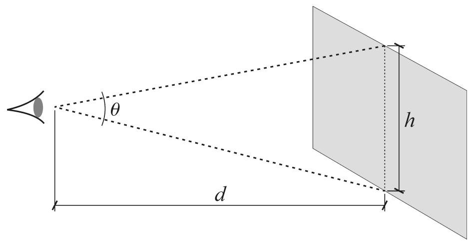
Angular Frequency



- The previous definition does not take into account the viewing distance.
- More useful measure is the angular frequency, expressed in cycles per degree:

$$\theta = 2\arctan\left(\frac{h}{2d}\right) \approx \frac{h}{2d}(radian) = \frac{180h}{\pi d}(\deg.)$$

Angular Frequency



$$f_\theta = \frac{f_s}{\theta} = \frac{\pi d}{180h} f_s (cpd)$$

f_s : cycles per picture height
f_θ : cycles per degree

- For the same picture and picture height (h), angular frequency increases with distance.
- For fixed viewing distance (d), larger displays give less angular frequency.

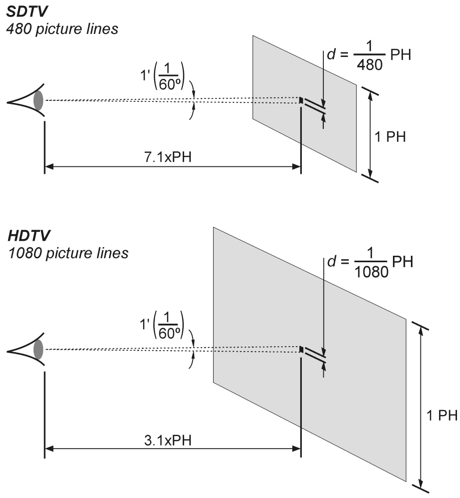
Resolution

- The ability to seperate two adjacent pixels, that is, resolve the details in test grating.
- This ability depends on several factors such as:
  - Picture (monitor) height (h)
  - Viewer's distance from monitor (d)
  - The viewing angle (theta)

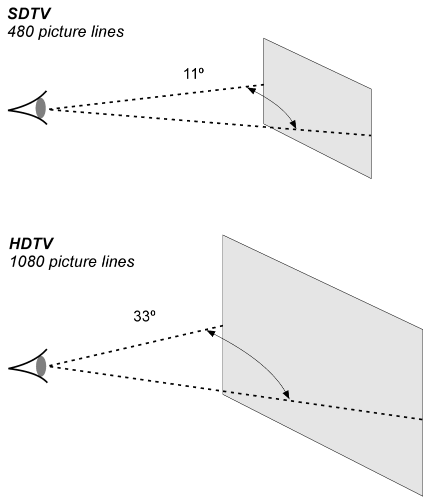
Viewing Distance

20/20 vision
= 1 min of arc
(1/60 degrees)

Optimum viewing distances:
- SDTV = 7.1 x PH (picture height)
- HDTV = 3.1 x PH

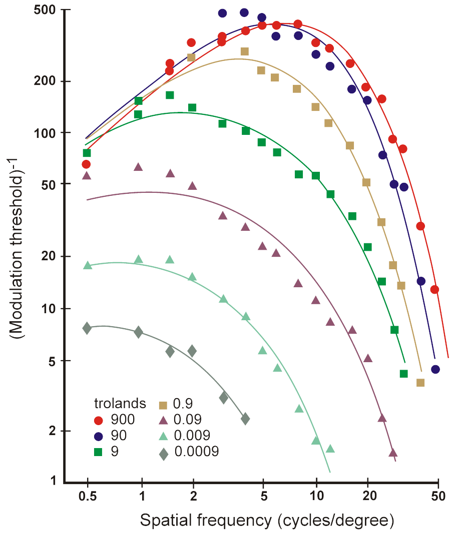


Horizontal Viewing Ranges at Optimum Distances



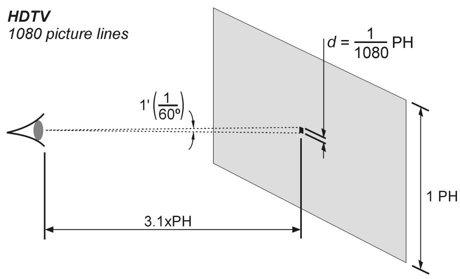
Contrast Sensitivity vs. Spatial Frequency of HVS

- Contrast sensitivity function (CSF) for various retinal illuminance values
- We can not perceive beyond a certain spatial frequency (50cpd).



Spatial Frequency for Peak Contrast Sensitivity

$$f_\theta = \frac{f_s}{\theta} = \frac{\pi(3.1)}{180} f_s = \frac{f_s}{18} (cpd)$$



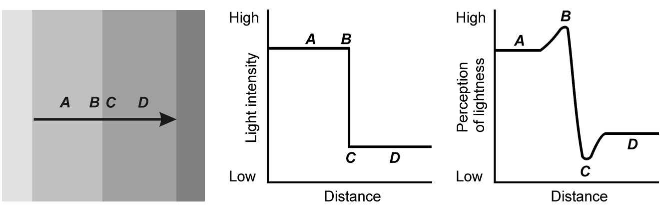
9Td curve peaks at $f_\theta = 4\text{cpd} \rightarrow f_s = 72\text{cpPH} \rightarrow 15$ lines per cycle

Implications and Applications

- The HVS is more sensitive to low spatial frequencies (i.e., luminance changes over a large area) than high spatial frequencies (i.e., rapid changes within small areas), which is an often-exploited aspect of most image compression techniques.
- The HVS is more sensitive to high contrast than low contrast regions within an image, which means that regions with large luminance variations (such as edges) are perceived as particularly important and should therefore be detected, preserved and/or enhanced.
  - *Hence, may discard redundant* ***high spatial frequency*** *content while preserving* ***edges***

Note: Importance of Edges

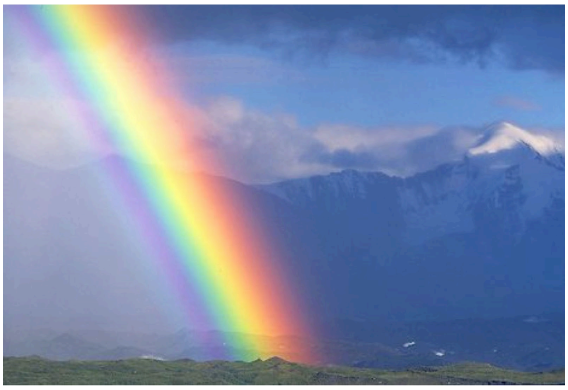
- Our visual system tends to overshoot and undershoot at the boundaries of regions with different intensities (recall Mach bands).
- Explains the ability to seperate objects even in dim light.



(a) (b) (c)

Note: Importance of Edges

Our visual system groups wavelengths of a rainbow to form distinct color bands. It draws artificial lines to separate one color from another.



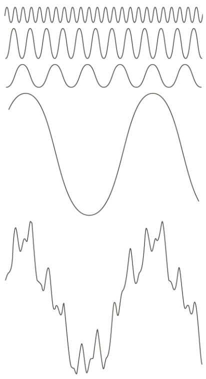
15

Frequency Representation of Images

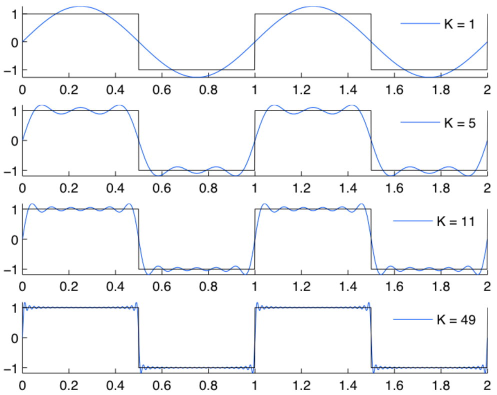
16

Signal Representation Using Sinusoids

All periodic signals can be represented as a sum of sinusoids.



Square Wave Example



2D Sinusoidal Images

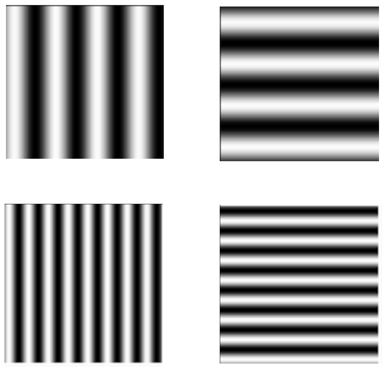


Image Example

original

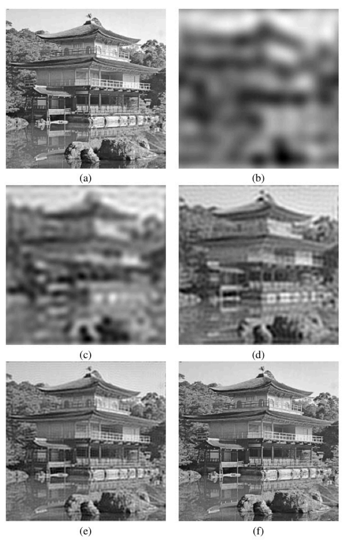
0.3%

1.2%

5%

20%

80%



Assumed Periodicity for Images

Fundamental Period N = image size

Fundamental Frequency $\omega = \frac{2\pi}{N}$

Sinusoidal Frequencies $k\omega,\ k \in \mathrm{Z}$

21

Signal Synthesis with Sinusoidals

$$x(t) = \sum_{k=0}^{\infty} a_k \cos(k\omega t) + \sum_{k=1}^{\infty} b_k \sin(k\omega t)$$

$$\cos\theta = \frac{1}{2}\left(e^{j\theta} + e^{-j\theta}\right) \quad \text{and} \quad \sin\theta = \frac{1}{2j}\left(e^{j\theta} - e^{-j\theta}\right)$$

$$x(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\omega t}$$

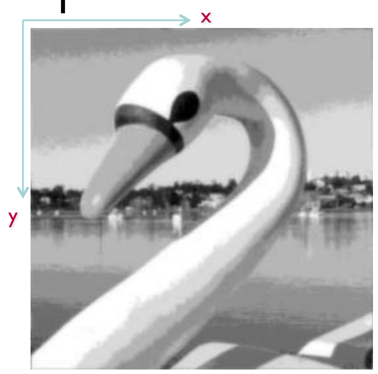
22

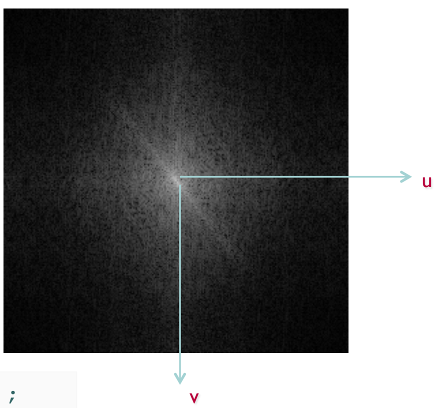
2D Fourier Transform

Discrete Fourier transform (DFT) of $f(x, y)$	$F(u, v) = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)}$ (frequency domain: $F(u,v)$; spatial domain: $f(x,y)$)
Inverse discrete Fourier transform (IDFT) of $F(u, v)$	$f(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M + vy/N)}$
Polar representation	$F(u, v) = \lvert F(u, v) \rvert e^{j\phi(u,v)}$
Spectrum	$\lvert F(u, v) \rvert = \left[R^2(u, v) + I^2(u, v) \right]^{1/2}$ $R = \text{Real}(F); \quad I = \text{Imag}(F)$
Phase angle	$\phi(u, v) = \tan^{-1}\left[\frac{I(u, v)}{R(u, v)} \right]$

23

MATLAB Example

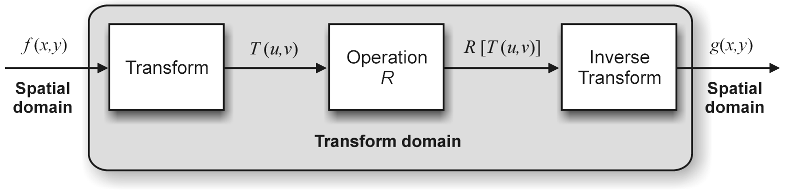





```
I = imread('Figure11_04_a.png');
Id = im2double(I);
ft = fft2(Id);
ft_shift = fftshift(ft);
imshow(log(1 + abs(ft_shift)), [])
```

Transform Domain Processing

- Certain image processing tasks (e.g., filtering, compression) can be better performed in the transform domain.



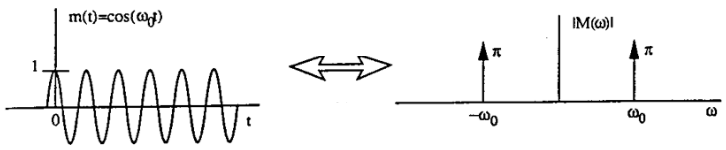
Separability of Fourier Transform

- The Fourier Transform is separable, i.e., the FT of a 2D image can be computed by two passes of the 1D FT algorithm, once along the rows (columns), followed by another pass along the columns (rows) of the result.

Fourier Transform Properties

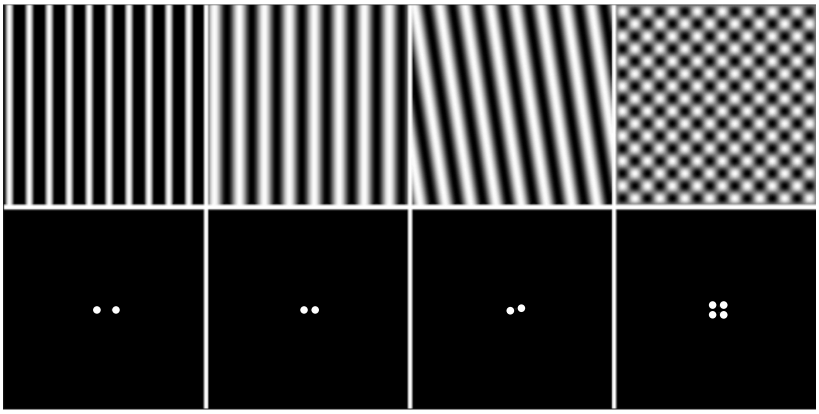
27

Fourier Spectrum of a 1D Sinusoidal

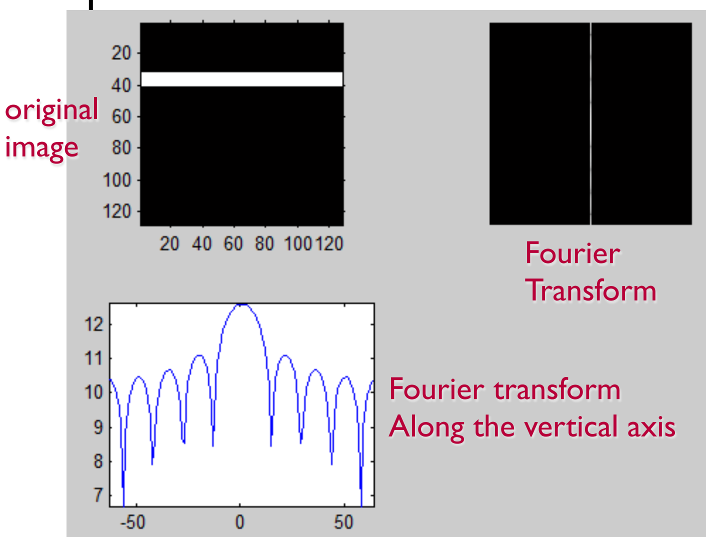


28

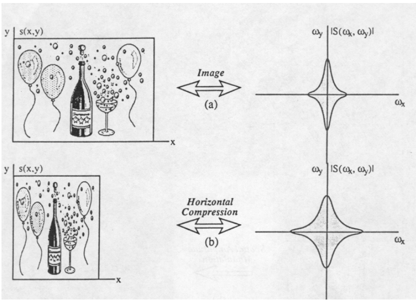
Fourier Transform of a 2D Sinusoidal

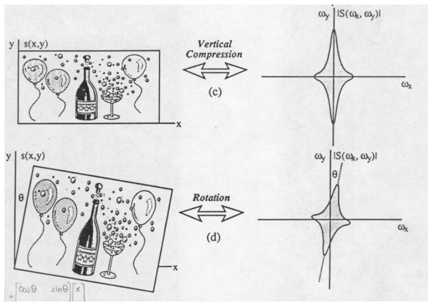


Fourier Transform of a Stripe



Effect of Scaling and Rotation



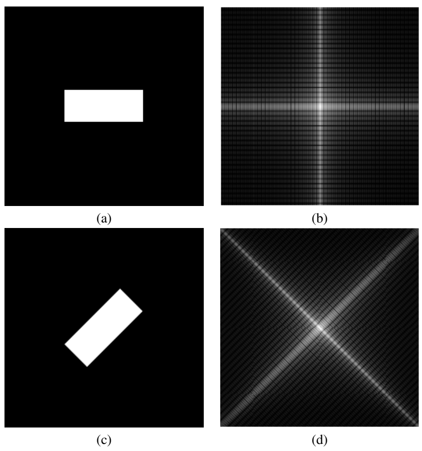


31

Effect of Rotation

- If an image is rotated by a certain angle θ, its 2D FT will be rotated by the same angle.

Image   Fourier Transform



Linearity

$$\mathfrak{F}[a \cdot f1(x,y) + b \cdot f2(x,y)] = a \cdot F1(u,v) + b \cdot F2(u,v)$$

Translation

$$\mathfrak{F}[f(x - x_0, y - y_0)] = F(u,v) \cdot \exp[-j2\pi(ux_0/M + vy_0/N)]$$

Periodicity

$$F(u,v) = F(u+M, v+N)$$

Symmetry

- Conjugate symmetry:

$$F(u,v) = F^*(-u,-v)$$

where:

$F^*(u,v)$ is the conjugate of $F(u,v)$

i.e., if: $F(u,v) = R(u,v) + jI(u,v)$

then:

$$F^*(u,v) = R(u,v) - jI(u,v)$$

$$|F(u,v)| = |F(-u,-v)|$$

More on Symmetry

$$
\begin{aligned}
f(x, y) \text{ real} \quad &\Leftrightarrow \quad F^*(u, v) = F(-u, -v) \\
&\Leftrightarrow \quad R(u, v) \text{ even}; I(u, v) \text{ odd} \\
&\Leftrightarrow \quad |F(u, v)| \text{ even}; \phi(u, v) \text{ odd}
\end{aligned}
$$

37

Average Value

$$
\bar{f}(x, y) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) = \frac{1}{MN} F(0, 0)
$$

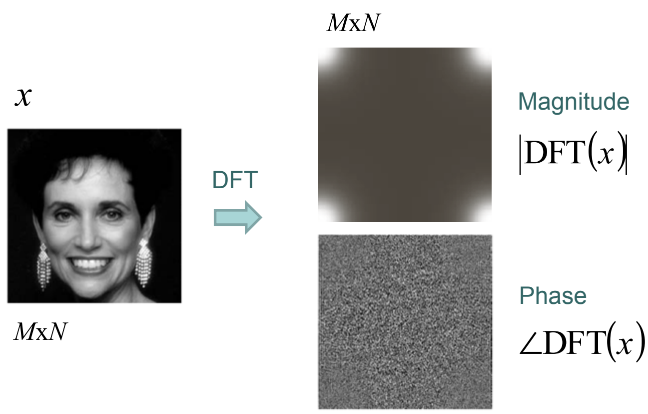
38

Parseval's Relation

$$\sum_x \sum_y |f(x,y)|^2 = \sum_u \sum_v |F(u,v)|^2$$

Magnitude & Phase of FT

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



$$f(x,y)(-1)^{x+y} \Leftrightarrow F(u - M/2, v - N/2)$$

Shifting the FT to the Center of the Rectangle

DFT

$$f(x, y)(-1)^{x+y} \Leftrightarrow F(u - M/2, v - N/2)$$

Multiply the intensity of the pixel at (m,n) with $(-1)^{m+n}$

41

Effect of the Shift in the Image

$$f(x - x_0, y - y_0) \Leftrightarrow F(u, v)e^{-j2\pi(ux_0/M + vy_0/N)}$$

Shifted by (m,n)

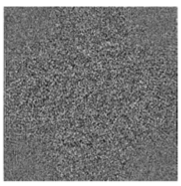
MxN

DFT

MxN

Magnitude not changed

$|\text{DFT}(x)|$



Phase changed

$\angle\text{DFT}(x)$

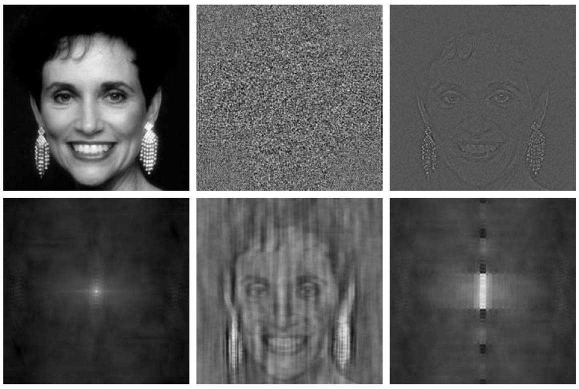
$+2\pi km / M$

$+2\pi ln / N$

42

Fourier Phase & Image Edges

Fig. 4.24(a). (f)

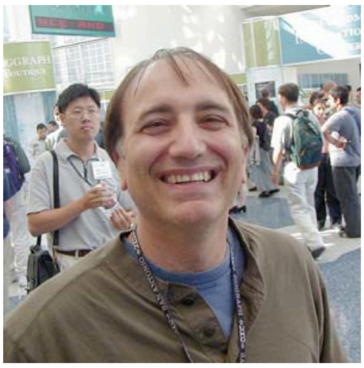


a b c
d e f

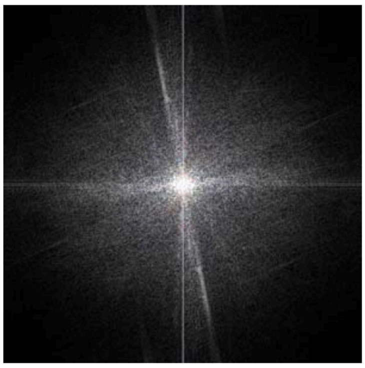
FIGURE 4.27 (a) Woman. (b) Phase angle. (c) Woman reconstructed using only the phase angle. (d) Woman reconstructed using only the spectrum. (e) Reconstruction using the phase angle corresponding to the woman and the spectrum corresponding to the rectangle in Fig. 4.24(a). (f) Reconstruction using the phase of the rectangle and the spectrum of the woman.

43

Boundary Effects



Spatial Domain



Frequency Domain

44

Periodicity Assumption

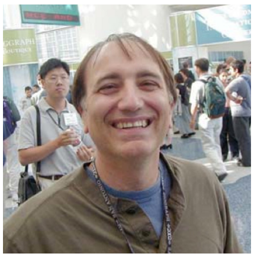
45

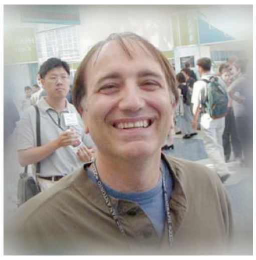
Periodicity Assumption

- Apply mirroring

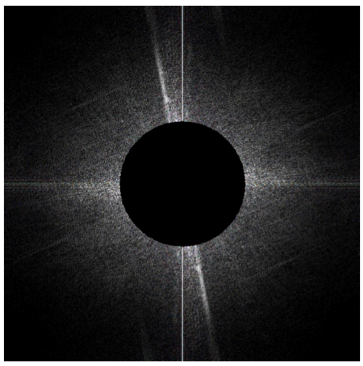
Periodicity Assumption

- Apply edge tapering





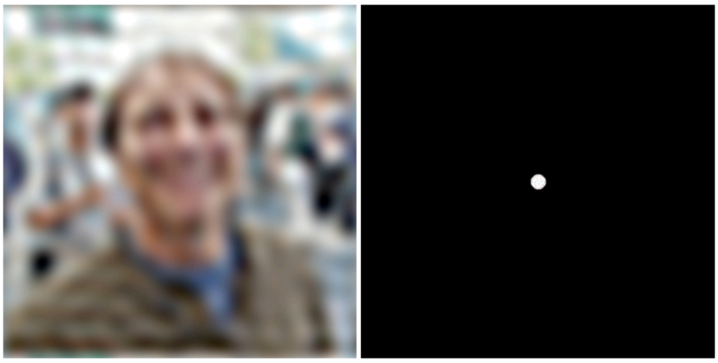
Remove Low Frequencies (Edge Detection)



Spatial Domain   Frequency Domain

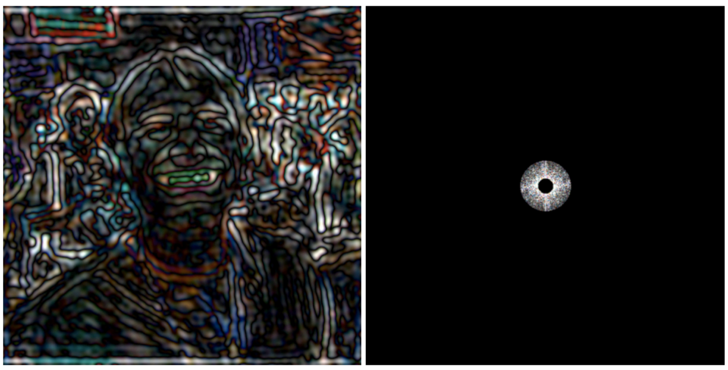
48

Remove High Frequencies (Blurring)



Spatial Domain   Frequency Domain

Remove Low and High Frequencies

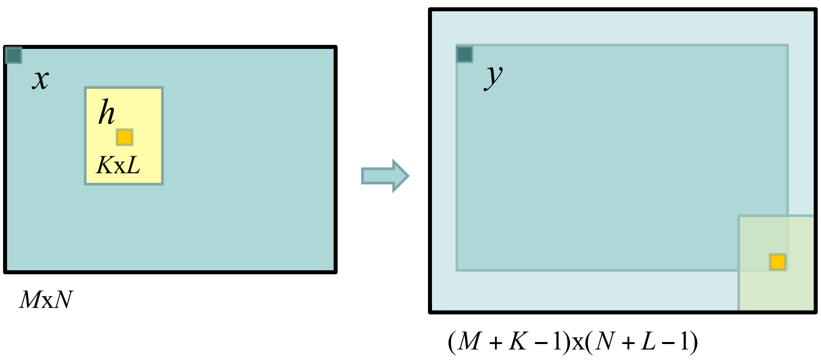


Spatial Domain   Frequency Domain

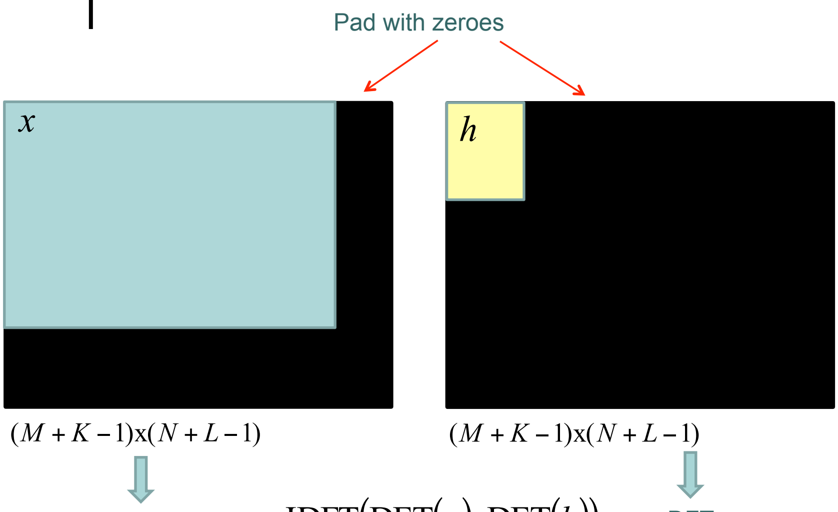
2-D Convolution & DFT

Linear 2-D Convolution

$$y[m,n] = \sum_i \sum_j h[m-i, n-j]x[i,j]$$



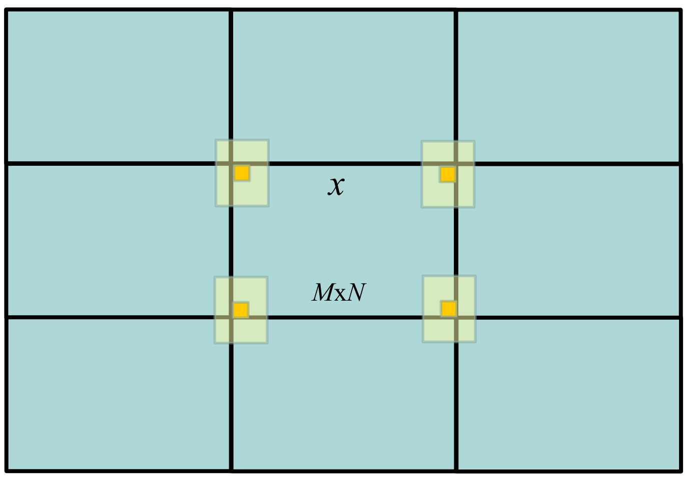
Linear 2-D Convolution via DFT



$$y = \mathrm{IDFT}\left(\mathrm{DFT}(x)\cdot \mathrm{DFT}(h)\right)$$

53

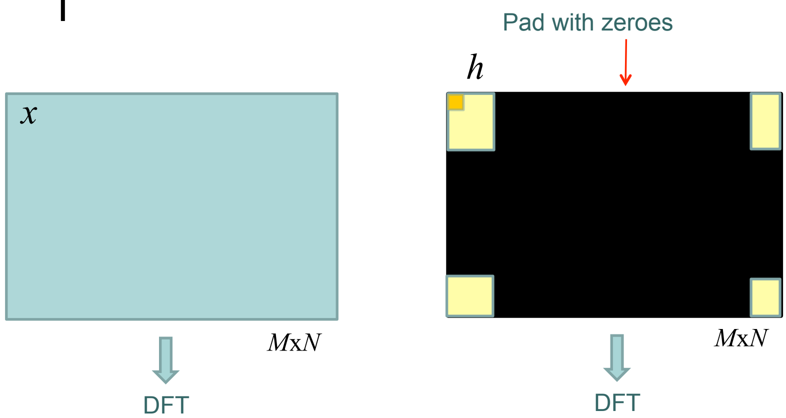
Circular 2-D Convolution



y is also MxN

54

Circular 2-D Convolution via DFT



$$y = \mathrm{IDFT}\left(\mathrm{DFT}(x) \cdot \mathrm{DFT}(h)\right)$$

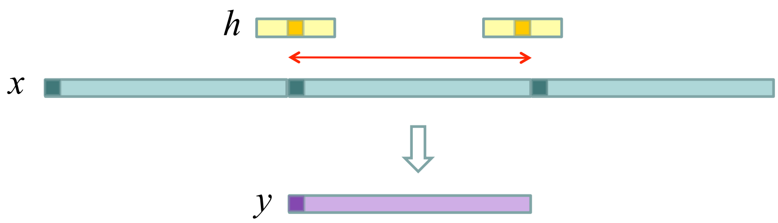
55

1-D Convolution with Matrix Operations

56

1-D Circular Convolution

$$y[m] = h[m] * x[m] = \sum_i h[m-i]x[i]$$



57

1-D Circular Convolution as a Matrix Multiplication

$$y[m] = h[m] * x[m] = \sum_{i=0}^{M-1} h[m-i]x[i]$$

$$Y = HX$$

$$\begin{bmatrix} y_0 \\ y_1 \\ \vdots \\ y_{M-2} \\ y_{M-1} \end{bmatrix} = \begin{bmatrix} h_0 & h_{-1} & \ddots & h_2 & h_1 \\ h_1 & h_0 & h_{-1} & \ddots & h_2 \\ \ddots & h_1 & h_0 & \ddots & \ddots \\ h_{-2} & \ddots & \ddots & \ddots & h_{-1} \\ h_{-1} & h_{-2} & \ddots & h_1 & h_0 \end{bmatrix} \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{M-2} \\ x_{M-1} \end{bmatrix}$$

Circulant matrix

58

1-D DFT as Matrix Multiplication

$$\tilde{x}[k] = \sum_{m=0}^{M-1} W_M^{km} x[m] \qquad W_M^{km} = \frac{1}{\sqrt{M}} e^{-j2\pi km/M} = W_M^{(k\pm M)(m\pm M)}$$

$$\boxed{\tilde{X} = WX}$$

$$\tilde{X} = \begin{bmatrix} \tilde{x}_0 \\ \vdots \\ \tilde{x}_{M-1} \end{bmatrix} \quad W = \begin{bmatrix} W_M^{00} & \cdots & W_M^{0(M-1)} \\ \vdots & \ddots & \vdots \\ W_M^{(M-1)0} & \cdots & W_M^{(M-1)(M-1)} \end{bmatrix} \quad X = \begin{bmatrix} x_0 \\ \vdots \\ x_{M-1} \end{bmatrix}$$

DFT Diagonalizes any Circulant Matrix

$$W = W^T = W^{-1} \Rightarrow$$ Columns and rows of W are orthonormal

$$\left(W^T H W\right)_{kl} = \sum_i \sum_j W_M^{ik} H_{ij} W_M^{jl} = \sum_j \sum_i W_M^{ik} h_{i-j} W_M^{jl}$$

$$= \sum_j \sum_m W_M^{(j+m)k} h_m W_M^{jl} = \sum_m h_m W_M^{mk} \sum_j W_M^{jk} W_M^{jl}$$

$$= \tilde{h}_k \delta_{kl}$$

$$\Rightarrow \quad W^T H W = \operatorname{diag}\left(\tilde{h}_0, \cdots, \tilde{h}_{M-1}\right)$$

Convolution in Space ⟺ Product in DFT

$$\tilde{Y} = WY = WHX = (WHW)(WX)$$
$$= \text{diag}(\tilde{h}_0, \cdots, \tilde{h}_{M-1})\tilde{X}$$

$$\Downarrow$$

$$\begin{bmatrix} \tilde{y}_0 \\ \vdots \\ \tilde{y}_{M-1} \end{bmatrix} = \begin{bmatrix} \tilde{h}_0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \tilde{h}_{M-1} \end{bmatrix} \begin{bmatrix} \tilde{x}_0 \\ \vdots \\ \tilde{x}_{M-1} \end{bmatrix}$$

61

Separable 2-D Filtering

62

Separable 2-D Filter

$$y[m,n] = \sum_i \sum_j h[m-i, n-j]x[i,j]$$

seperable $\Longrightarrow$ $h[m,n] = h_1[m]h_2[n]$

Filtering the Rows

$$\bar{x}[m,j] = \sum_i h_1[m-i]x[i,j]$$

$$\begin{bmatrix} \bar{x}(0,0) & \cdots & \bar{x}(M-1,0) \\ \vdots & \ddots & \vdots \\ \bar{x}(0,N-1) & \cdots & \bar{x}(M-1,N-1) \end{bmatrix}$$

$$= \begin{bmatrix} x(0,0) & \cdots & x(M-1,0) \\ \vdots & \ddots & \vdots \\ x(0,N-1) & \cdots & x(M-1,N-1) \end{bmatrix} \begin{bmatrix} h_1(0) & \cdots & h_1(-1) \\ \vdots & \ddots & \vdots \\ h_1(1) & \cdots & h_1(0) \end{bmatrix}$$

Filtering the Columns

$$y[m,n] = \sum_j h_2[n-j]\bar{x}[m,j]$$

$$\begin{bmatrix} y(0,0) & \cdots & y(M-1,0) \\ \vdots & \ddots & \vdots \\ y(0,N-1) & \cdots & y(M-1,N-1) \end{bmatrix}$$

$$= \begin{bmatrix} h_2(0) & \cdots & h_2(1) \\ \vdots & \ddots & \vdots \\ h_2(-1) & \cdots & h_2(0) \end{bmatrix} \begin{bmatrix} \bar{x}(0,0) & \cdots & \bar{x}(M-1,0) \\ \vdots & \ddots & \vdots \\ \bar{x}(0,N-1) & \cdots & \bar{x}(M-1,N-1) \end{bmatrix}$$

65

Images and Filters as Matrices

$$Y = \begin{bmatrix} y(0,0) & \cdots & y(M-1,0) \\ \vdots & \ddots & \vdots \\ y(0,N-1) & \cdots & y(M-1,N-1) \end{bmatrix}$$

$$X = \begin{bmatrix} x(0,0) & \cdots & x(M-1,0) \\ \vdots & \ddots & \vdots \\ x(0,N-1) & \cdots & x(M-1,N-1) \end{bmatrix}$$

Circulant matrix

Circulant matrix

$$H_1 = \begin{bmatrix} h_1(0) & \cdots & h_1(-1) \\ \vdots & \ddots & \vdots \\ h_1(1) & \cdots & h_1(0) \end{bmatrix}_{NxN} \quad H_2 = \begin{bmatrix} h_2(0) & \cdots & h_2(1) \\ \vdots & \ddots & \vdots \\ h_2(-1) & \cdots & h_2(0) \end{bmatrix}_{MxM}$$

66

Result 1: 2-D Convolution with a Separable Filter as Matrix Multiplication

$$y[m,n] = \sum_i \sum_j h_1[m-i]h_2[n-j]x[i,j]$$

⇓

$$Y = H_2^T X H_1$$

67

Result 2: 2-D DFT as Matrix Multiplication

$$\text{DFT}(x) = \tilde{x}[k,l] = \sum_m \sum_n W_M^{km} W_N^{ln} x[m,n]$$

$$= \sum_n W_N^{ln} \underbrace{\sum_m W_M^{km} x[m,n]}$$ ← rows

$$= \sum_n W_N^{ln} \tilde{x}[k,n]$$ ← columns

$$W_M^{km} = e^{-j2\pi km/M}$$

$$W_N^{ln} = e^{-j2\pi ln/M}$$

⇓

$$\tilde{X} = W_N X W_M$$

68

Result 3: 2-D Convolution with a Separable Filter as a Product of DFTs

$$\tilde{Y} = W_N Y W_M = W_N \left(H_2^T X H_1\right) W_M$$
$$= \left(W_N H_2^T W_N\right)\left(W_N X W_M\right)\left(W_M H_1 W_M\right)$$
$$= \tilde{H}_2^T \tilde{X} \tilde{H}_1$$

H_1 is circulant ⇩ $\tilde{H}_1 = diag(\tilde{h}_{1,0} \cdots \tilde{h}_{1,M-1})$

H_2 is circulant ⇩ $\tilde{H}_2 = diag(\tilde{h}_{2,0} \cdots \tilde{h}_{2,N-1})$

⇩

$$\tilde{Y}_{kl} = \tilde{h}_{1,l} \tilde{h}_{2,k} \tilde{X}_{kl}$$

2-D Convolution with Matrix Operations

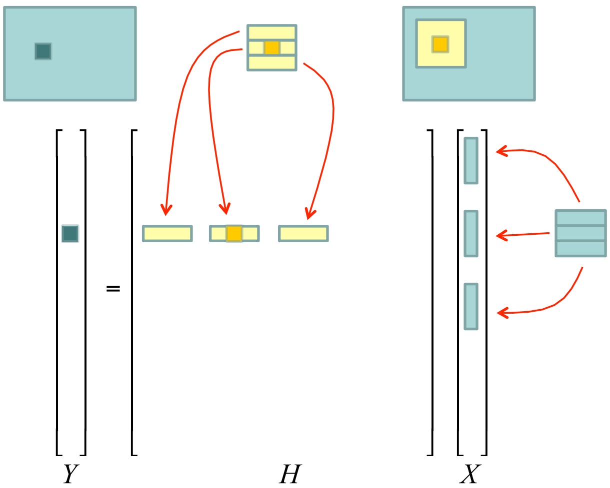
(Non-Separable Filters)

Lexicographical Ordering of Pixels

$$Y = \begin{bmatrix} y(0,0) \\ \vdots \\ y(0,N-1) \\ \vdots \\ y(M-1,0) \\ \vdots \\ y(M-1,N-1) \end{bmatrix} \qquad X = \begin{bmatrix} x(0,0) \\ \vdots \\ x(0,N-1) \\ \vdots \\ x(M-1,0) \\ \vdots \\ x(M-1,N-1) \end{bmatrix}$$

71

Non-Separable Filter as a Matrix Multiplication



72

2-D Filter as a **Block-Circulant** Matrix

First row of filter as a circulant matrix

$$
\left[\begin{array}{ccccc}
\begin{matrix} h_0(0) & \cdots & h_0(1) \\ \vdots & \ddots & \vdots \\ h_0(-1) & \cdots & h_0(0) \end{matrix} &
\begin{matrix} h_{-1}(0) & \cdots & h_{-1}(1) \\ \vdots & \ddots & \vdots \\ h_{-1}(-1) & \cdots & h_{-1}(0) \end{matrix} &
 & \ddots &
\begin{matrix} h_1(0) & \cdots & h_1(1) \\ \vdots & \ddots & \vdots \\ h_1(-1) & \cdots & h_1(0) \end{matrix} \\
\begin{matrix} h_1(0) & \cdots & h_1(1) \\ \vdots & \ddots & \vdots \\ h_1(-1) & \cdots & h_1(0) \end{matrix} &
\begin{matrix} h_0(0) & \cdots & h_0(1) \\ \vdots & \ddots & \vdots \\ h_0(-1) & \cdots & h_0(0) \end{matrix} &
\begin{matrix} h_{-1}(0) & \cdots & h_{-1}(1) \\ \vdots & \ddots & \vdots \\ h_{-1}(-1) & \cdots & h_{-1}(0) \end{matrix} &
 & \ddots \\
\ddots &
\begin{matrix} h_1(0) & \cdots & h_1(1) \\ \vdots & \ddots & \vdots \\ h_1(-1) & \cdots & h_1(0) \end{matrix} &
\begin{matrix} h_0(0) & \cdots & h_0(1) \\ \vdots & \ddots & \vdots \\ h_0(-1) & \cdots & h_0(0) \end{matrix} &
 & \\
 & & & \ddots & \\
\begin{matrix} h_{-1}(0) & \cdots & h_{-1}(1) \\ \vdots & \ddots & \vdots \\ h_{-1}(-1) & \cdots & h_{-1}(0) \end{matrix} &
 & \ddots & &
\begin{matrix} h_0(0) & \cdots & h_0(1) \\ \vdots & \ddots & \vdots \\ h_0(-1) & \cdots & h_0(0) \end{matrix}
\end{array}\right]
$$

2-D Convolution with a Non-Separable Filter as a Product of DFTs

Lexicographical ordering of Y

Block circulant DFT matrix

Block circulant filter matrix

Lexicographical ordering of X

$$
\begin{aligned}
\tilde{Y} &= WY = WHX = (WHW)(WX) \\
&= \operatorname{diag}\left(\operatorname{diag}\left(\tilde{h}_{0,0},\cdots,\tilde{h}_{0,M-1}\right),\cdots,\operatorname{diag}\left(\tilde{h}_{N-1,0},\cdots,\tilde{h}_{N-1,M-1}\right)\right)\tilde{X}
\end{aligned}
$$

Lexicographical ordering of DFT(Y)

Lexicographical ordering of DFT(Y)

Frequency Domain Filtering

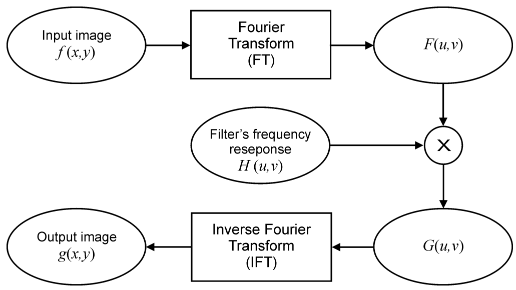
Convolution Theorem

A filter can be implemented in the spatial domain using convolution

A filter can also be implemented in the frequency domain

- **Convert image to frequency domain**
- **Convert filter to frequency domain**
- **Multiply filter times image in frequency domain**
- **Convert result to the spatial domain**

Frequency-Domain Filtering



Mathematical foundation

- Convolution theorem

$$g(x,y) = f(x,y) * h(x,y)$$

FT

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

$$g(x,y) = \mathfrak{F}^{-1}\left[F(u,v)H(u,v)\right]$$

Inverse 2D Fourier Transform (FT)

$h(x,y)$ is a linear, position invariant operator

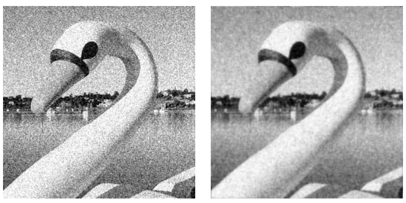
Low-Pass Filtering (LPF)

- Low-pass filters attenuate the high frequency components of an image, while leaving the low frequency components unchanged.
- The typical overall effect of applying a low-pass filter (LPF) to an image is a controlled degree of blurring.

Low-Pass Filtering (LPF)

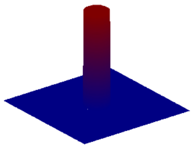
- Example of LPF for smoothing of false contours

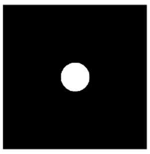
- Example of LPF for noise reduction



Ideal Low-Pass Filtering

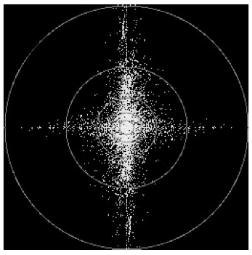
$$H_I(u,v) = \begin{cases} 1 & \text{if } D(u,v) \le D_0 \\ 0 & \text{if } D(u,v) > D_0 \end{cases}$$





$D(u,v) = \sqrt{(u^2 + v^2)}$: distance between a point and origin

D_0 : cutoff frequency (cutoff radius)

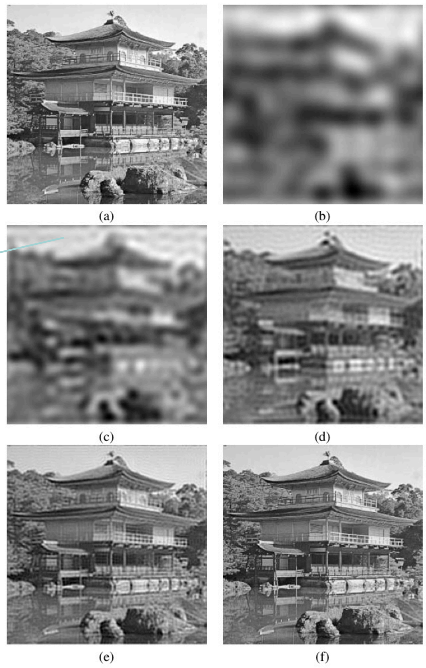


Rings denote s different cutoff frequencies

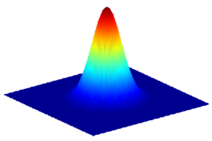
Ideal Low-Pass Filter

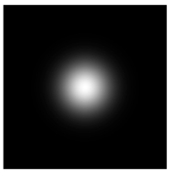
- Ideal LPF example
- Results are for cutoff frequencies:
  - (b) 8 pixels
  - (c) 16 pixels
  - (d) 32 pixels
  - (e) 64 pixels
  - (f) 128 pixels
- There are noticeable ringing artifacts due to the sharp transition between passpand and stopband.

Ringing artifact



Gaussian Low-Pass Filtering

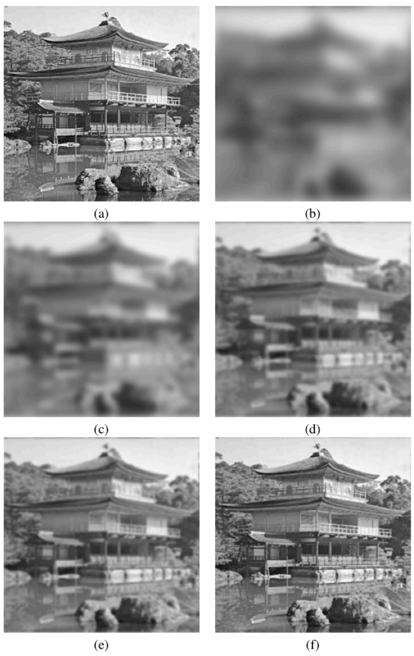




$$H_G(u,v) = e^{-\frac{D(u,v)^2}{2\sigma^2}}$$

- The width of the bell shaped curve is controlled by the parameter sigma, which is equivalent to the cutoff frequency.
- Lower sigma means more strict filtering.
- The smooth transition between passband and stopband guarantees that there will be no noticeable ringing artifacts in the output image.

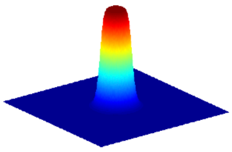
Gaussian Low-Pass Filter



- Gaussian LPF example for various sigma:
  - (b) 75
  - (c) 30
  - (d) 20
  - (e) 10
  - (f) 5

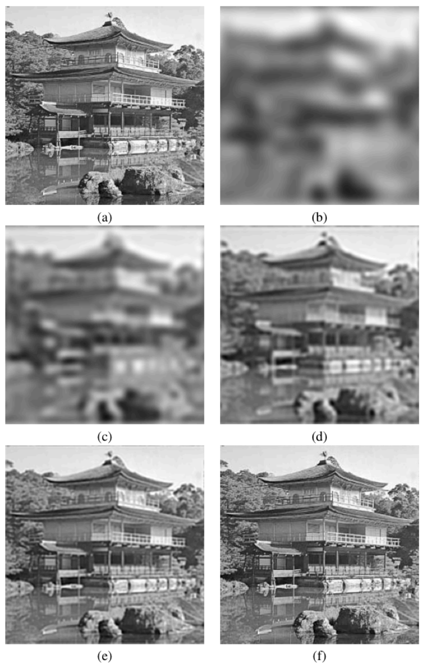
Butterworth Low-Pass Filtering

- Behaviour is a function of the cutoff frequency D_0 and the order of the filter *n*.
- The steepness of the transition between passband and stopband is controlled by *n*.
- Higher n corresponds to steeper transitions.



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

Butterworth Low-Pass Filter

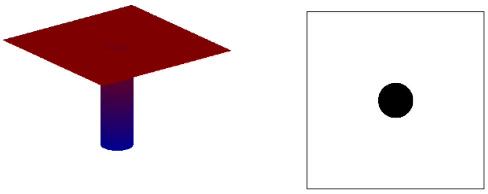


- Butterworth LPF example for n = 4 and various cutoff frequencies:
  - (b) 8 pixels
  - (c) 16 pixels
  - (d) 32 pixels
  - (e) 64 pixels
  - (f) 128 pixels

Ideal High-Pass Filter

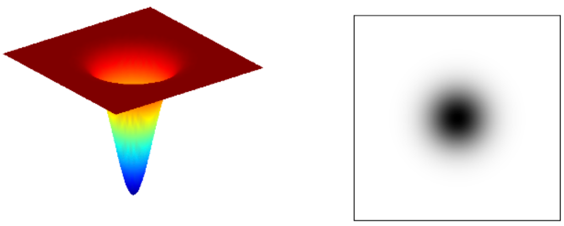
- Ideal HPF attenuates all frequency components within a certain radius, while enhancing others.

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



Gaussian High-Pass Filter

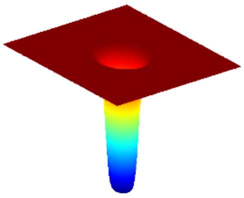
$$H_G(u,v) = 1 - e^{-\frac{D(u,v)^2}{2\sigma^2}}$$

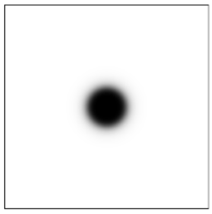


By Oge Marques Copyright © 2011 by John Wiley & Sons, Inc. All rights reserved.

Butterworth High-Pass Filter

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





High-Frequency Emphasis

$$H_{hfe}(u,v) = a + bH(u,v)$$

(a) (b) (c)

(b) Second order Butterworth HPF with cutoff freuency 30 pixels.

(c) High-frequency emphasis with a = 0.5 and b = 1.

Project 1.5

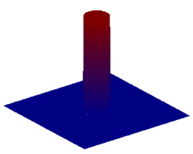
Fourier Transform

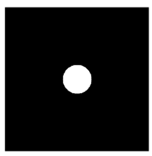
Due 31.10.2013

91

Ideal Low-Pass Filtering

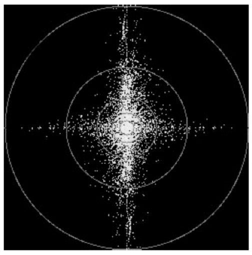
$$H_I(u,v) = \begin{cases} 1 & \text{if } D(u,v) \le D_0 \\ 0 & \text{if } D(u,v) > D_0 \end{cases}$$





$D(u,v) = \sqrt{(u^2 + v^2)}$: distance between a point and origin

D_0 : cutoff frequency (cutoff radius)



Project 1.5

1. Select an arbitrary NxM image. Let N denote the size of the smaller side of the image (usually the vertical side).
2. Find and display the luminance image (Y band) and its Fourier transform (in the logarithm domain).
3. Apply an ideal low pass filter of **circular** shape with diameter N/4 in the Fourier domain. Display the resulting image.
4. Apply an ideal low pass filter of **square** shape with the **same support area** as in Step 3 in the Fourier domain. Display the resulting image.
5. Apply an ideal low pass filter of **diamond** shape with the **same support area** as in Step 3 in the Fourier domain. Display the resulting image.
6. Calculate the RMSE values between the original luminance image and the images obtained in Steps 3, 4, and 5.
7. Compare the images obtained in Steps 3, 4, and 5, and the RMSE values obtained in Step 6 and comment on their differences.

93

Root Mean Squared Error

$$RMSE = \left(\frac{1}{L} \sum_{m} \sum_{n} \left(s_1[m,n] - s_2[m,n] \right)^2 \right)^{1/2}$$

where *L* is the total number of pixels used in the above double summation.

Next Lecture

- SAMPLING