CMPE 685 Computer Vision

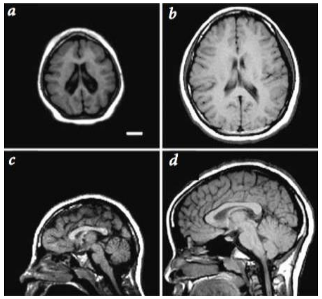
2-D Fourier Transform and Discrete Cosine Transform

Andreas Savakis

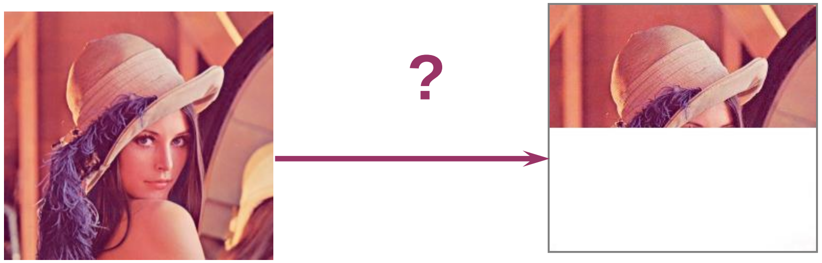
RIT Computer Engineering

Transform Advantages

- Better image processing
  - Conceptual insights using frequency information.
    what it means to be low frequency, high frequency
  - Fast computation: convolution vs. multiplication
- Alternative representation and sensing
  - Obtain transformed data as measurement in radiology images and take inverse transform to recover image



- Efficient storage and transmission
  - Energy compaction
  - Pick a few “representatives” (basis)
  - Just store/send the “contribution” from each basis



The Continuous Fourier Transform

- **Fourier transform: a continuous signal can be represented as a (countable) weighted sum of sinusoids.**

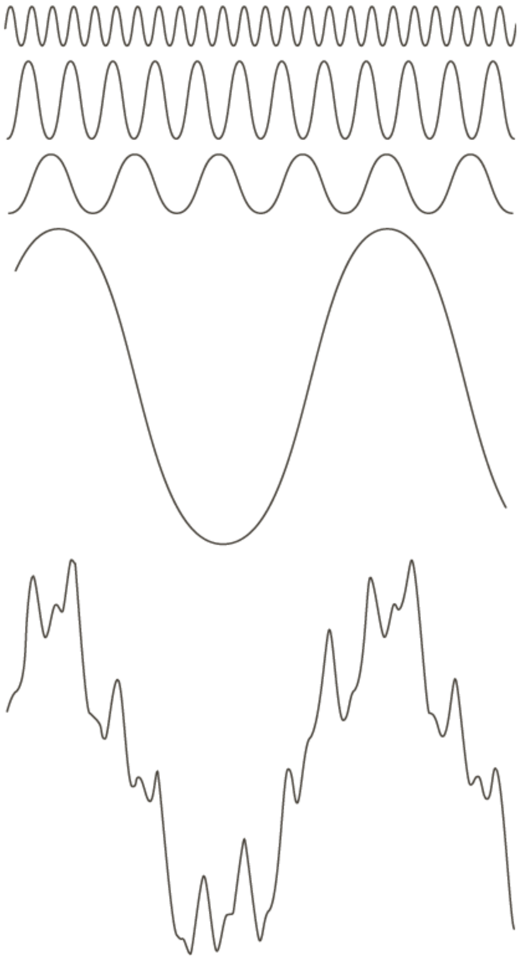


FIGURE 4.1 The function at the bottom is the sum of the four functions above it. Fourier's idea in 1807 that periodic functions could be represented as a weighted sum of sines and cosines was met with skepticism.

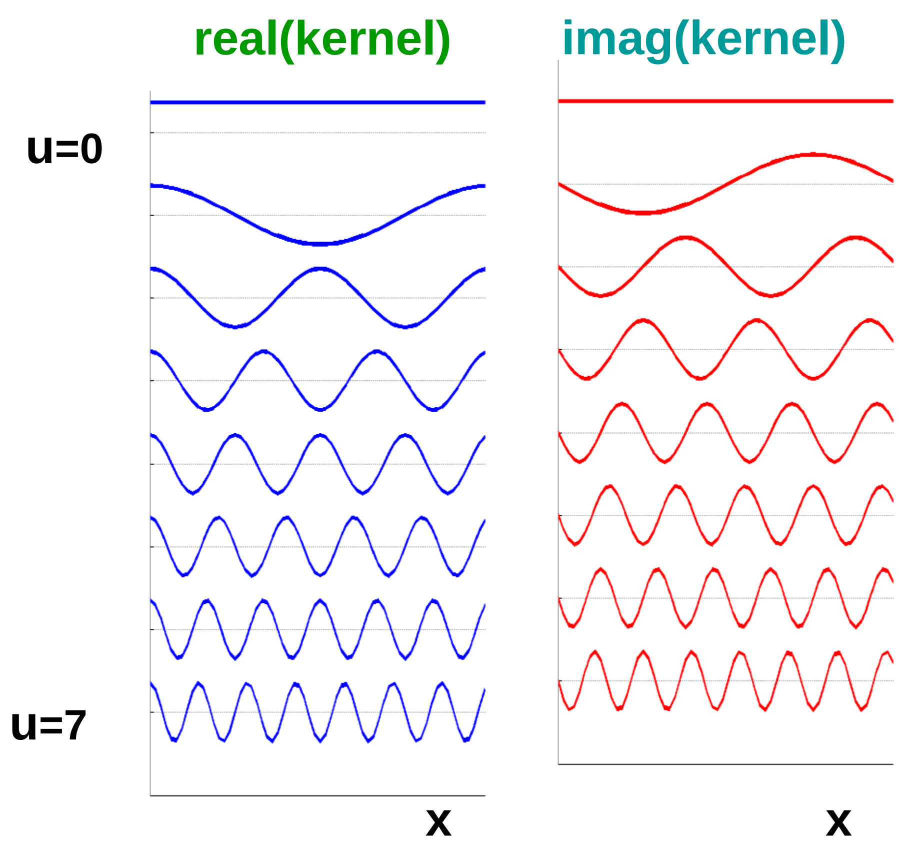
Continuous Fourier Transform

- **1D Continuous Fourier Transform**

$$F(u) = \int_{-\infty}^{\infty} f(x)\, e^{-j2ux} dx$$

- **Transform kernel**

$$e^{-j2ux}$$

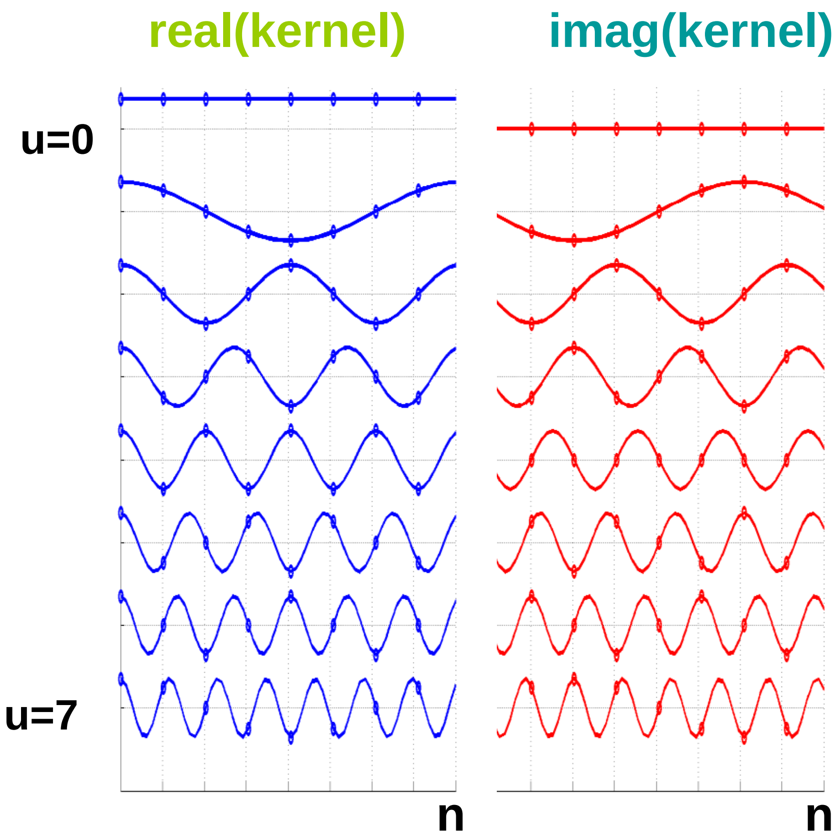


1-D DFT as basis expansion

$$F(u) = \frac{1}{N} \sum_{n=0}^{N-1} f(n) e^{\frac{-j2\pi un}{N}}$$

basis

$$a(u,n) = e^{-j2\pi\frac{un}{N}}$$
$$= cos(2\pi\frac{un}{N}) - jsin(2\pi\frac{un}{N})$$



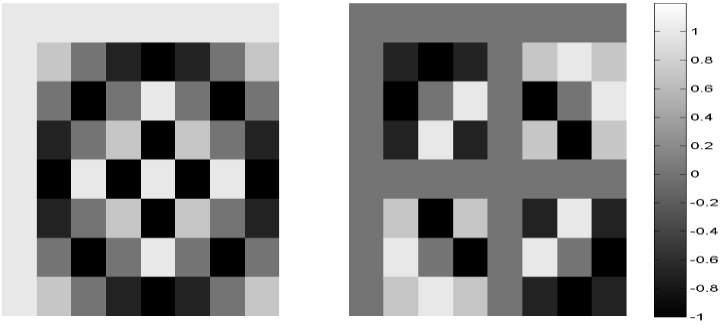
1-D DFT as matrix operation

$$F(u) = \frac{1}{N} \sum_{n=0}^{N-1} f(n) e^{\frac{-j2\pi un}{N}}$$

$$\begin{aligned} a(u,n) &= e^{-j2\pi\frac{un}{N}} \\ &= cos(2\pi\frac{un}{N}) - jsin(2\pi\frac{un}{N}) \end{aligned}$$

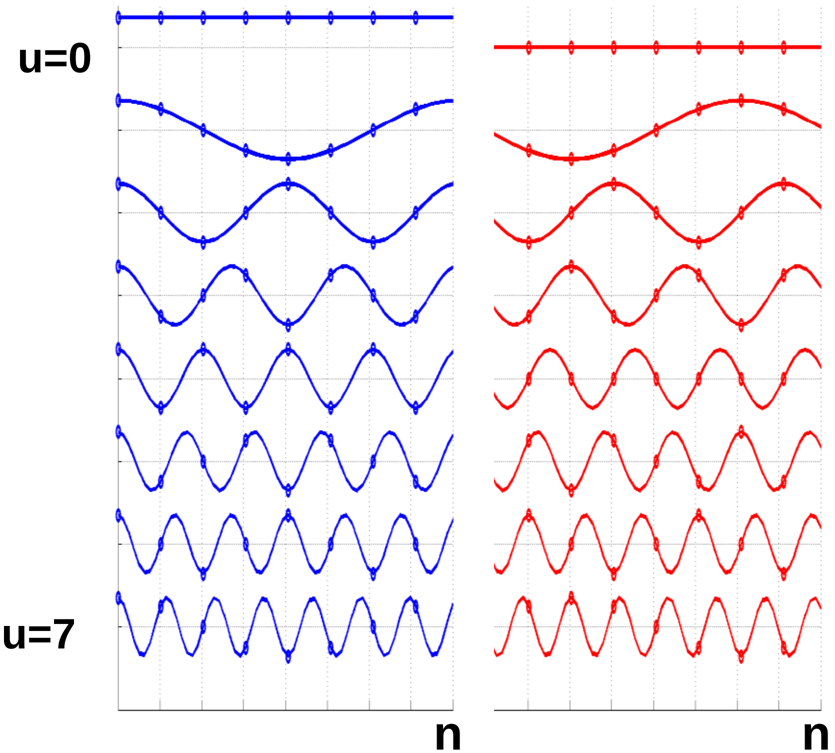
$$u = 0, 1, \ldots, N-1$$

$$\begin{aligned} y &= Ax \\ x &= A^{-1}y \end{aligned}$$



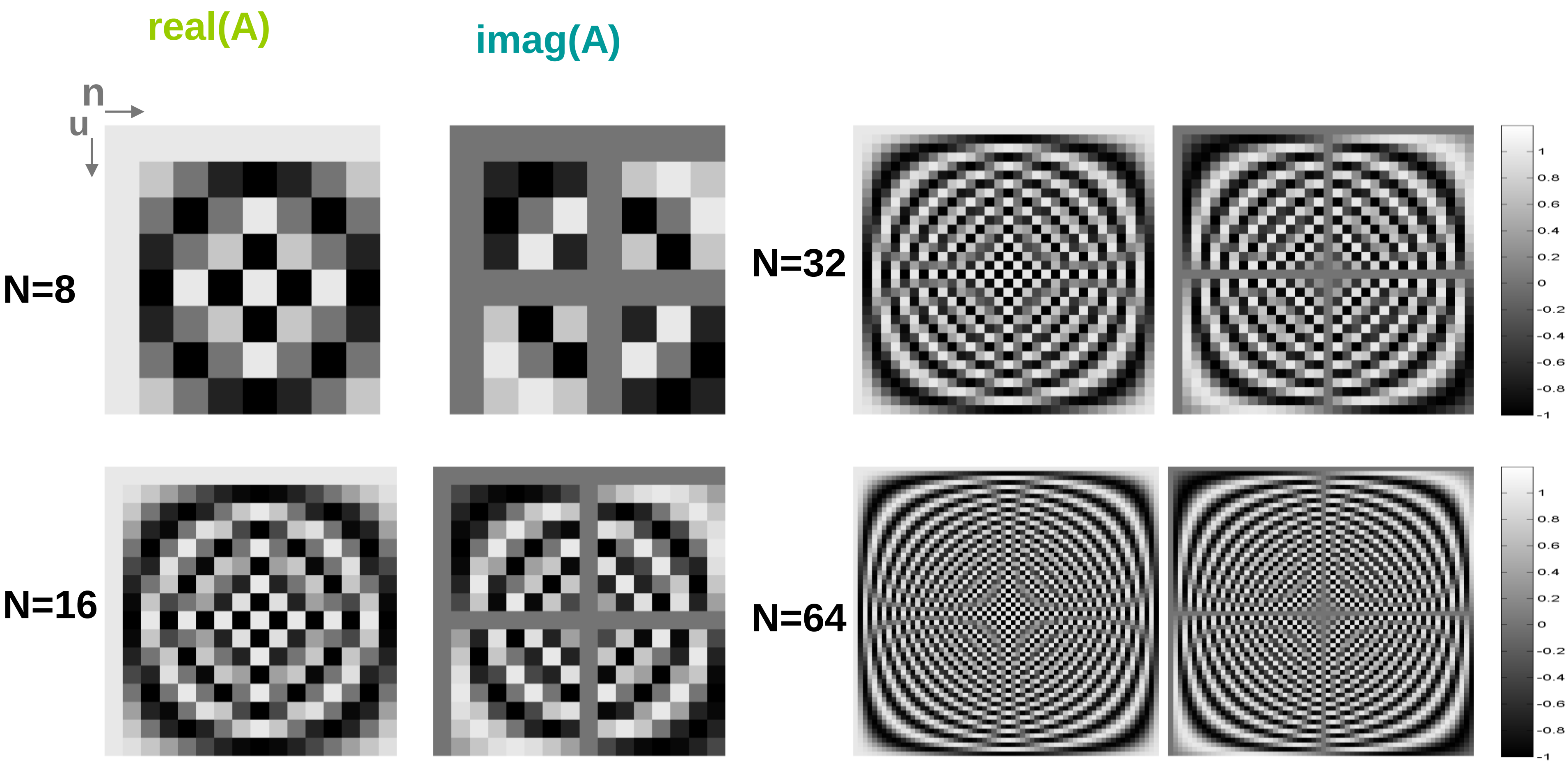
real(A) imag(A)

N=8



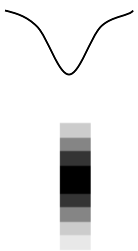
1-D DFT of different lengths

$$
\begin{aligned}
y &= Ax & a(u,n) &= e^{-j2\pi\frac{un}{N}} & u = 0, 1, \ldots, N-1 \\
x &= A^{-1}y & &= cos(2\pi\frac{un}{N}) - jsin(2\pi\frac{un}{N})
\end{aligned}
$$



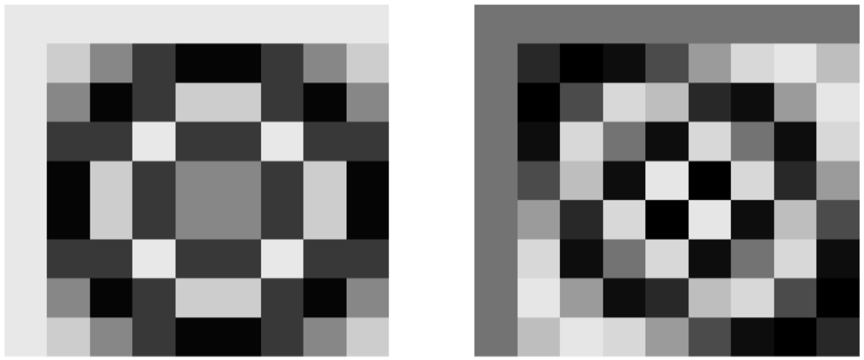
performing 1D DFT

$x(n)$



real-valued input

$$a(u,n) = e^{-j2\pi\frac{un}{N}}$$



$y(u)$

$$y \;=\; Ax$$

$$real(y) = real(A) * x \qquad imag(y) = imag(A) * x$$

Note: the coefficients in x and y on this slide are only meant for illustration purposes, which are not numerically ac

The 2D-DFT

DFT

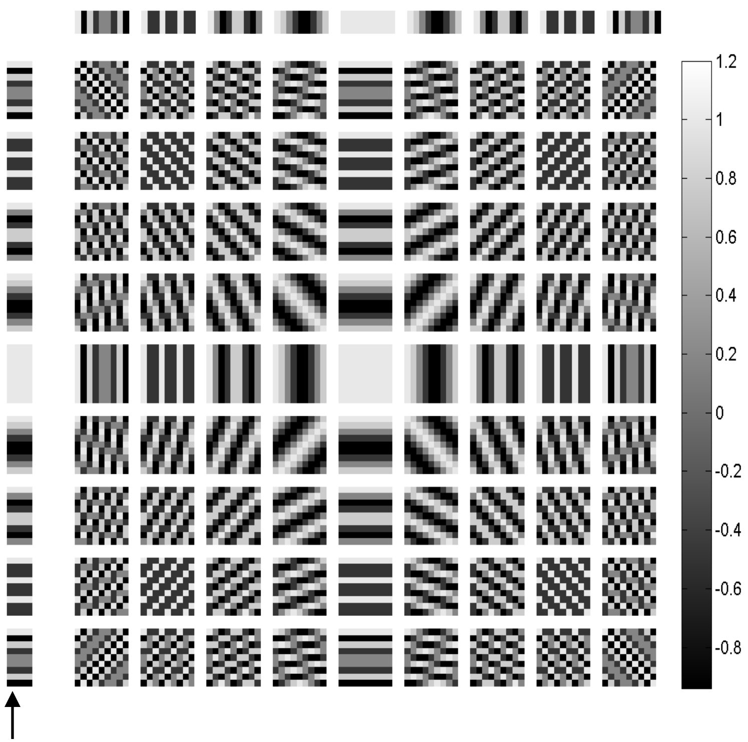
$$y(u,v) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} x(m,n) e^{\frac{-j2\pi um}{M}} e^{\frac{-j2\pi vn}{N}}$$

IDFT

$$x(m,n) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} y(u,v) e^{\frac{j2\pi um}{M}} e^{\frac{j2\pi vn}{N}}$$

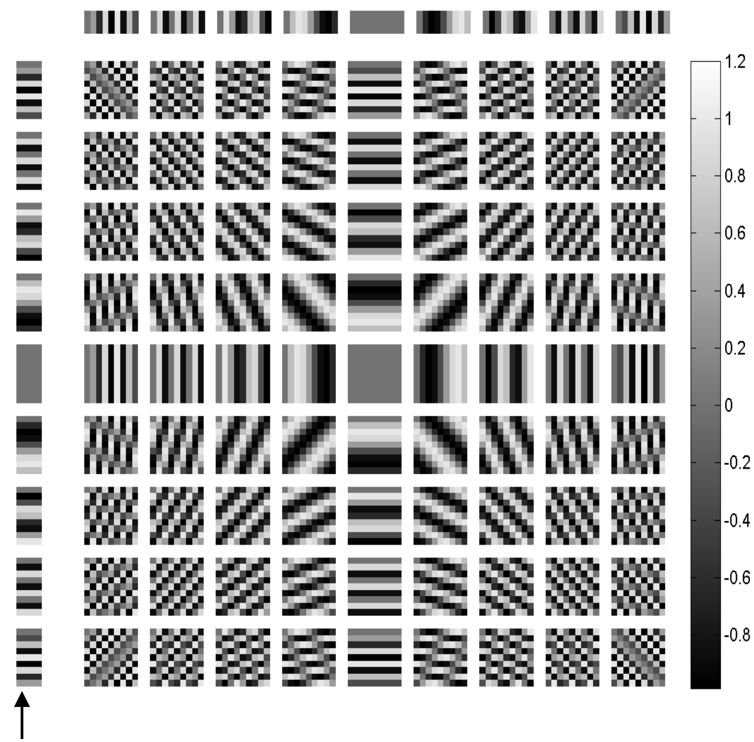
2-D Fourier basis

real $e^{-j2\pi\left(\frac{um}{N}+\frac{vn}{N}\right)}$



real($e^{-j2\pi\frac{um}{M}}$ **)**

imag $e^{-j2\pi\left(\frac{um}{N}+\frac{vn}{N}\right)}$



imag($e^{-j2\pi\frac{um}{M}}$ **)**

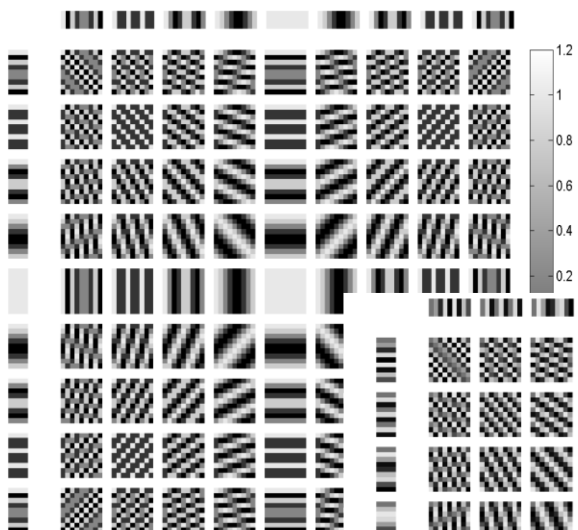
2-D FT illustrated

real-valued

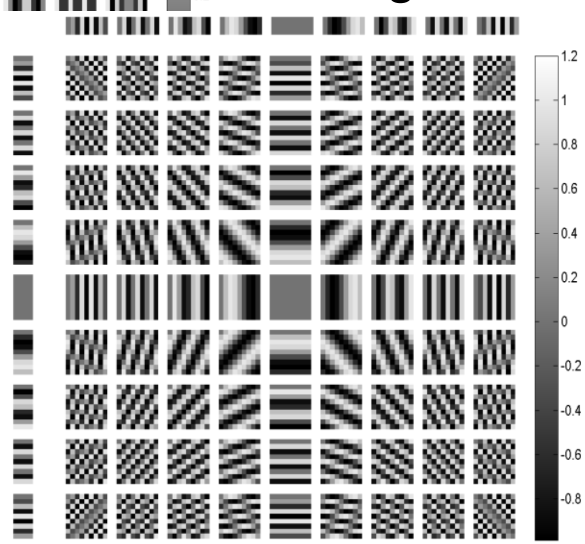
$x(m,n)$

$$a(u,v,m,n) = e^{-j2\pi\left(\frac{um}{N}+\frac{vn}{N}\right)}$$

real

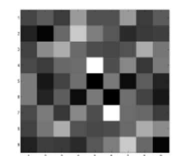


imag

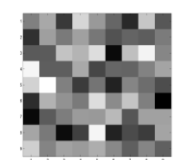


$y(u,v)$

$real(y(u,v))$



$imag(y(u,v))$



Notes about 2D-DFT

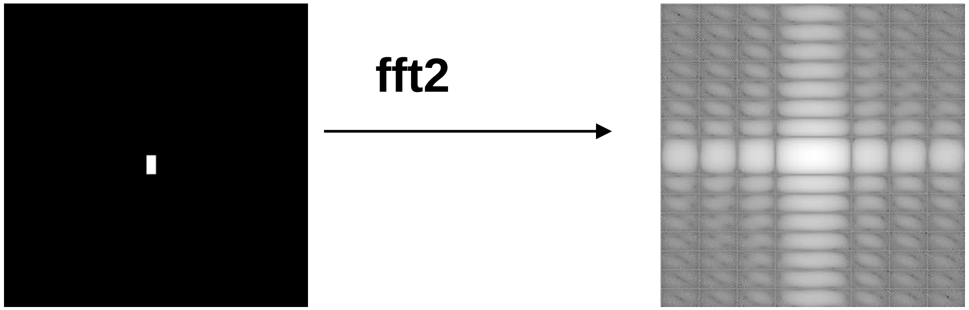
- **Output of the Fourier transform is a complex number**
  - **Decompose the complex number as the magnitude and phase components**
- **`Fourier Transform Pairs`**

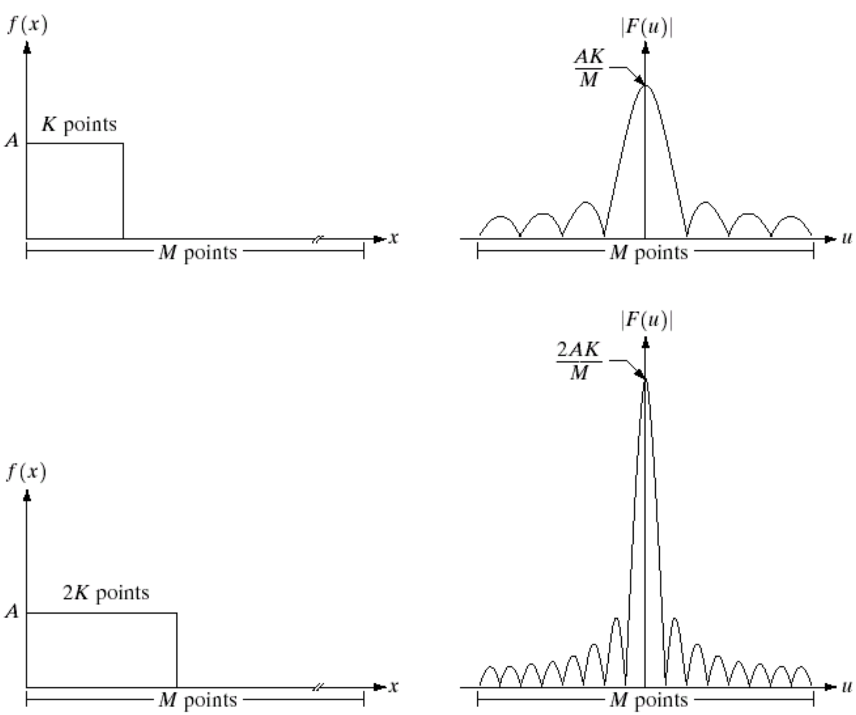
Some useful FT pairs:

Impulse	$\delta(x, y) \Leftrightarrow 1$
Gaussian	$A\sqrt{2\pi}\sigma e^{-2\pi^2\sigma^2(x^2+y^2)} \Leftrightarrow Ae^{-(u^2+v^2)/2\sigma^2}$
Rectangle	$\mathrm{rect}[a, b] \Leftrightarrow ab\frac{\sin(\pi ua)}{(\pi ua)}\frac{\sin(\pi vb)}{(\pi vb)}e^{-j\pi(ua+vb)}$
Cosine	$\cos(2\pi u_0 x + 2\pi v_0 y) \Leftrightarrow \frac{1}{2}\left[\delta(u + u_0, v + v_0) + \delta(u - u_0, v - v_0)\right]$
Sine	$\sin(2\pi u_0 x + 2\pi v_0 y) \Leftrightarrow j\frac{1}{2}\left[\delta(u + u_0, v + v_0) - \delta(u - u_0, v - v_0)\right]$

† Assumes that functions have been extended by zero padding.

2D-DFT Example



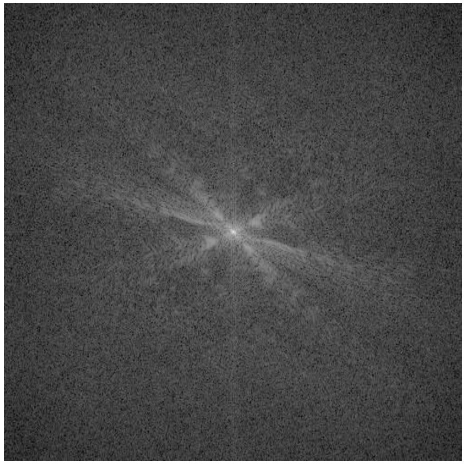
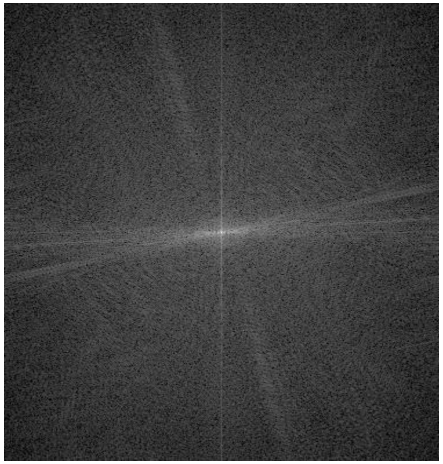


a b
c d

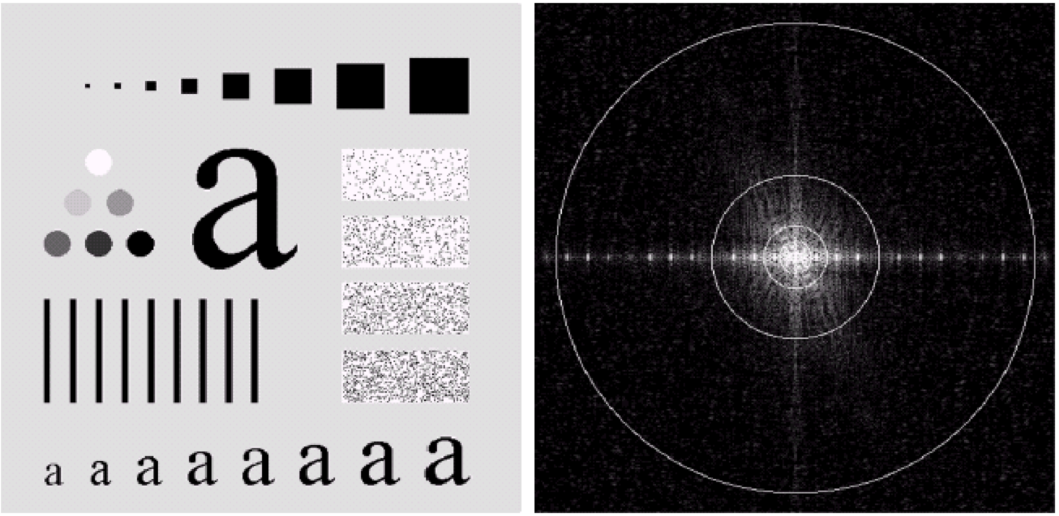
FIGURE 4.2 (a) A discrete function of M points, and (b) its Fourier spectrum. (c) A discrete function with twice the number of nonzero points, and (d) its Fourier spectrum.

DFT Examples



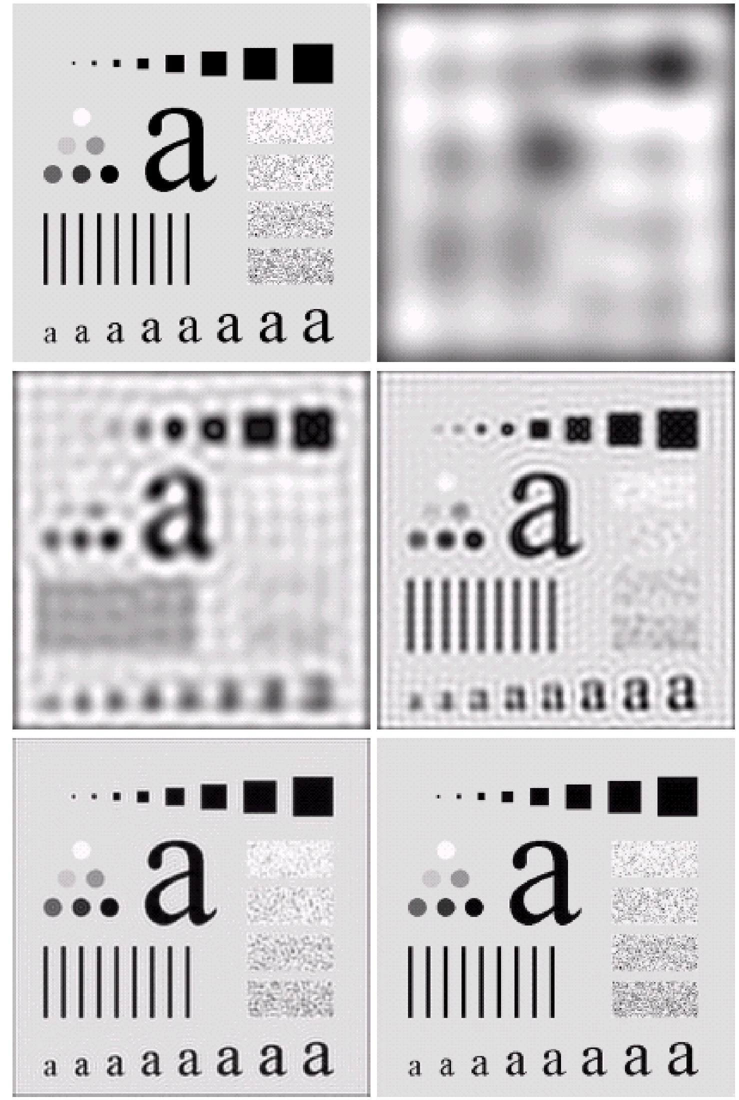



observation 1: compacting energy



a b

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



a b
c d
e f

FIGURE 4.12 (a) Original image. (b)–(f) Results of ideal lowpass filtering with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). The power removed by these filters was 8, 5.4, 3.6, 2, and 0.5% of the total, respectively.

observation 2: amplitude vs. phase

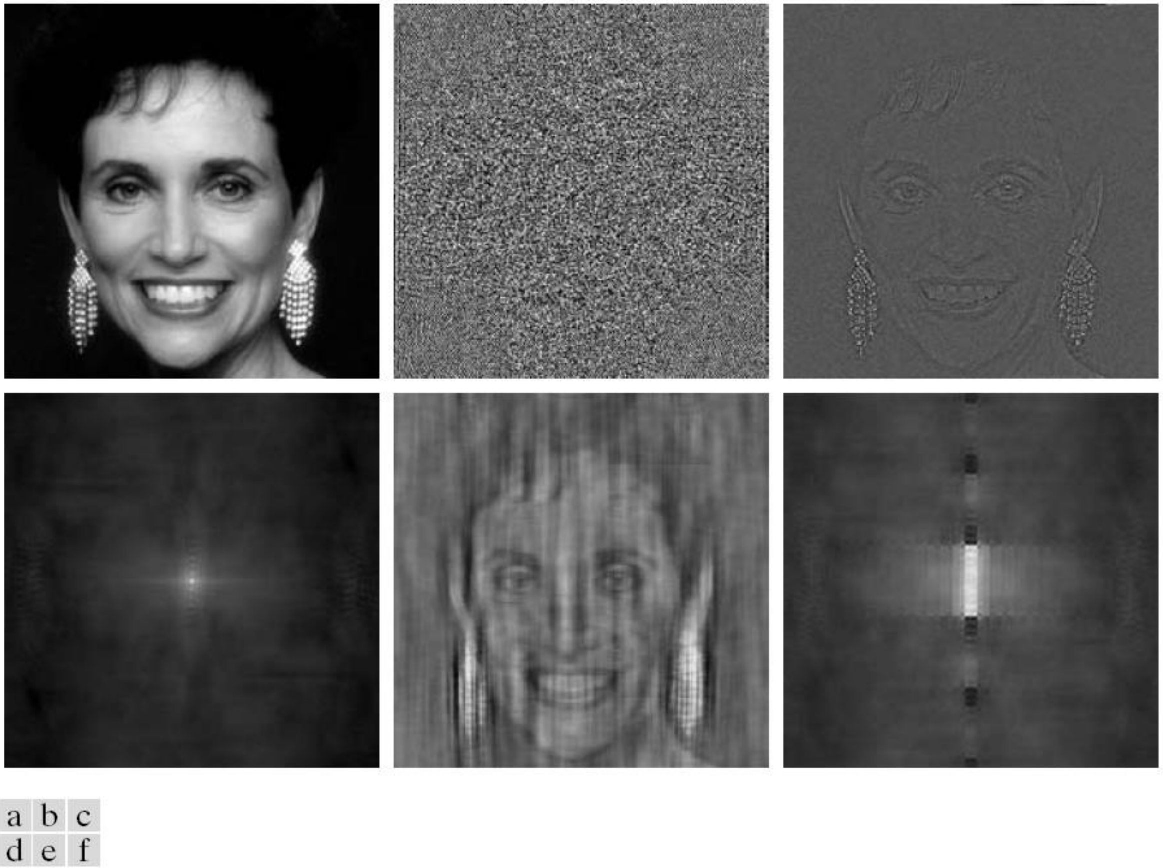


FIGURE 4.27 (a) Woman. (b) Phase angle. (c) Woman reconstructed using only the phase angle. (d) Woman reconstructed using only the spectrum. (e) Reconstruction using the phase angle corresponding to the woman and the spectrum corresponding to the rectangle in Fig. 4.24(a). (f) Reconstruction using the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

- Amplitude: relative prominence of sinusoids
- Phase: relative displacement of sinusoids

another example: amplitude vs. phase

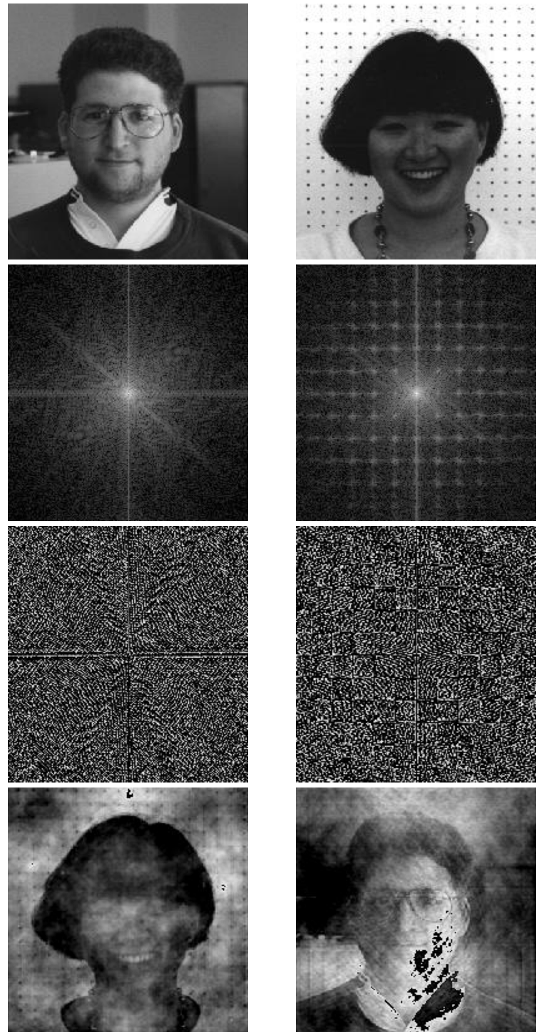
A = “Aron”

FA = fft2(A)

log(abs(FA))

angle(FA)

ifft2(abs(FA),
angle(FP))



P = “Phyllis”

FP = fft2(P)

log(abs(FP))

angle(FP)

ifft2(abs(FP),
angle(FA))

Adpated from http://robotics.eecs.berkeley.edu/~sastry/ee20/vision2/vision2.html

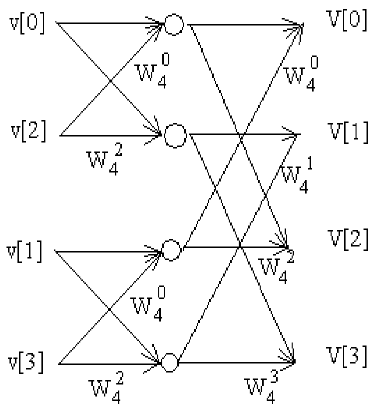
fast implementation of 2-D DFT

- 2 Dimensional DFT is separable

$$
\begin{aligned}
F(u,v) &= \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(m,n)e^{\frac{-2\pi jum}{M}}e^{\frac{-2\pi jvn}{N}} \\
&= \frac{1}{M}\sum_{x=0}^{M-1} e^{\frac{-2\pi jum}{M}} \cdot \frac{1}{N}\sum_{y=0}^{N-1} f(m,n)e^{\frac{-2\pi jvn}{N}} \\
&= \frac{1}{M}\sum_{x=0}^{M-1} e^{\frac{-2\pi jum}{M}} F(m,v)
\end{aligned}
$$

1-D DFT of f(m,n) w.r.t n

1-D DFT of F(m,v) w.r.t m



- **1D FFT: O(N¢log$_2$N)**
- **2D DFT naïve implementation: O(N^4)**
- **2D DFT as 1D FFT for each row and then for each column**

Implement IDFT as DFT

DFT2 $$F(u,v) = \frac{1}{MN}\sum_{m=0}^{M-1}\sum_{n=0}^{N-1} f(m,n)e^{-j2\pi(\frac{um}{M}+\frac{vn}{N})}$$

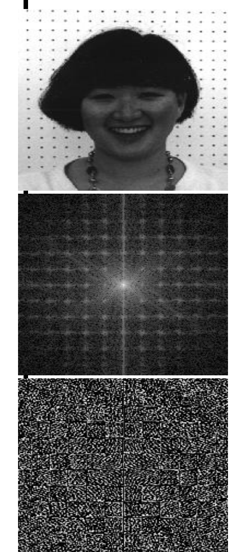
IDFT2 $$f(m,n) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F(u,v)e^{j2\pi(\frac{um}{M}+\frac{vn}{N})}$$

$$\Rightarrow \quad f^*(m,n) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} F^*(u,v)e^{-j2\pi(\frac{um}{M}+\frac{vn}{N})}$$

$$= (MN)\cdot DFT2[F^*(u,v)]$$

Properties of 2D-DFT

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	$F(u, v) = \lvert F(u, v)\rvert e^{-j\phi(u,v)}$
Spectrum	$\lvert F(u, v)\rvert = \left[R^2(u, v) + I^2(u, v)\right]^{1/2}$, $R = \text{Real}(F)$ and $I = \text{Imag}(F)$
Phase angle	$\phi(u, v) = \tan^{-1}\left[\frac{I(u, v)}{R(u, v)}\right]$
Power spectrum	$P(u, v) = \lvert F(u, v)\rvert^2$
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	$f(x, y)e^{j2\pi(u_0x/M + v_0y/N)} \Leftrightarrow F(u - u_0, v - v_0)$ $f(x - x_0, y - y_0) \Leftrightarrow F(u, v)e^{-j2\pi(ux_0/M + vy_0/N)}$ When $x_0 = u_0 = M/2$ and $y_0 = v_0 = N/2$, then $f(x, y)(-1)^{x+y} \Leftrightarrow F(u - M/2, v - N/2)$ $f(x - M/2, y - N/2) \Leftrightarrow F(u, v)(-1)^{u+v}$

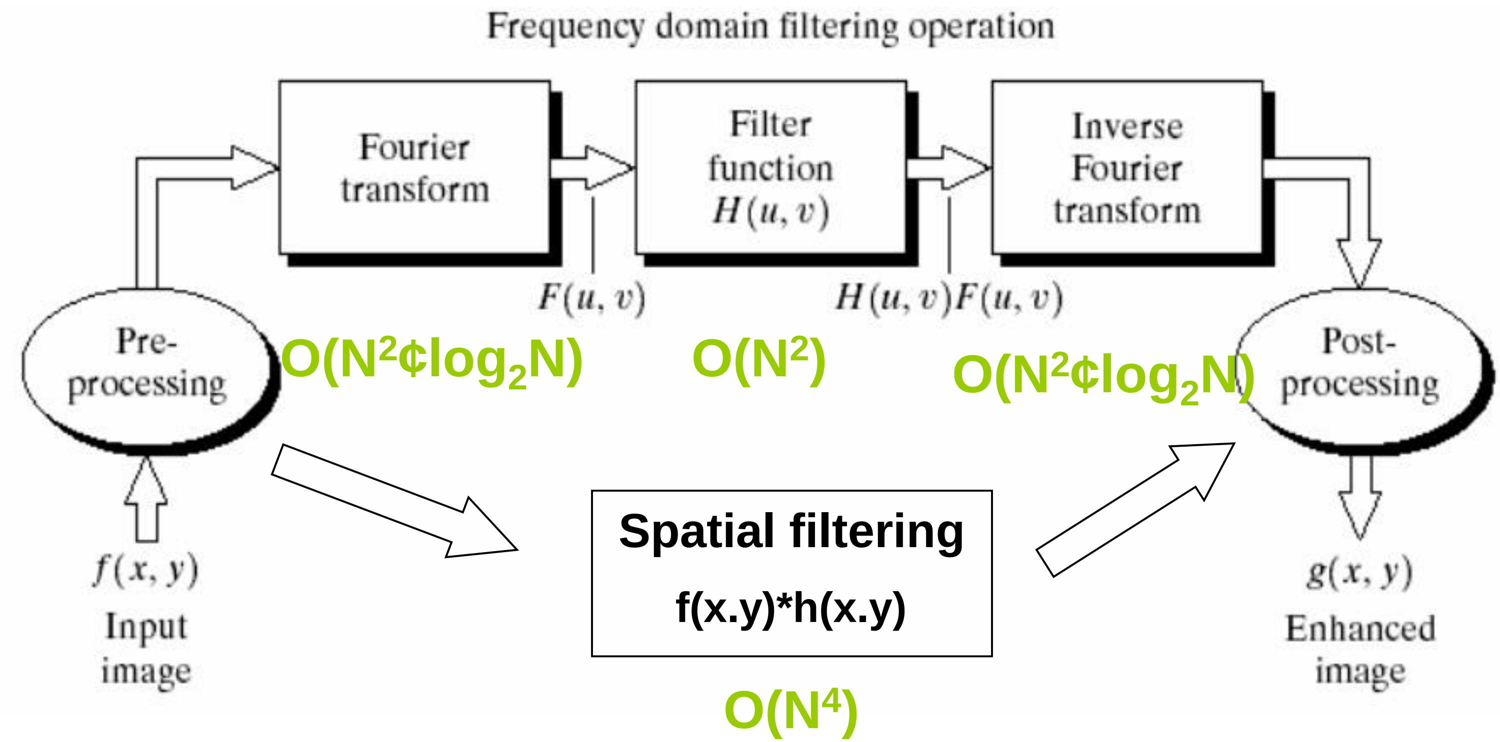


DFT Properties

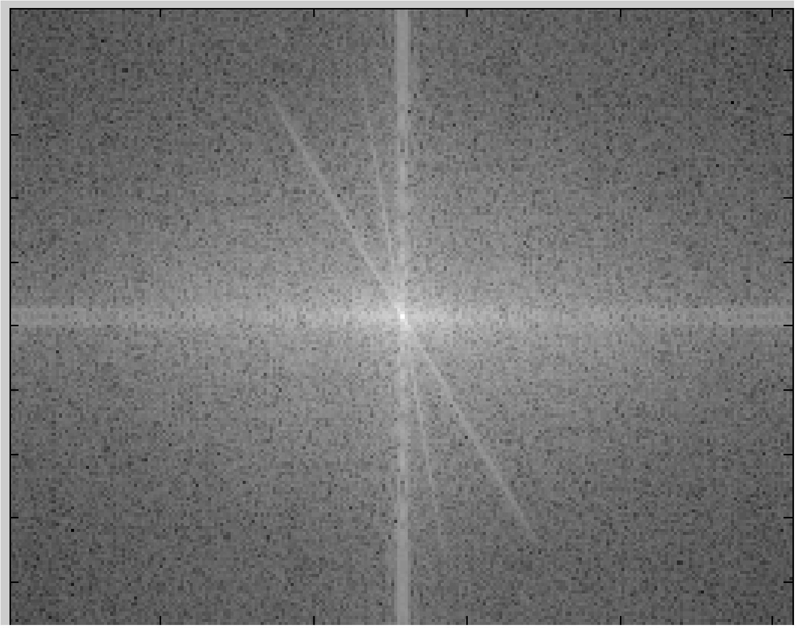
Property	Expression(s)
Computation of the inverse Fourier transform using a forward transform algorithm	$\frac{1}{MN} f^*(x, y) = \frac{1}{MN} \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F^*(u, v) e^{-j2\pi(ux/M + vy/N)}$ This equation indicates that inputting the function $F^*(u, v)$ into an algorithm designed to compute the forward transform (right side of the preceding equation) yields $f^*(x, y)/MN$. 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†	$f(x, y) \circ h(x, y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f^*(m, n) h(x + m, y + n)$
Convolution theorem†	$f(x, y) * h(x, y) \Leftrightarrow F(u, v) H(u, v)$; $f(x, y) h(x, y) \Leftrightarrow F(u, v) * H(u, v)$
Correlation theorem†	$f(x, y) \circ h(x, y) \Leftrightarrow F^*(u, v) H(u, v)$; $f^*(x, y) h(x, y) \Leftrightarrow F(u, v) \circ H(u, v)$

duality result

DFT for fast convolution

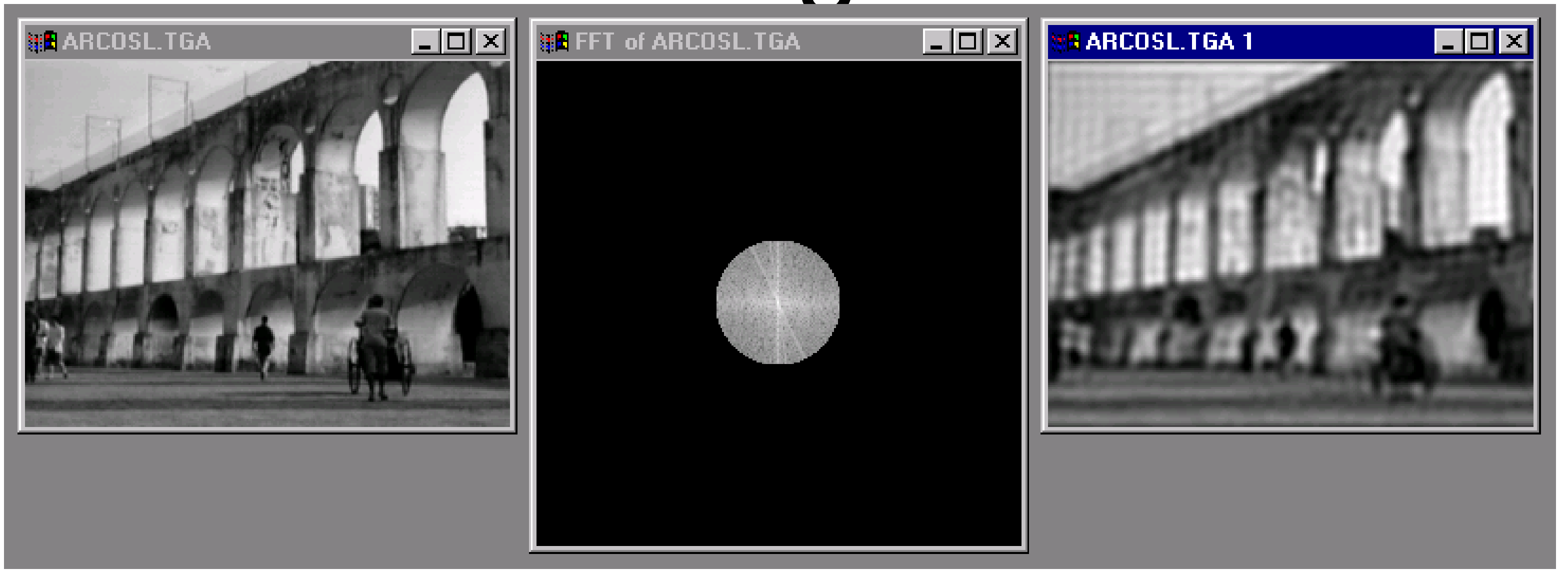


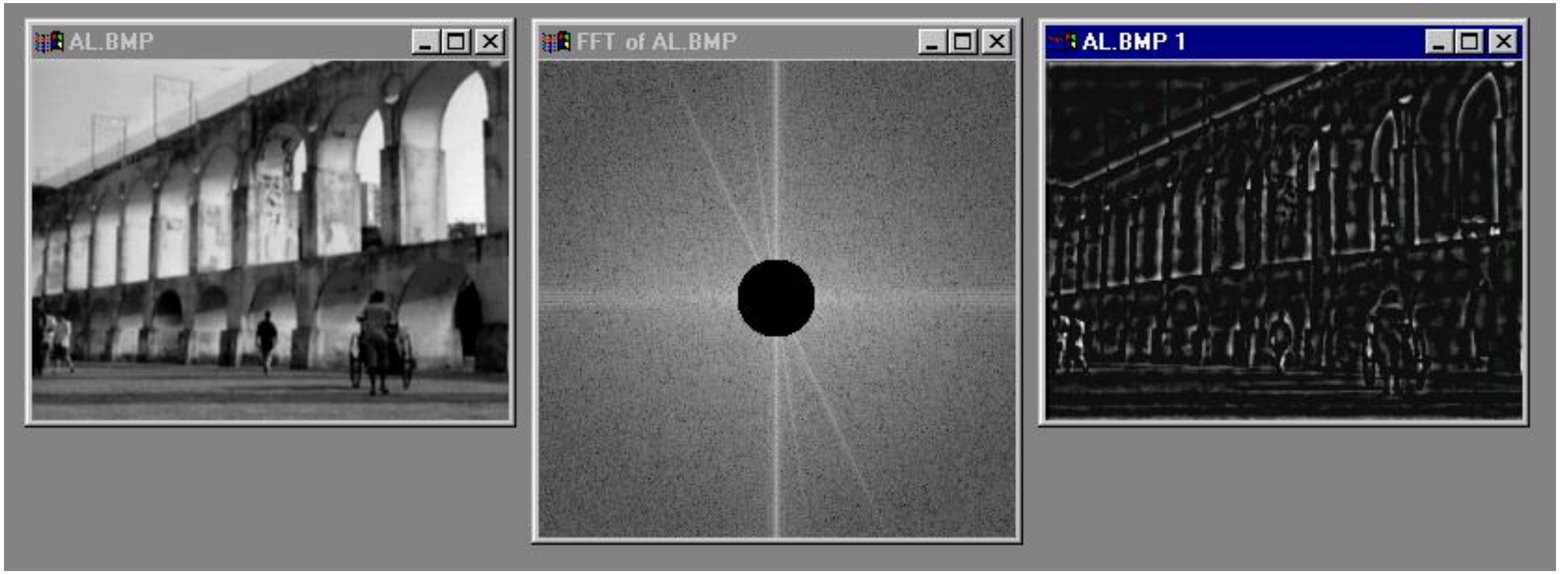
DFT Magnitude example



Can change DFT, then reconstruct

DFT Low and High Pass filtering





smoothing filters: ideal low-pass

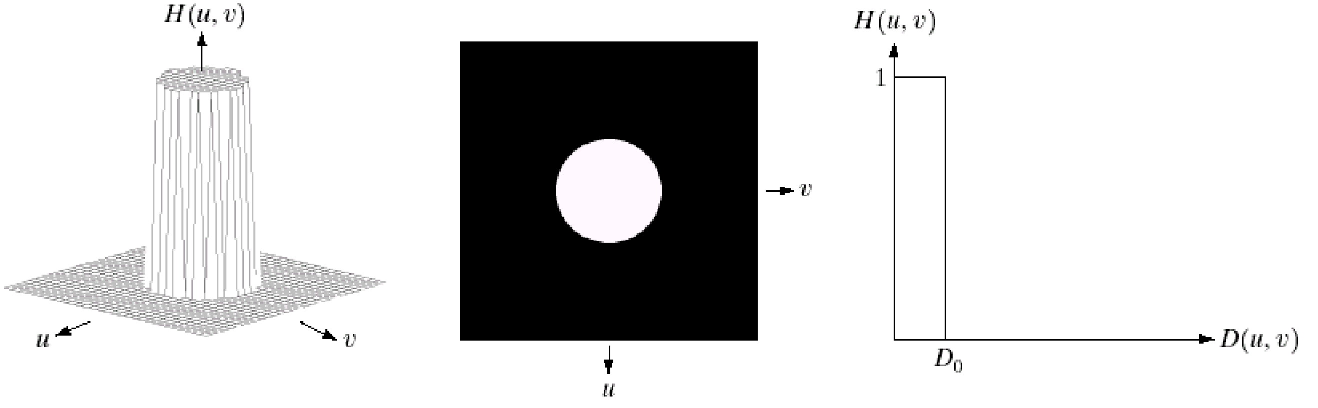
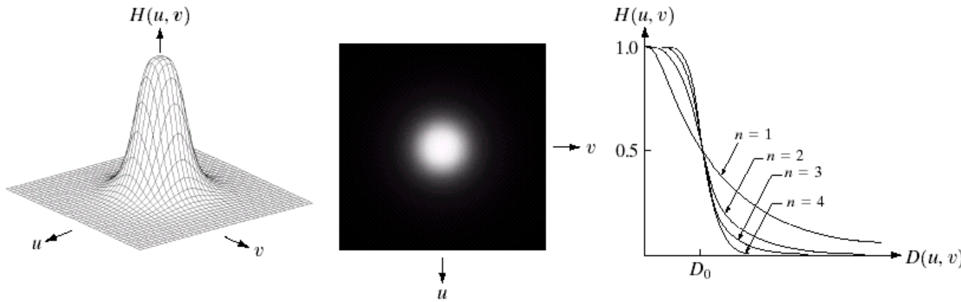


FIGURE 4.10 (a) Perspective plot of an ideal lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

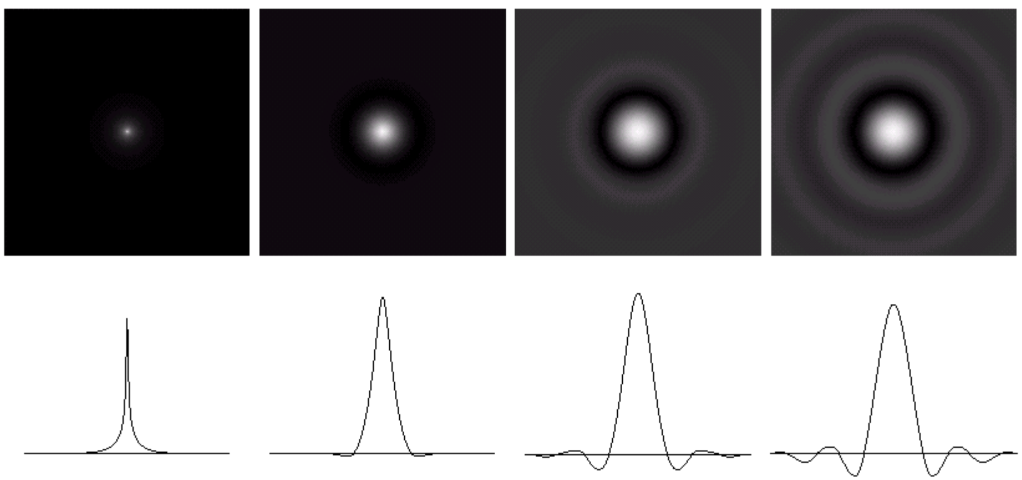
butterworth filters

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



a b c

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



a b c d

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

Gaussian filters

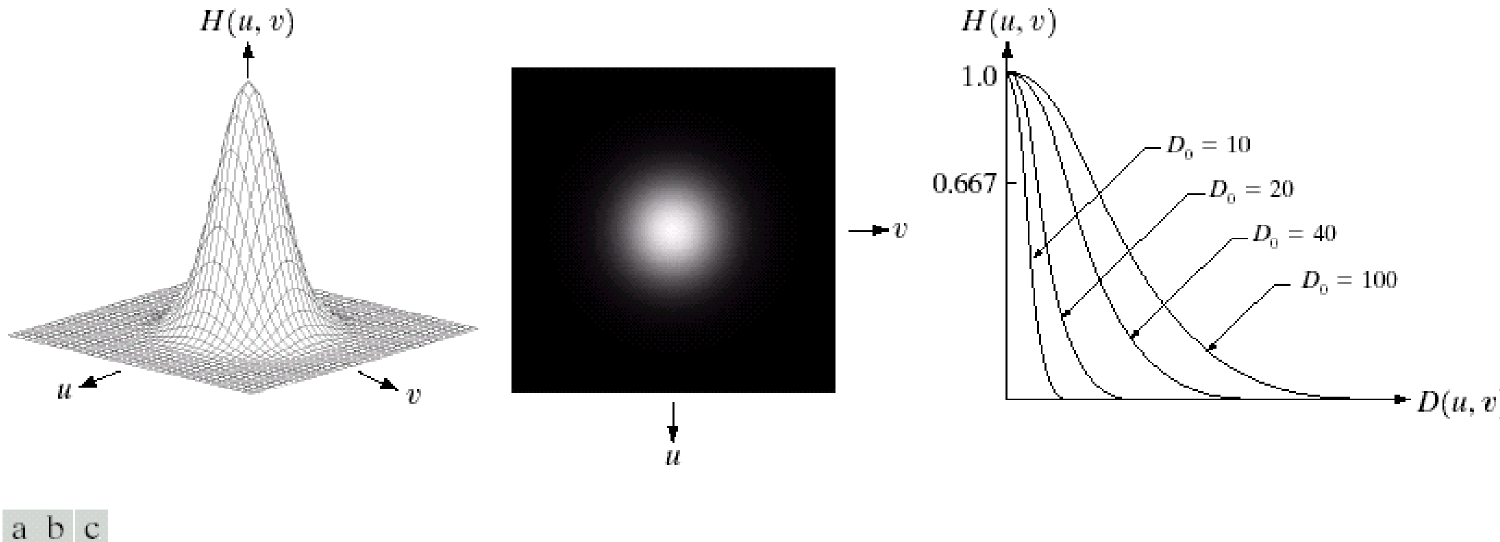
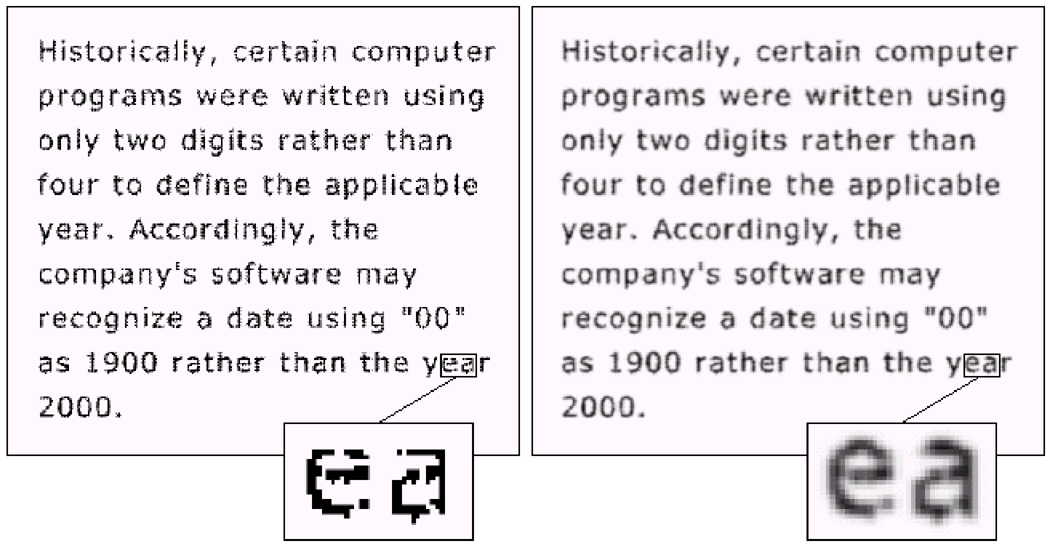


FIGURE 4.17 (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of D_0.

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

smoothing filter application 1

text enhancement



a b

FIGURE 4.19
(a) Sample text of poor resolution (note broken characters in magnified view). (b) Result of filtering with a GLPF (broken character segments were joined).

high-pass filters

$$H_{HPF}(u,v) = 1 - H_{LPF}(u,v)$$

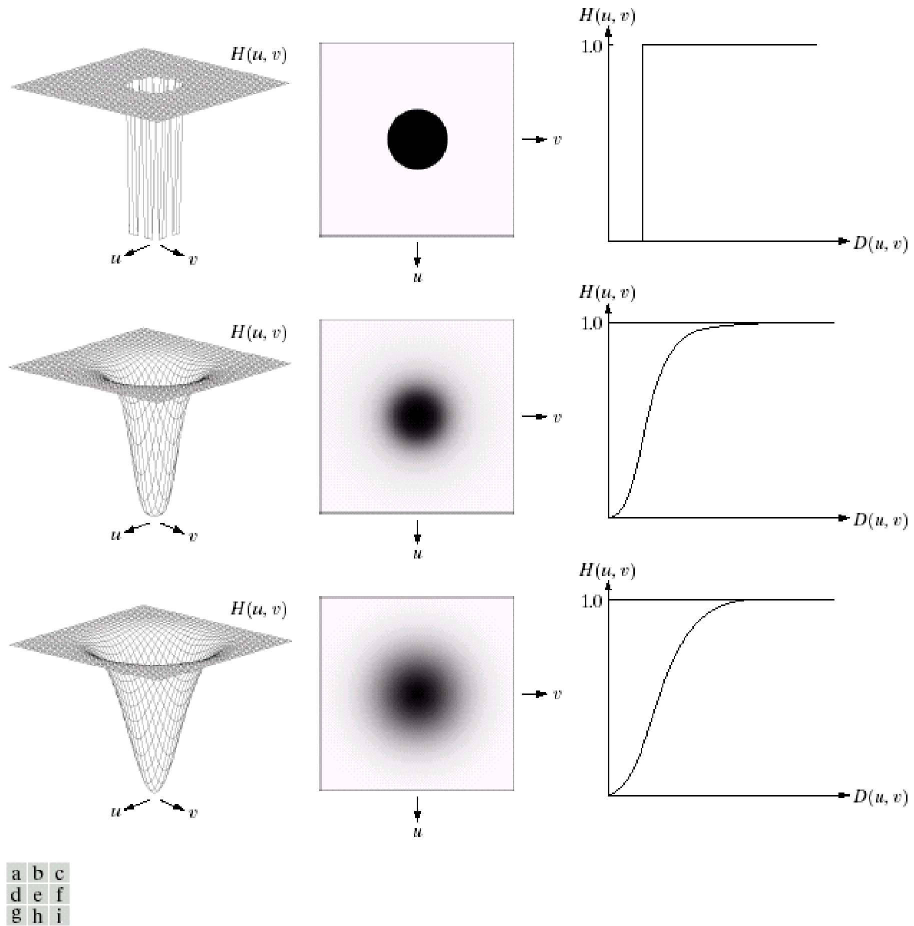
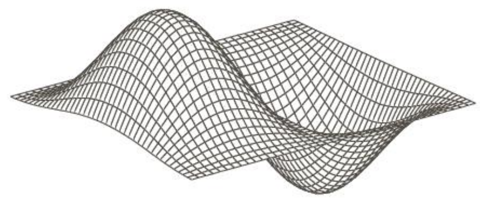
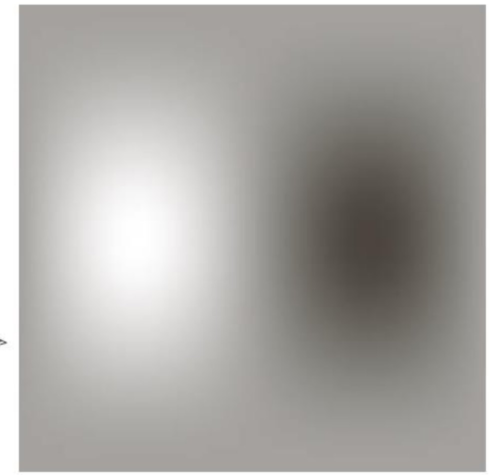


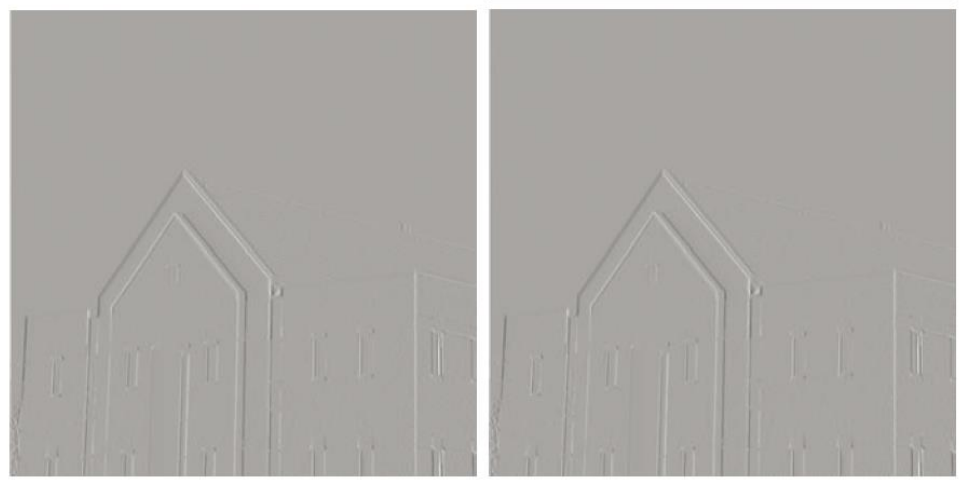
FIGURE 4.22 Top row: Perspective plot, image representation, and cross section of a typical ideal highpass filter. Middle and bottom rows: The same sequence for typical Butterworth and Gaussian highpass filters.

sobel operator in frequency domain

−1	0	1
−2	0	2
−1	0	1







a b
c d

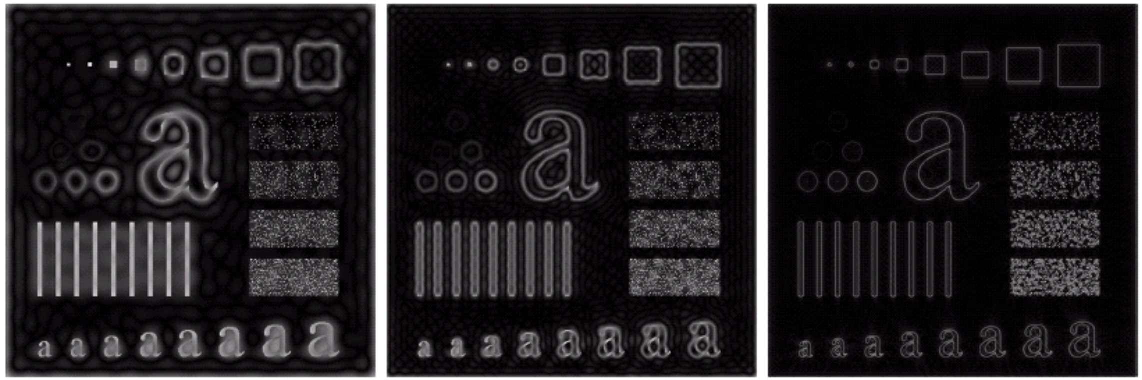
FIGURE 4.39 (a) A spatial mask and perspective plot of its corresponding frequency domain filter. (b) Filter shown as an image. (c) Result of filtering Fig. 4.38(a) in the frequency domain with the filter in (b). (d) Result of filtering the same image with the spatial filter in (a). The results are identical.

Question:

Sobel vs. other high-pass filters?

Spatial vs frequency domain implementation?

high-pass filter examples



a b c

FIGURE 4.24 Results of ideal highpass filtering the image in Fig. 4.11(a) with $D_0 = 15$, 30, and 80, respectively. Problems with ringing are quite evident in (a) and (b).

a b c

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

The Desirables for Image Transforms

- Theory
  - Inverse transform available
  - Energy conservation (Parseval's Thm)
  - Compacting energy
  - Orthonormal, complete basis
  - Shift Invariant, Rotation Invariant
- Implementation
  - Separable
  - Real-valued
  - Fast to compute (FFT)
  - Same implementation for forward and inverse transform
- Application
  - Useful for image enhancement
  - Useful for image representation (compression)
  - Captures meaningful structures in images for characterization, segmentation, matching, detection, recognition, Image retrieval,…

DFT vs. DCT

$$y = Ax$$

1D-DCT

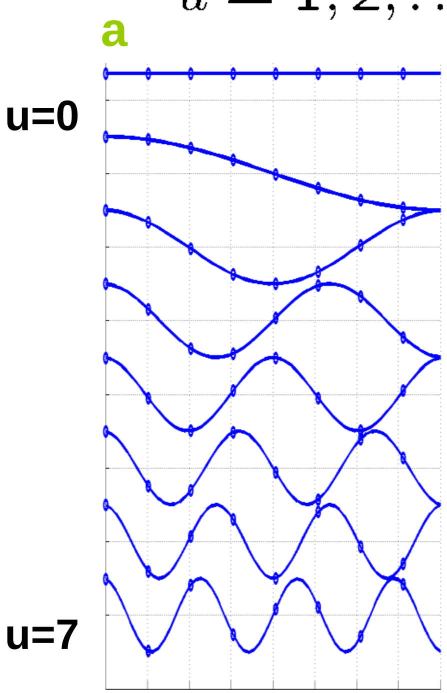
$$a(0,n) = \sqrt{\frac{1}{N}} \qquad u = 0$$

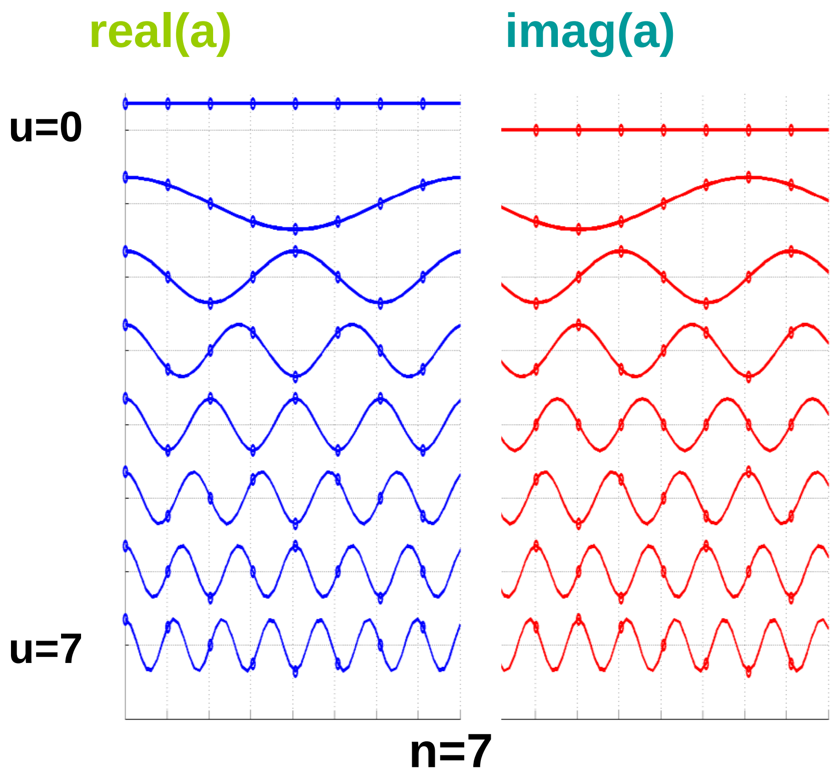
$$a(u,n) = \sqrt{\frac{2}{N}} cos \frac{\pi(2n+1)u}{2N}$$

$$u = 1, 2, \ldots, N-1$$

1D-DFT

$$\begin{aligned} a(u,n) &= e^{-j2\pi \frac{un}{N}} \\ &= cos(2\pi \frac{un}{N}) + j sin(2\pi \frac{un}{N}) \end{aligned}$$





1-D Discrete Cosine Transform (DCT)

$$\begin{cases} Z(k) = \sum_{n=0}^{N-1} z(n) \cdot \alpha(k) \cos\left[\frac{\pi(2n+1)k}{2N} \right] \\ z(n) = \sum_{k=0}^{N-1} Z(k) \cdot \alpha(k) \cos\left[\frac{\pi(2n+1)k}{2N} \right] \end{cases}$$

$$\alpha(0) = \frac{1}{\sqrt{N}}, \alpha(k) = \sqrt{\frac{2}{N}}$$

- Transform matrix *A*
  - a(k,n) = α(0) for k=0
  - a(k,n) = α(k) cos[π(2n+1)/2N] for k>0

- A is real and orthogonal
  - rows of A form orthonormal basis
  - A is not symmetric!
  - DCT is <u>not</u> the real part of unitary DFT!

DFT and DCT in Matrix Notations

Matrix notation for 1D transform

$$y = Ax, \;\; x = A^{-1}y$$

1D-DCT

$$a(0,n) \;=\; \sqrt{\frac{1}{N}} \qquad u = 0$$

$$a(u,n) \;=\; \sqrt{\frac{2}{N}} cos\frac{\pi(2n+1)u}{2N}$$

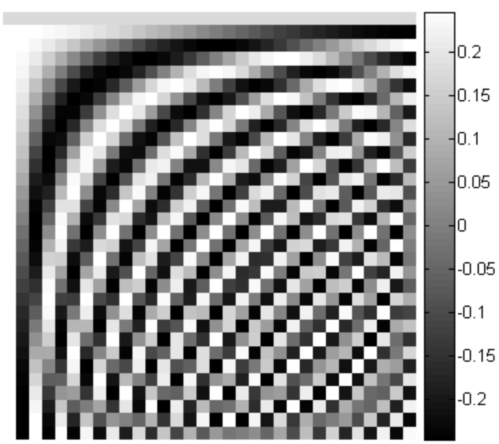
$$u = 1, 2, \ldots, N-1$$

1D-DFT

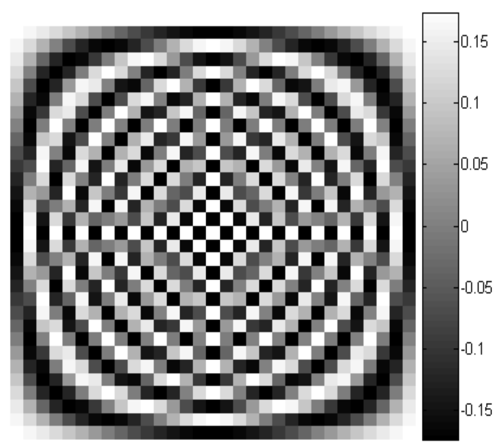
$$\begin{aligned} a(u,n) &= e^{-j2\pi\frac{un}{N}} \\ &= cos(2\pi\frac{un}{N}) - jsin(2\pi\frac{un}{N}) \end{aligned}$$

N=32

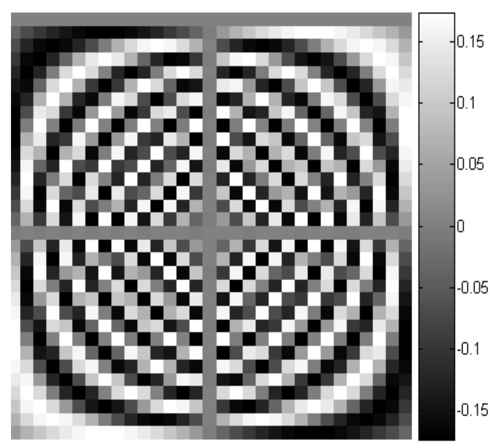
A



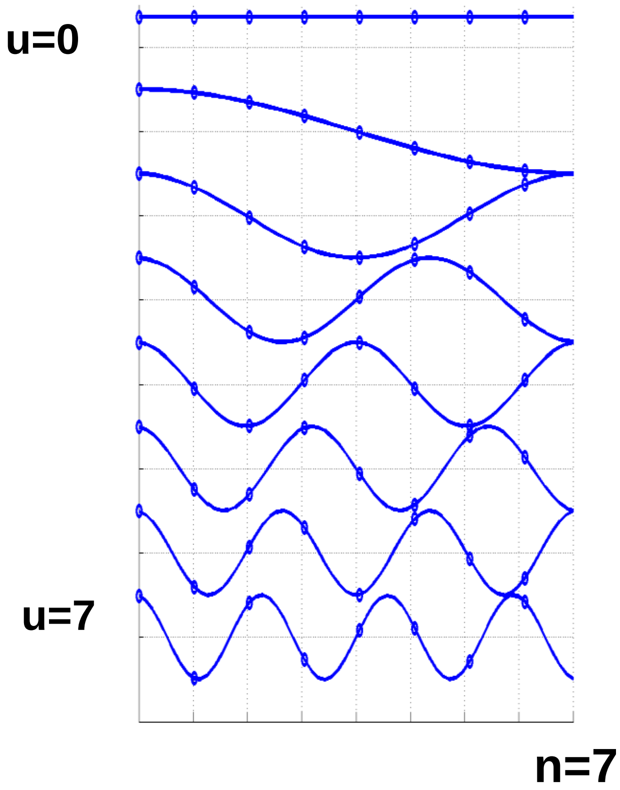
real(A)

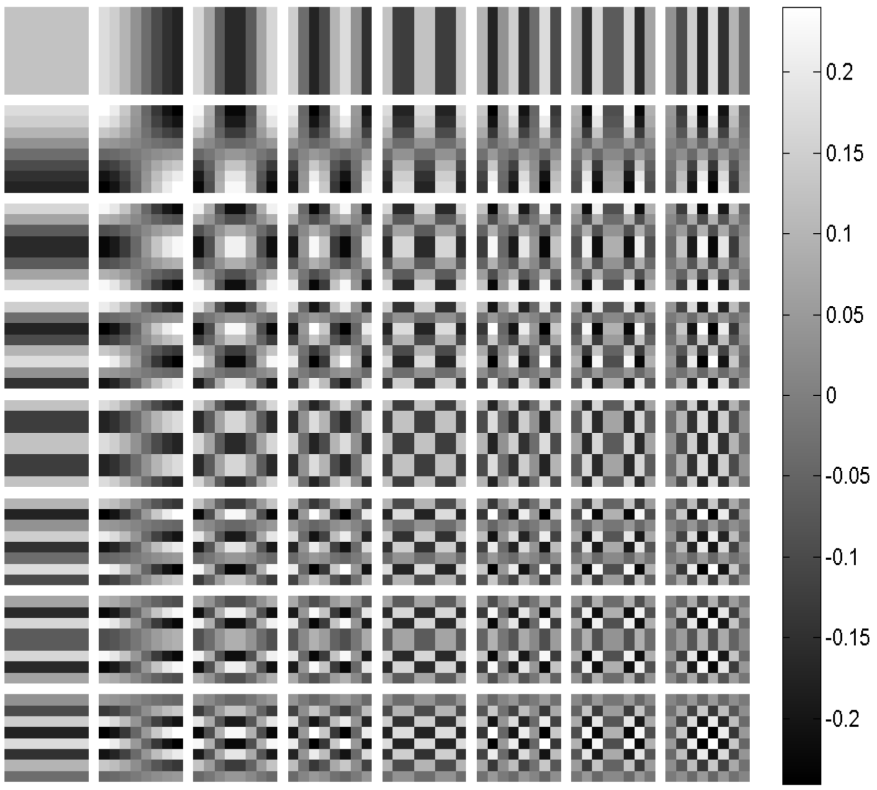


imag(A)



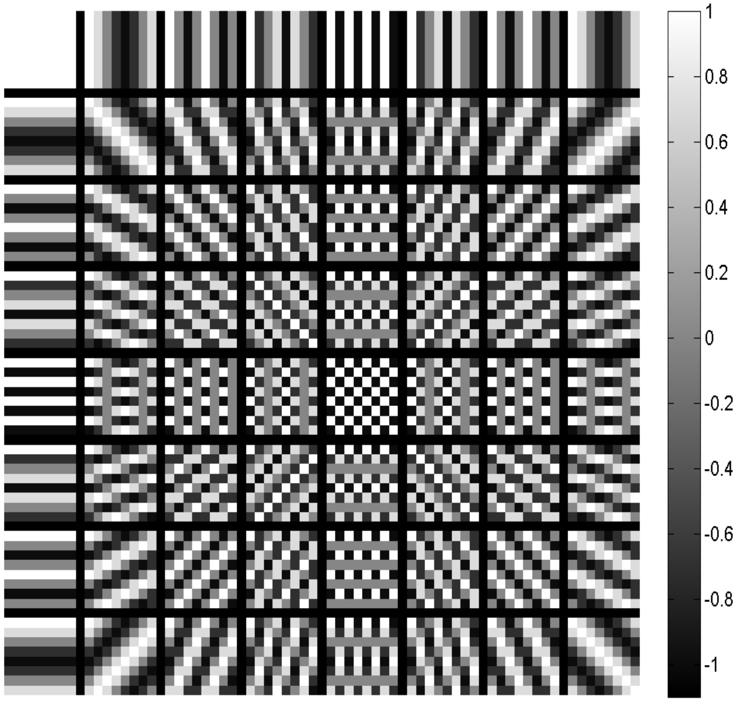
From 1D-DCT to 2D-DCT

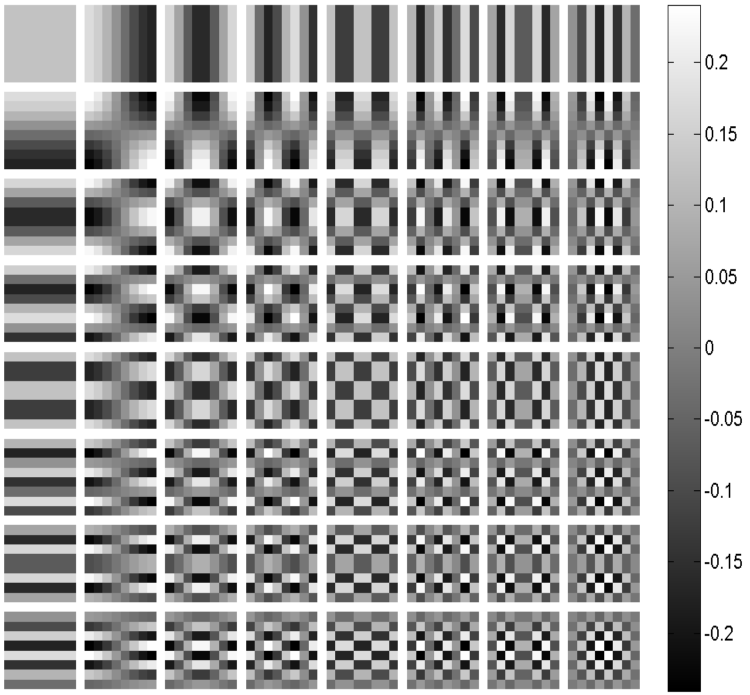




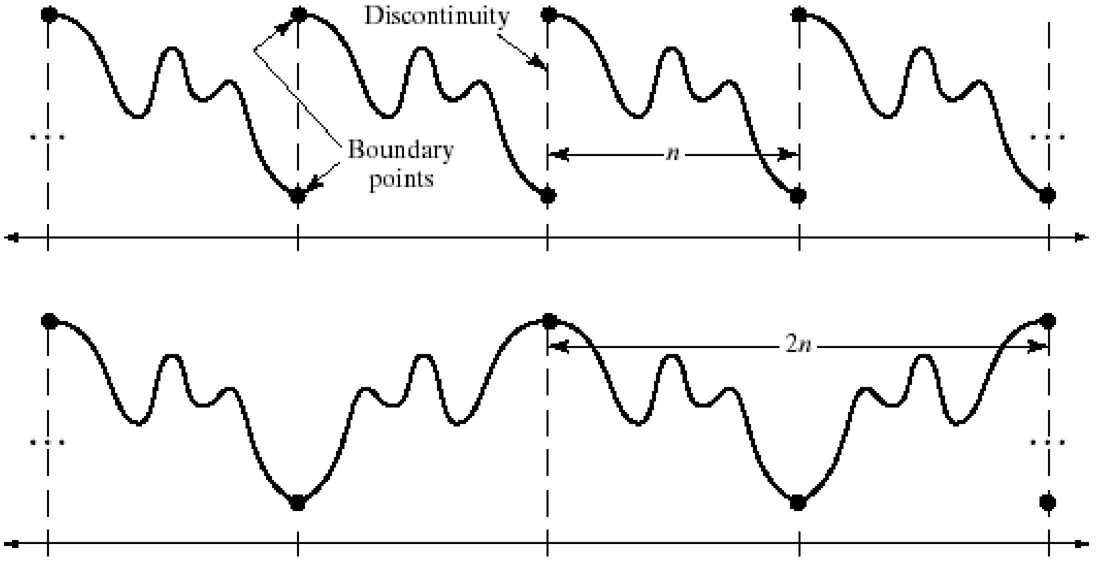
- **Rows of A form a set of orthonormal basis**
- **A is not symmetric!**
- **DCT is <u>not</u> the real part of unitary DFT!**

basis images: DFT (real) vs DCT





Periodicity Implied by DFT and DCT



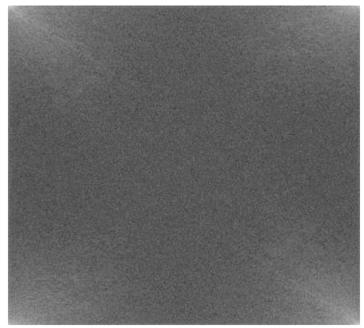
a
b

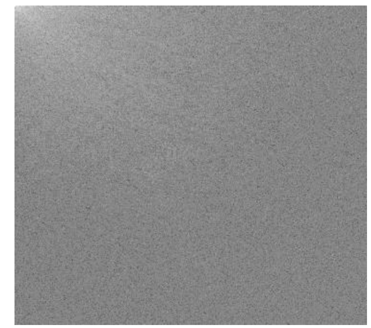
FIGURE 8.32 The periodicity implicit in the 1-D (a) DFT and (b) DCT.

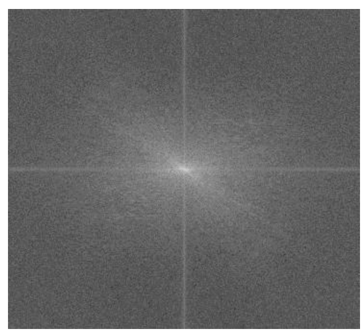
DFT and DCT on Lena

DFT2

DCT2







Shift low-freq to the center