ELEG404/604: Digital Imaging & Photography

Gonzalo R. Arce

Department of Electrical and Computer Engineering
University of Delaware

Chapter VI

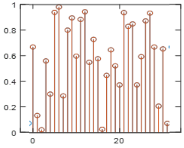
The Discrete Fourier Transform

The DFT is

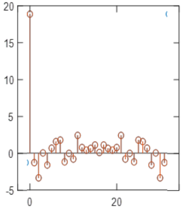
$$G(u) = \Sigma_{x=0}^{M-1} g(x) e^{-2j\pi ux/M} \quad u = 0, 1, \ldots M-1.$$

$$g(x) = \frac{1}{M} \Sigma_{u=0}^{M-1} G(u) e^{2j\pi ux/M} \quad x = 0, 1, \ldots M-1.$$

g(x)



G(u)

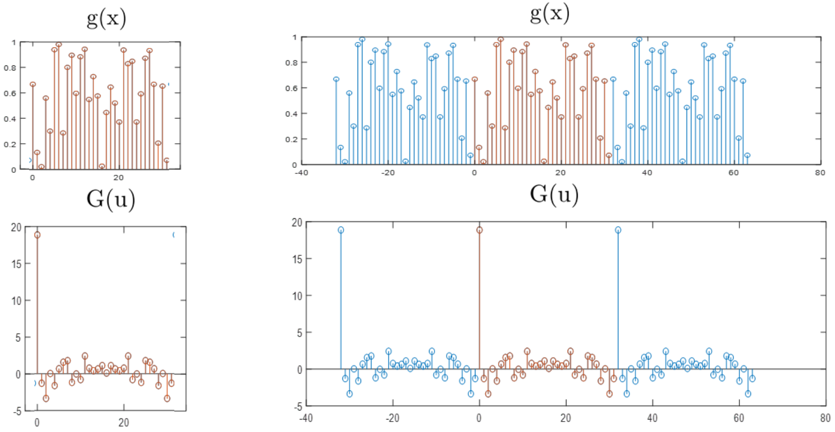


The Discrete Fourier Transform

The DFT is

$$G(u) = \Sigma_{x=0}^{M-1} g(x) e^{-2j\pi ux/M} \quad u = 0, 1, \ldots M-1.$$

$$g(x) = \frac{1}{M} \Sigma_{u=0}^{M-1} G(u) e^{2j\pi ux/M} \quad x = 0, 1, \ldots M-1.$$

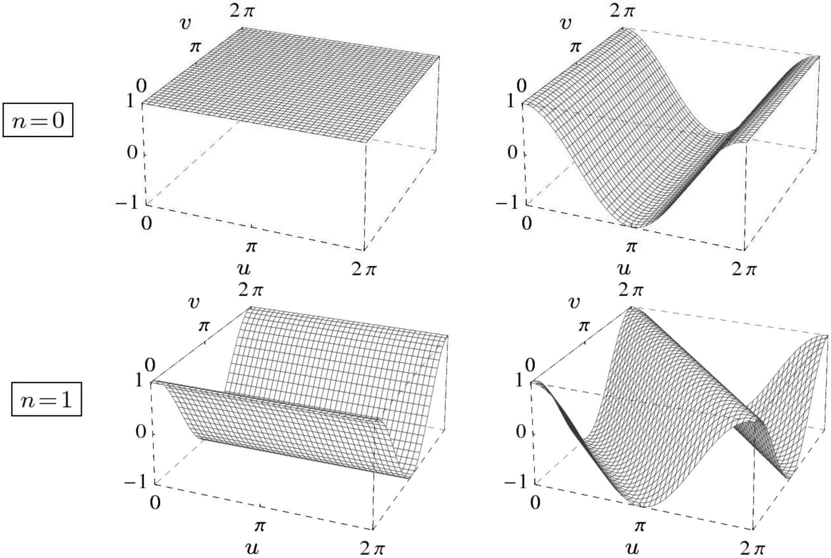


In 2-dimensions the DFT is

$$G(u,v) = \Sigma_{x=0}^{M-1}\Sigma_{y=0}^{N-1} g(x,y)e^{-2j\pi(ux/M+vy/N)} \quad u = 0,1,...M-1; v = 0,1,...N-1.$$

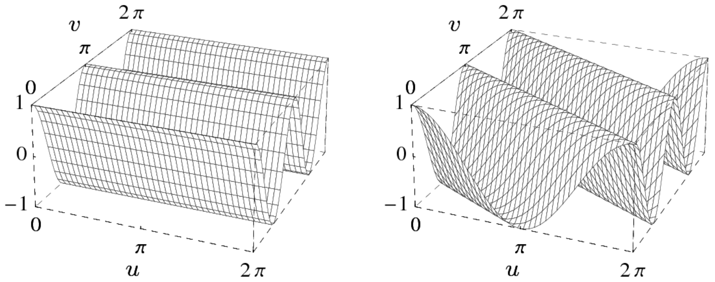
$$g(x,y) = \frac{1}{MN}\Sigma_{u=0}^{M-1}\Sigma_{u=0}^{N-1} G(u,v)e^{2j\pi(ux/M+vy/N)} \quad x = 0,1,...M-1; y = 0,1,...N-$$

Two Dimensional cosine functions

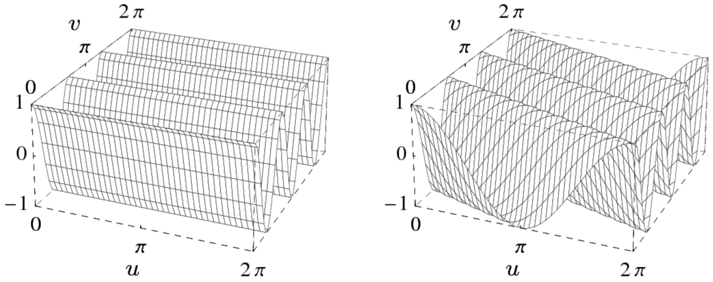


Two Dimensional cosine functions

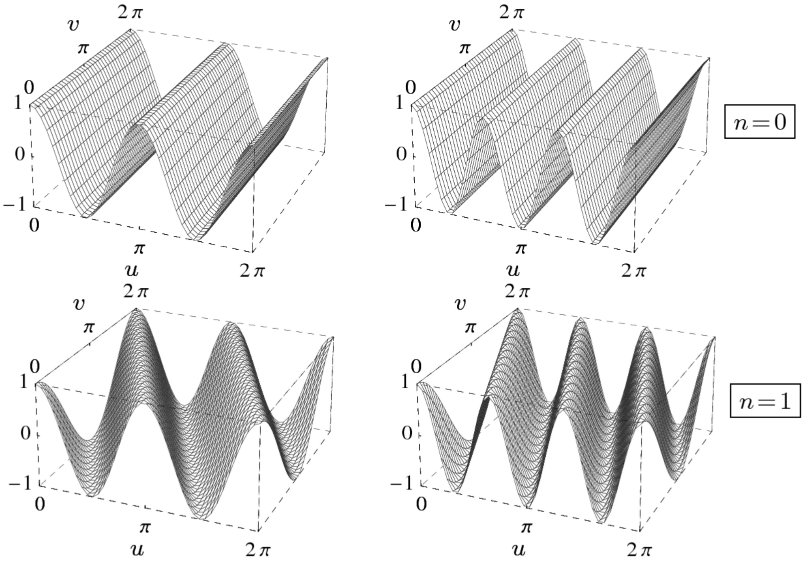
$n=2$



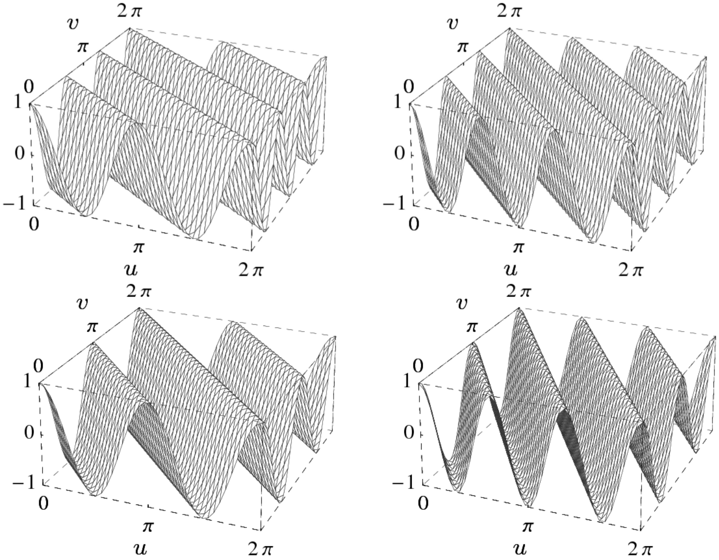
$n=3$



Two Dimensional cosine functions



Two Dimensional cosine functions



$n = 3$

$n = 2$

Some Properties

- Translation

$$g(x,y)e^{j2\pi u_0 x/M} \leftrightarrow G(u-u_0)$$

Example: for $u_0 = M/2$

$$g(x,y)(-1)^x \leftrightarrow G(u-M/2)$$

Example in 2D: for $u_0 = M/2, v_0 = M/2$

$$g(x,y)(-1)^{(x+y)} \leftrightarrow G(u-M/2, v-M/2)$$

- Translation

$$g(x - x_0, y - y_0)$$

- Translation

$$g(x - x_0, y - y_0) \quad \leftrightarrow \quad G(u, v)e^{-j2\pi(x_0u/M + y_0v/M)}$$

- Translation

$$g(x - x_0, y - y_0) \quad \leftrightarrow \quad G(u,v)e^{-j2\pi(x_0 u/M + y_0 v/M)}$$

- Magnitude and Phase: Since the DFT is complex

$$G(u,v) = |G(u,v)|e^{j\varphi(u,v)}$$

where:

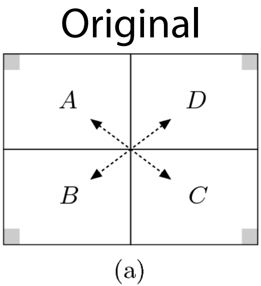
$$|G(u,v)| = [(Re(u,v))^2 + (Im(u,v))^2]^{1/2} \triangleq \text{Fourier spectrum}$$

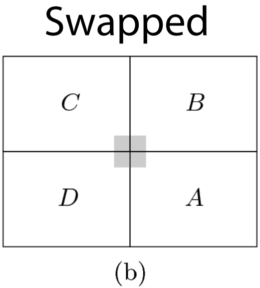
$$\varphi(u,v) = \arctan\left(\frac{Im(u,v)}{Re(u,v)}\right) \triangleq \text{Phase angle}$$

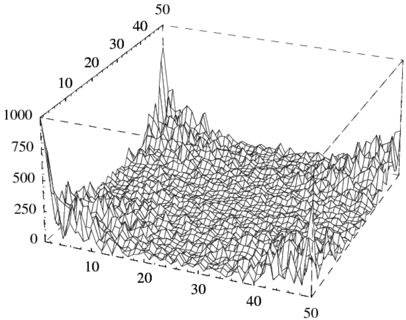
- Power spectrum

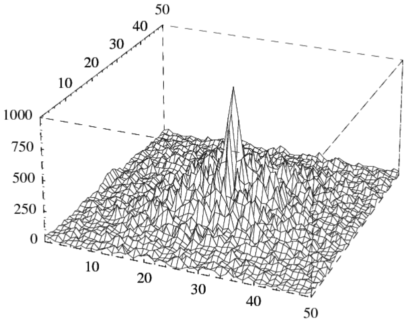
$$P(u,v) = |G(u,v)|^2 = [(Re(u,v))^2 + (Im(u,v))^2]$$

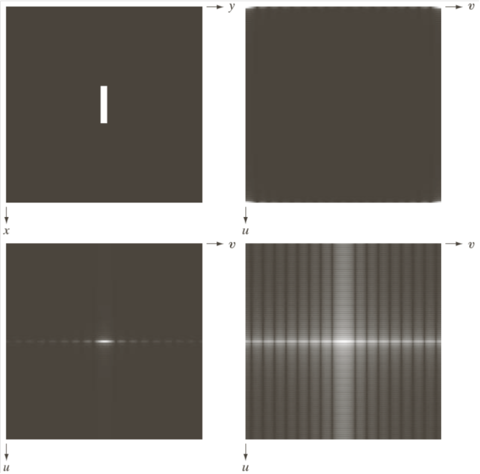
Centering the 2D Spectrum





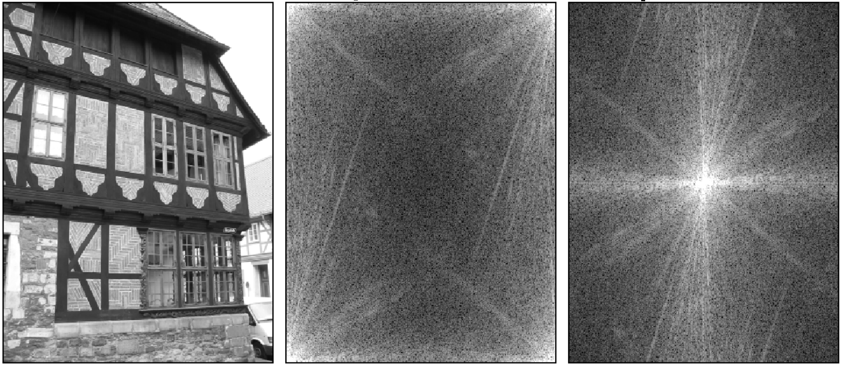


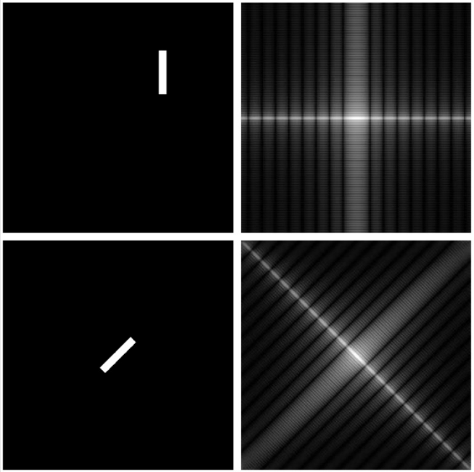


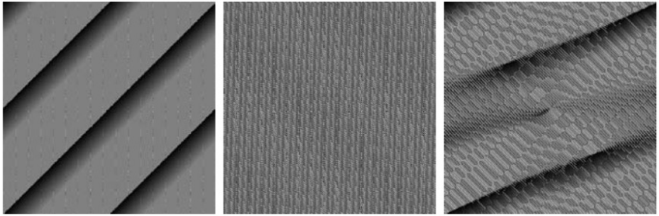

y
x
v
u
v
u
v
u

Intensity plot of 2D power spectrum

Original	Non-centered Spectrum	Centered Spectrum

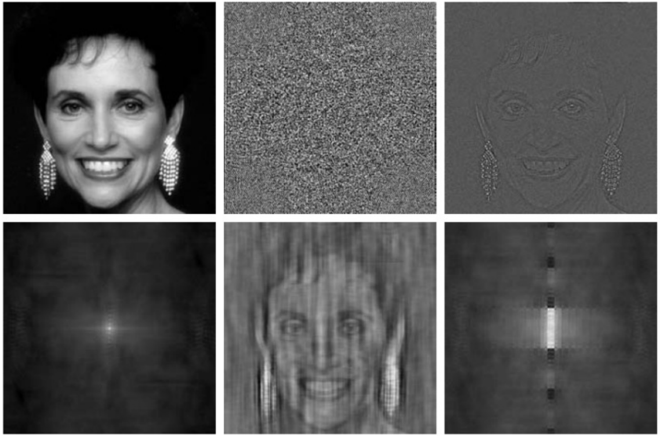






a b c

FIGURE 4.26 Phase angle array corresponding (a) to the image of the centered rectangle in Fig. 4.24(a), (b) to the translated image in Fig. 4.25(a), and (c) to the rotated image in Fig. 4.25(c).



a	b	c
d	e	f

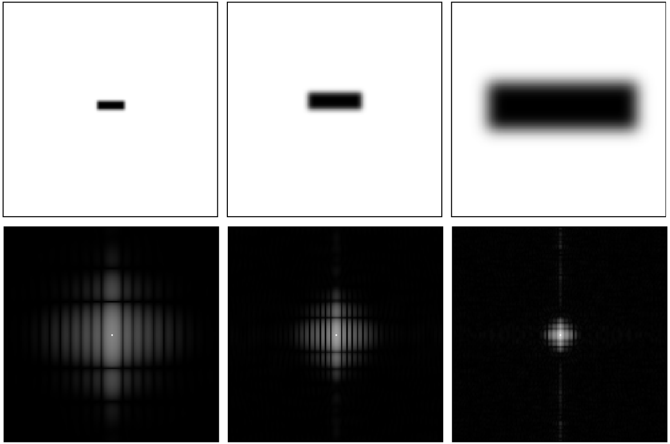
FIGURE 4.27 (a) Woman. (b) Phase angle. (c) Woman reconstructed using only the phase angle. (d) Woman reconstructed using only the spectrum. (e) Reconstruction using the phase angle corresponding to the woman and the spectrum corresponding to the rectangle in Fig. 4.24(a). (f) Reconstruction using the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

DFT- image scaling

Stretching the image causes the spectrum to contract and vice versa

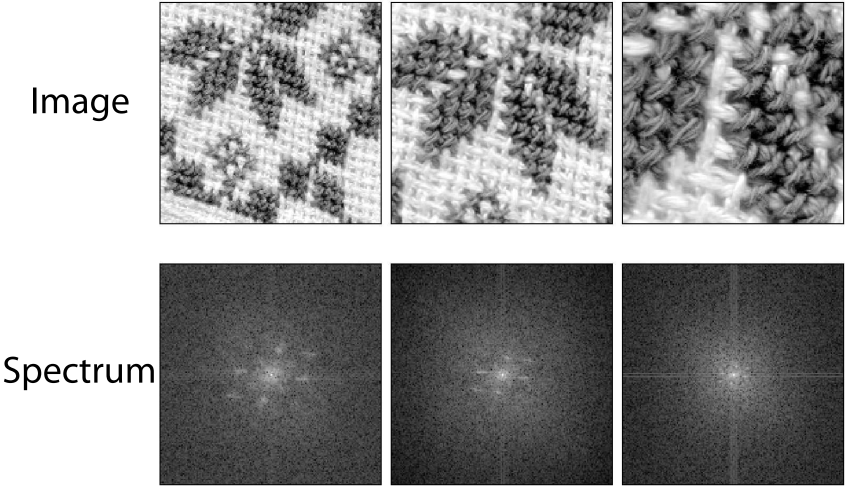
Rectangular Pulse

Spectrum



DFT- oriented, repetitive patterns

Enlarging the image causes the spectrum to contract.



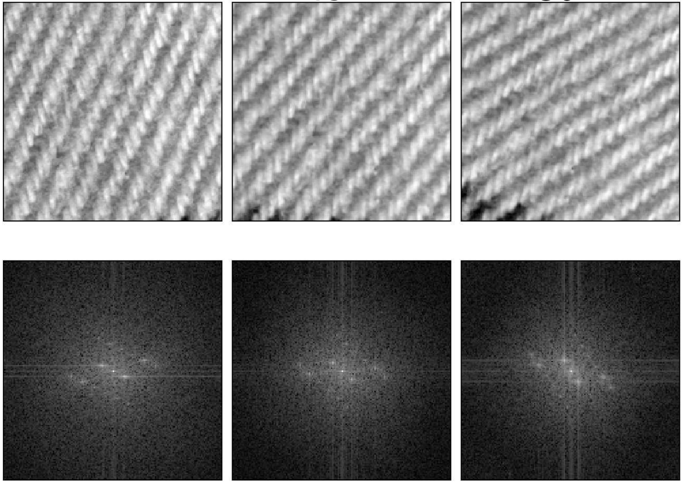
DFT- image rotation

The spectrum turns in the same direction and the same amount as the image.

Original | Rotated 15° | Rotated 30°

Image

Spectrum

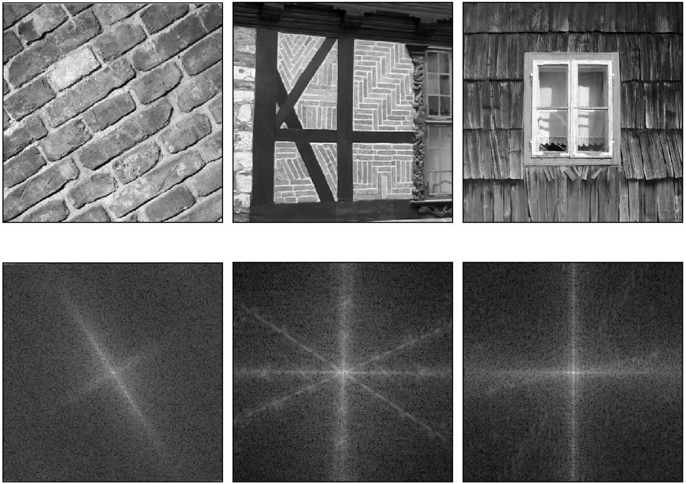


DFT- superposition of image patterns

Broadband effects caused by straight structures, e.g. dark beam on the wall.

Image

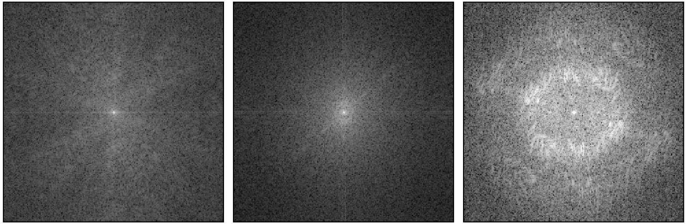
Spectrum



DFT- natural image patterns

Image

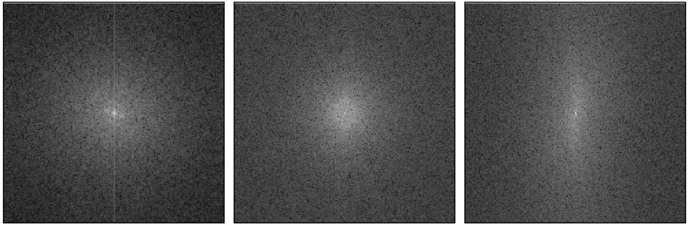
Spectrum



DFT- natural image patterns

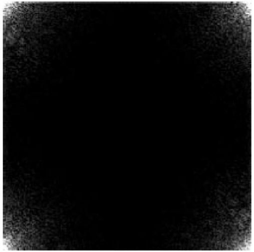
Image

Spectrum

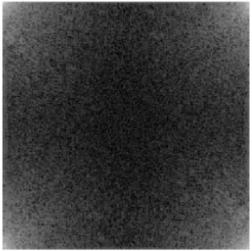


Example of Image DFT (I)

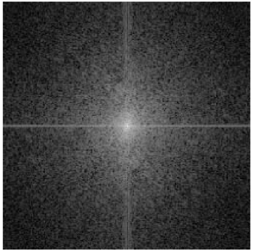
(a) Original



(b) Clipped magnitude, nonordered



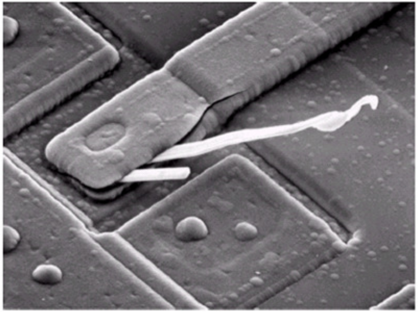
(c) Log magnitude, nonordered

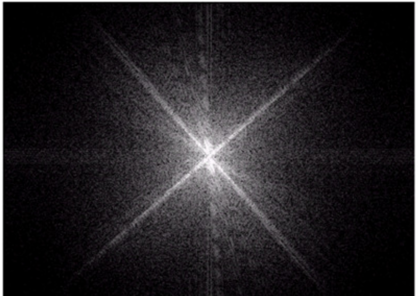


(d) Log magnitude, ordered

Example of Image DFT (II)

- Scanning Electron Microscope (SEM) image of IC board
- Edges correspond to high frequencies
- Note directionality of edges

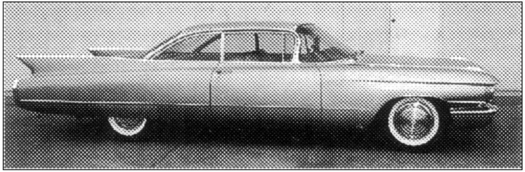




DFT of a print pattern

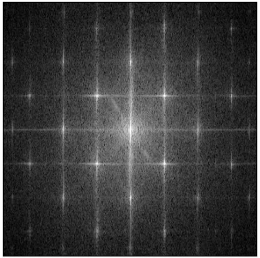
Possible to remove the raster pattern by erasing the peaks in the Fourier spectrum and reconstructing the smoothed image from the modified spectrum using the inverse DFT.

Image



Zoom

Spectrum



The DFT can be redefined as

$$X[n] = \frac{1}{\sqrt{N}} \sum_{m=0}^{N-1} x[m] e^{-mnj2\pi/N} = \sum_{m=0}^{N-1} w_N^{mn} x[m],$$

$$x[m] = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} X[n] e^{mnj2\pi/N} = \sum_{n=0}^{N-1} w_N^{-mn} X[n]$$

$$m, n = 0, 1, \cdots, N-1$$

where $w_N \triangleq e^{-j2\pi/N}/\sqrt{N}$. The time function and its spectrum are periodic: $x[m+kN] = x[m]$ and $X[n+kN] = X[n]$.

The forward and inverse DFT can be written as:

$$X[n] = \frac{1}{\sqrt{N}} \sum_{m=0}^{N-1} x[m] e^{-mnj2\pi/N} = \sum_{m=0}^{N-1} w_N^{mn} x[m],$$

$$x[m] = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} X[n] e^{mnj2\pi/N} = \sum_{n=0}^{N-1} w_N^{-mn} X[n]$$

$$m, n = 0, 1, \cdots, N-1$$

Here we have defined

$$w^{mn} \triangleq \frac{1}{\sqrt{N}} (e^{-j2\pi/N})^{mn}, \qquad w^{*mn} = \frac{1}{\sqrt{N}} (e^{j2\pi/N})^{mn}$$

and w^{*mn} is its complex conjugate of w^{mn}. We further define an $N \times N$ matrix

$$\mathbf{W} = \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & w^{mn} & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix}_{N \times N}$$

where w^{mn} is the element in the mth row and nth column of $\mathbf{W}$.

$\mathbf{W}$ is symmetric ($w^{mn} = w^{nm}$)

$$\mathbf{W}^T = \mathbf{W}$$

and the rows (or columns) of $\mathbf{W}$ are orthogonal:

$$\langle \mathbf{w}_m, \mathbf{w}_n \rangle = \sum_{k=0}^{N-1} w^{*km} w^{kn} = \frac{1}{N} \sum_{k=0}^{N-1} (e^{j2\pi/N})^{mk} (e^{-j2\pi/N})^{nk}$$

$$= \frac{1}{N} \sum_{k=0}^{N-1} (e^{j2\pi/N})^{(m-n)k} \stackrel{*}{=} \delta_{mn} = \begin{cases} 1 & m = n \\ 0 & m \neq n \end{cases}$$

as

- If $m = n$, $(e^{j2\pi/N})^{(n-m)k} = 1$ and $\langle \mathbf{w}_m, \mathbf{w}_n \rangle = 1$,
- If $m \neq n$, the summation becomes:

$$\sum_{k=0}^{N-1} (e^{j2\pi(n-m)/N})^k = \frac{1 - (e^{j2\pi(n-m)/N})^N}{1 - e^{j2\pi(n-m)/N}} = 0$$

We see that $\mathbf{W}$ is a unitary matrix (and symmetric):

$$\mathbf{W}^{*T} = \mathbf{W}^* = \mathbf{W}^{-1}$$

Matrix Form of the 1-D DFT

Define the two N-long vectors:

$$\mathbf{X} \triangleq \begin{bmatrix} X[0] \\ \vdots \\ X[N-1] \end{bmatrix}_{N\times 1}, \quad \mathbf{x} \triangleq \begin{bmatrix} x[0] \\ \vdots \\ x[N-1] \end{bmatrix}_{N\times 1}$$

The DFT can then be written more conveniently as a matrix-vector multiplication:

$$\mathbf{X} = \begin{bmatrix} X[0] \\ \vdots \\ X[N-1] \end{bmatrix} = \frac{1}{\sqrt{N}} \begin{bmatrix} \vdots & \vdots & \vdots \\ \vdots & (e^{-j2\pi/N})^{mn} & \vdots \\ \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} x[0] \\ \vdots \\ x[N-1] \end{bmatrix} = \mathbf{W}\mathbf{x}$$

Matrix Form of the 1-D DFT

and

$$\mathbf{x} = \begin{bmatrix} x[0] \\ \cdot \\ \cdot \\ x[N-1] \end{bmatrix} = \frac{1}{\sqrt{N}} \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & (e^{j2\pi/N})^{mn} & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} \begin{bmatrix} X[0] \\ \cdot \\ \cdot \\ X[N-1] \end{bmatrix} = \mathbf{W}^* \mathbf{X} = \mathbf{W}^{-1} \mathbf{X}$$

The computational complexity of the 1-D DFT is $O(N^2)$, which, as we will see later, can be reduced to $O(N\ log_2 N)$ by the Fast Fourier Transform (FFT) algorithm.

Matrix Form of the 2D DFT

Reconsider the 2D DFT:

$$X[k,l] = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} \underbrace{[\frac{1}{\sqrt{M}} \sum_{m=0}^{M-1} x[m,n] e^{-j2\pi \frac{mk}{M}}]}_{X'[k,n]} e^{-j2\pi \frac{nl}{N}}$$

$$= \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} X'[k,n] e^{-j2\pi \frac{nl}{N}} \qquad \text{for} \quad 0 \leq m,k \leq N-1, \; 0 \leq n,l \leq N-1$$

$$X'[k,n] \triangleq \frac{1}{\sqrt{M}} \sum_{m=0}^{N-1} x[m,n] e^{-j2\pi \frac{mk}{M}} \qquad (n = 0, 1, \cdots, N-1)$$

The summation above is with respect to the row index m and the column index n is a fixed parameter, this expression is a one-dimensional Fourier transform of the nth column of $[x]$, which can be written in column vector (vertical) form as:

$$\mathbf{X'}_n = \mathbf{W}^* \mathbf{x}_n$$

Matrix Form of the 2D DFT

Putting all these N columns together, we can write

$$\left[\mathbf{X'}_0, \cdots, \mathbf{X'}_{N-1}\right] = \mathbf{W}\left[\mathbf{x}_0, \cdots, \mathbf{x}_{N-1}\right]$$

or more concisely

$$\mathbf{X'} = \mathbf{W}\mathbf{x}$$

where $\mathbf{W}$ is a M by N Fourier transform matrix.

Matrix Form of the 2D DFT

$X[k,l] = \frac{1}{\sqrt{N}} \sum_{n=0}^{N-1} X'[k,n] e^{-j2\pi \frac{nl}{N}}$ The sum is with respect to the column index n and the row index number k is fixed, this is a one-dimensional Fourier transform of the kth row of $\mathbf{X}'$, which can be written in row vector (horizontal) form as

$$\mathbf{X}_k^T = \mathbf{X'}_k^T \mathbf{W}^T, \quad (k = 0, \cdots, N-1)$$

Putting all these N rows together, we can write

$$\begin{bmatrix} \mathbf{X}_0^T \\ \cdot \\ \cdot \\ \mathbf{X}_{N-1}^T \end{bmatrix} = \begin{bmatrix} \mathbf{X'}_0^T \\ \cdot \\ \cdot \\ \mathbf{X'}_{N-1}^T \end{bmatrix} \mathbf{W}$$

($\mathbf{W}$ is symmetric: $\mathbf{W}^T = \mathbf{W}$), or more concisely

$$\mathbf{X} = \mathbf{X}'\mathbf{W}$$

Matrix Form of the 2D DFT

But since $\mathbf{X}' = \mathbf{W}\,\mathbf{x}$, we have

$$\mathbf{X} = \mathbf{W}\,\mathbf{x}\,\mathbf{W}$$

Hence the 2D DFT can be implemented by transforming all the rows of $\mathbf{x}$ and then transforming all the columns of the resulting matrix. The order of the row and column transforms is not important.
Similarly, the inverse 2D DFT can be written as

$$\mathbf{x} = \mathbf{W}^*\,\mathbf{X}\,\mathbf{W}^*$$

Again note that $\mathbf{W}$ is a symmetric Unitary matrix:

$$\mathbf{W}^{-1} = \mathbf{W}^{*T} = \mathbf{W}^*$$

The complexity of 2D DFT is $O(N^3)$ which can be reduced to $O(N^2 log_2 N)$ if FFT is used.

The Fast Fourier Transform - FFT (1D)

The DFT pair is given by

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j\frac{2\pi}{N}nk} \qquad k = 0, \ldots, N-1 \tag{1}$$

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j\frac{2\pi}{N}nk} \qquad n = 0, \ldots, N-1 \tag{2}$$

The computational complexity for each point of the DFT is:

- $(N-1)$ Complex multiplications
- $(N-1)$ Complex additions

Hence for N points in the sequence we have:

- $O[N(N-1)]$ Complex multiplications
- $O[N(N-1)]$ Complex additions

Consider the decimation in time FFT algorithm.

Divide the DFT in even and odd terms:

$$X(k) = \sum_{r=0}^{(N/2)-1} x(2r)W_N^{2rk} + \sum_{r=0}^{(N/2)-1} x(2r+1)W_N^{(2r+1)k}$$
$$= \sum_{r=0}^{(N/2)-1} x(2r)W_N^{2rk} + W_N^k \sum_{r=0}^{(N/2)-1} x(2r+1)W_N^{2rk} \quad (3)$$

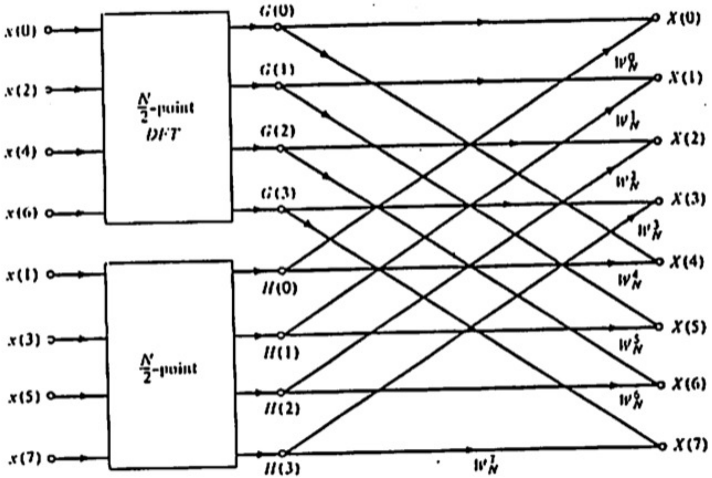
Notice $W_N^{2rk} = e^{-j\frac{2\pi}{N}2rk} = e^{-j\left(\frac{2\pi}{N/2}\right)rk} = W_{N/2}^{rk}$

Hence

$$X(k) = \underbrace{\sum_{r=0}^{(N/2)-1} x(2r)W_{N/2}^{rk}}_{\frac{N}{2}-point\,DFT} + W_N^k \underbrace{\sum_{r=0}^{(N/2)-1} x(2r+1)W_{N/2}^{rk}}_{\frac{N}{2}-point\,DFT} \qquad k = 0, 1, \cdots, N-1 \quad (4)$$

$$X(k) = G(k) + W_N^k H(k) \qquad k = 0, 1, \cdots, N-1 \quad (5)$$

But $G(k)$ and $H(k)$ are periodic in $\frac{N}{2}$.

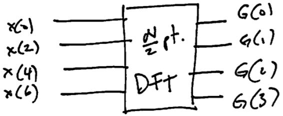


(a) Result of one decimation of the time samples

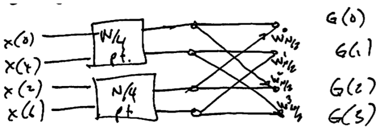
For instance

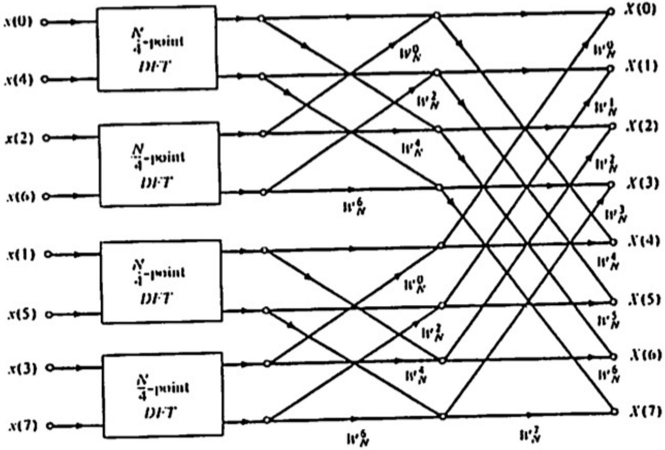
$$\begin{aligned} X(1) &= G(1) + W_N^1 H(1) \qquad (N=8) \\ X(5) &= G(5) + W_N^5 H(5) \\ &= G(1) + W_N^5 H(1) \end{aligned} \tag{6}$$

Each of the $G(k)$ and $H(k)$ are $N/2$ DFT's; however, these can be computed using $N/4$ point DFT's and so on.
For instance the $N/2$ point DFT:



Can be found as:

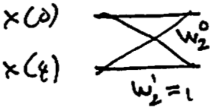




(b) Result of applying two decimations

FIGURE 10-4. Flow graphs showing the decimation-in-time decomposition of an N-point DFT computation ($N = 8$).

It self each 2 point DFT:



If $N = 2^b$ (a power of 2), then we have $log_2 N = b$ decompositions. At each stage we have N complex multiplications and additions. Hence the total number of complexity operations is:

- $O(N log_2 N)$ multiplications.
- $O(N log_2 N)$ additions

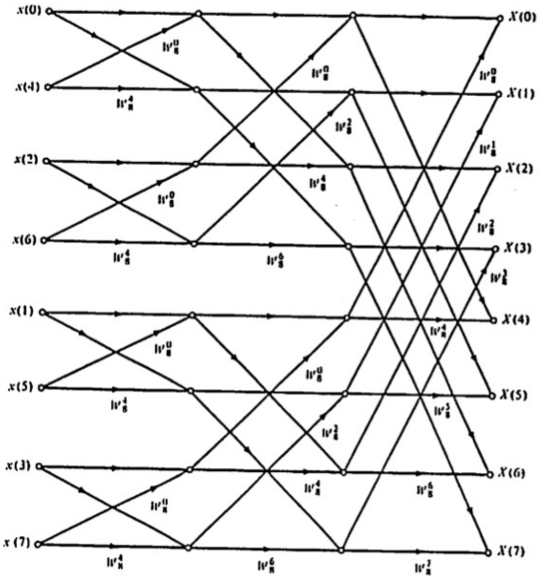


FIGURE 10-5. Complete flow graph for an FFT developed by applying decimation in time ($N = 8$).

CALCULATION OF THE 2-D DFT

1. Direct Calculation

The direct calculation of the 2-D DFT is the double sum:

$$X(k_1,k_2) = \sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1} x(n_1,n_2) w_{N_1}^{n_1 k_1} w_{N_2}^{n_2 k_2}$$
$$0 \leq k_1 \leq N_1 - 1$$
$$0 \leq k_2 \leq N_2 - 1 \qquad (7)$$

where $w_N = e^{\frac{-j2\pi}{N}}$ The evaluation of one sample of $X(k_1,k_2)$ requires $N_1 N_2$ complex multiplications and $N_1 N_2$ complex additions.

Thus, since there are $N_1 N_2$ points. The complexity is in the order of $[N_1^2 N_2^2]$.

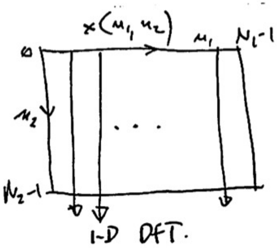
2. Row-Column Decomposition

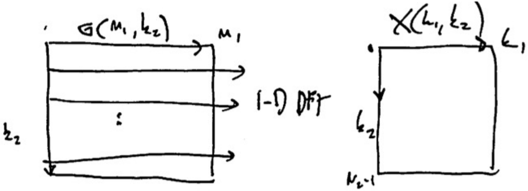
The 2-D DFT can be written as:

$$X(k_1,k_2) = \sum_{n_1=0}^{N_1-1} \underbrace{\left[\sum_{n_2=0}^{N_2-1} x(n_1,n_2) w_{N_2}^{n_2 k_2} \right]}_{G(n_1,k_2)} w_{N_1}^{n_1 k_1} \tag{8}$$

$$X(k_1,k_2) = \sum_{n_1=0}^{N_1-1} G(n_1,k_2) w_{N_1}^{n_1 k_1} \tag{9}$$

Hence





The complexity here is as follow:

$$N_1(1D\, N_2\, pt.\, DFT_s) + N_2(1D\, N_1\, pt.\, DFT) = N_1 N_2^2 + N_2 N_1^2$$

or $\quad N_1 N_2(N_1 + N_2)$

3. Row column FFT

If N_1 and N_2 are powers of 2 then each $1D\, DFT$ can be computed with a $1D\, FFT$. Recall they each $N\, pt\, 1D\, FFT$ has a complexity $N \log N$.

Hence, the complexity is reduced to:

$$N_1 N_2 \log N_2 + N_2 N_1 \log N_1 = N_1 N_2 \log(N_2 N_1) \tag{10}$$

To get a feeling for a numerical savings involved consider a $1024 \times 1024\, 2D\ DFT$.

$$C_{direct} = 2^{40} \approx 10^{12} \text{ complex multiplications}$$
$$C_{r/cdirect} = 2^{31} \approx 10^{9} \text{ complex multiplications}$$
$$C_{r/cFFT} = 10 \times 2^{20} \approx 10^{7} \text{ complex multiplications}$$

If it would take 1 day to process a $2D$ direct, then it would take 1 sec with the r/c FFT!!

Linear Convolution Via DFT

- Recall in one dimension

$$X(k) = \sum_{n=0}^{N-1} x(n) e^{-j2\pi kn/N}, \; k = 0, 1, \ldots, N-1.$$

- The $N \times N$ unitary DFT matrix $\mathbb{W}$ is given by

$$\mathbf{W} = \begin{bmatrix} \cdot & \cdot & \cdot \\ \cdot & w^{mn} & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix}_{N \times N}$$

- Circular convolution Theorem: If

$$x_2(n) = \sum_{k=0}^{N-1} h(n-k)_c x_1(k), \; 0 \leq n \leq N-1$$

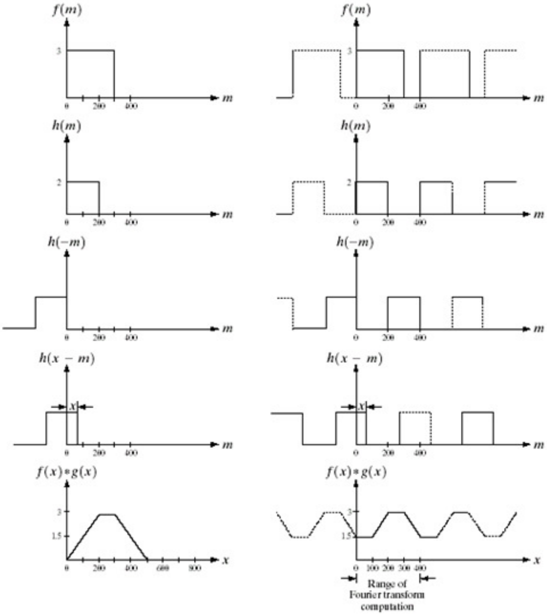
then

$$DFT\{x_2(n)\}_N = DFT\{h(n)\}_N DFT\{x_1(n)\}_N$$

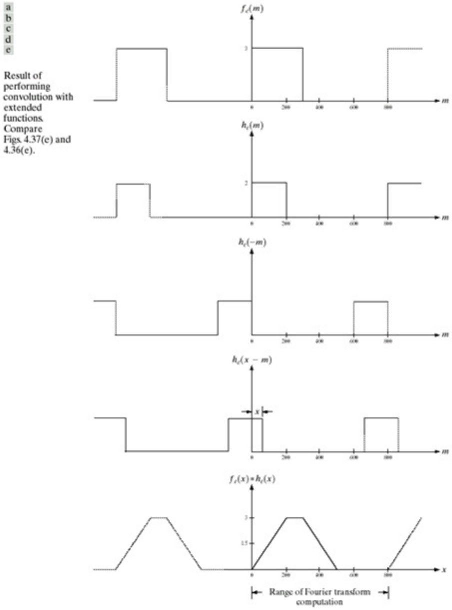
Linear Convolution via DFT (I)

a	f
b	g
c	h
d	i
e	j

Left: convolution of two discrete functions. Right: convolution of the same functions, taking into account the implied periodicity of the DFT. Note in (j) how data from adjacent periods corrupt the result of convolution.



Linear Convolution via DFT (II)



Result of performing convolution with extended functions. Compare Figs. 4.37(e) and 4.36(e).

Linear Convolution via DFT Algorithm

The linear convolution of two sequences $\{h(n)\}_{n=0}^{P-1}$ and $\{x(n)\}_{n=0}^{N-1}$ can be obtained by the following algorithm:

1. Define $M \geq P + N$
2. Define $\tilde{h}(n)$ and $\tilde{x}(n)$ as the M zero extended sequences of $h(n)$ and $x(n)$ respectively
3. Compute $\hat{Y}(k) = \hat{H}(k)\hat{X}(k)$, where $\hat{H}(k) = DFT\{\tilde{h}(n)\}_M$ and $\hat{X}(k) = DFT\{\tilde{x}(n)\}_M$
4. Take the inverse DFT of $\hat{Y}(k)$ to obtain $y(n)$

Two Dimensional DFT

The two dimensional DFT of an $N \times N$ image is a separable transform defined as

$$X(u,v) = \sum_{m=0}^{N-1}\sum_{n=0}^{N-1} f(m,n) w_N^{km} w_N^{ln}, \quad 0 \leq k,l \leq N-1$$

and the inverse transform is defined as

$$x(m,n) = \frac{1}{N^2}\sum_{k=0}^{N-1}\sum_{l=0}^{N-1} X(u,v) w_N^{-km} w_N^{-ln}, \quad 0 \leq m,n \leq N-1.$$

Two Dimensional Linear Convolution

- The DFT of the two dimensional circular convolution of two arrays is the product of their DFTs, *i.e.*, if

$$y(m,n) = \sum_{m'=0}^{N-1} \sum_{n'=0}^{N-1} h(m-m',n-n')_c u(m',n'), \; 0 \leq m,n \leq N-1$$

then

$$DFT\{y(m,n)\}_N = DFT\{h(m,n)\}_N DFT\{u(m,n)\}_N$$

Two Dimensional Linear Convolution

- The DFT of the two dimensional circular convolution of two arrays is the product of their DFTs, *i.e.*, if

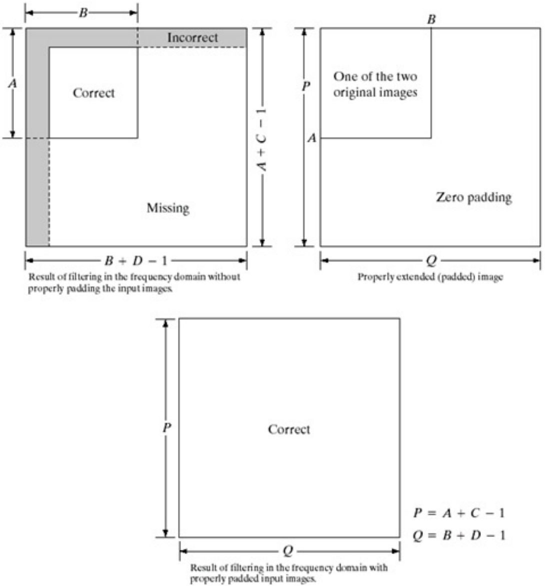
$$y(m,n) = \sum_{m'=0}^{N-1}\sum_{n'=0}^{N-1} h(m-m',n-n')_c u(m',n'),\ 0 \leq m,n \leq N-1$$

then

$$DFT\{y(m,n)\}_N = DFT\{h(m,n)\}_N DFT\{u(m,n)\}_N$$

- Extensions to linear filtering can be done using zero padding

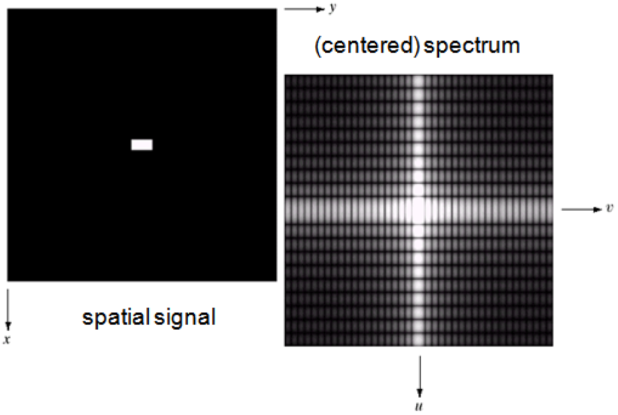
Two dimensional Example of Zero Padding



a b
c

Illustration of the need for function padding.
(a) Result of performing 2-D convolution without padding.
(b) Proper function padding.
(c) Correct convolution result.

Example of Image DFT (II)



We can center the DFT by premultiplying image U by the array $(-1)^{m+n}$

$$X(k+N/2, l+N/2) = \frac{1}{N}\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} x(m,n)(-1)^{m+n} w_N^{km} w_N^{ln}$$

Ideal Low Pass Filters (I)

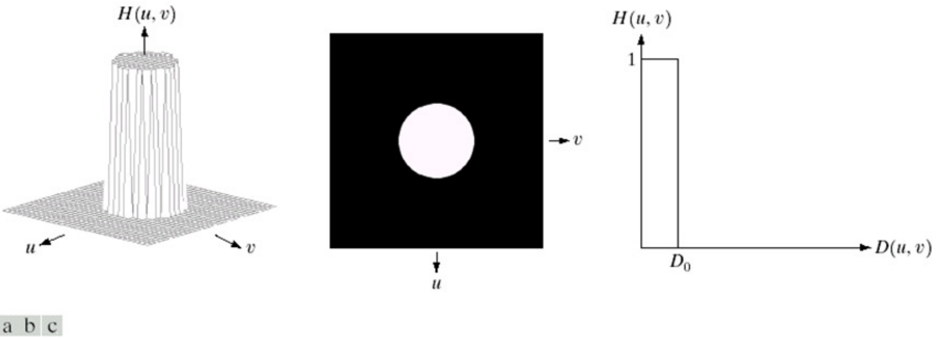


FIGURE (a) Perspective plot of an ideal lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

Ideal Low Pass Filters (II)

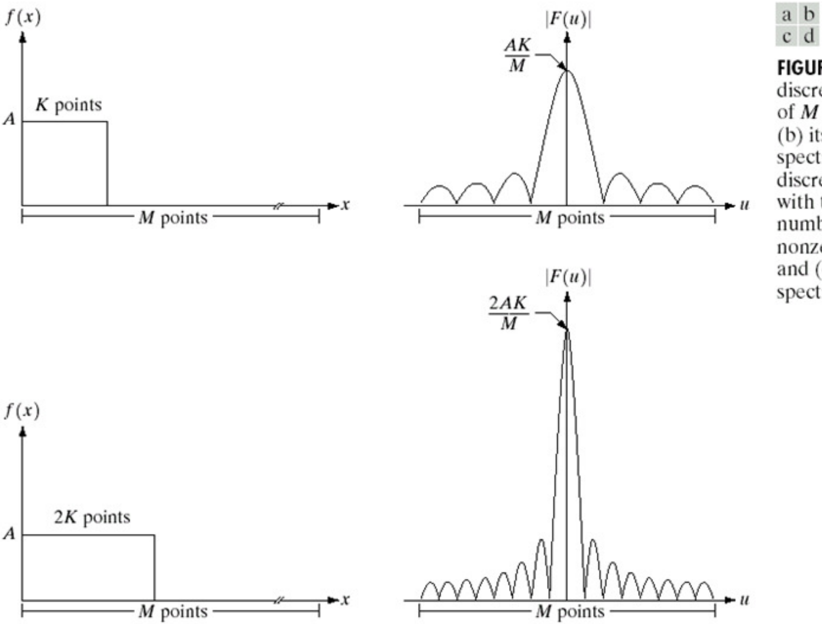
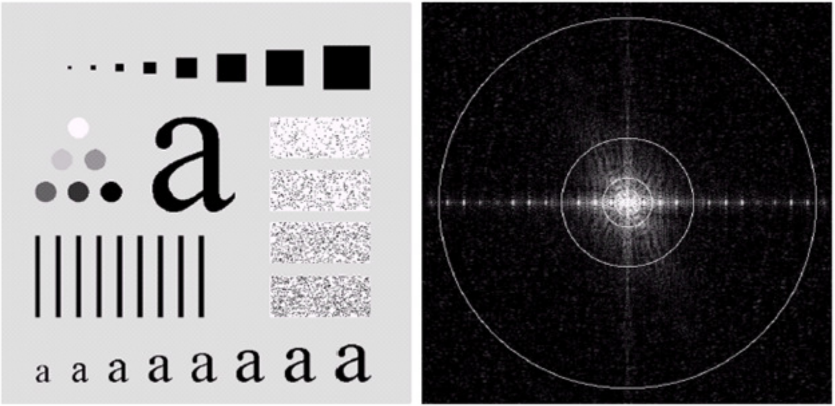


FIGURE (a) A discrete function of M points, and (b) its Fourier spectrum. (c) A discrete function with twice the number of nonzero points, and (d) its Fourier spectrum.

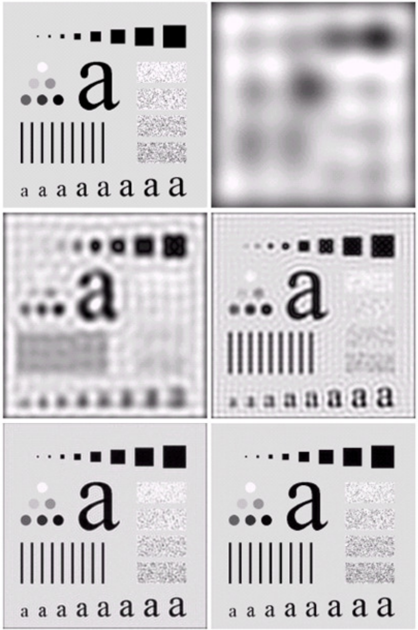
Energy Compaction (I)



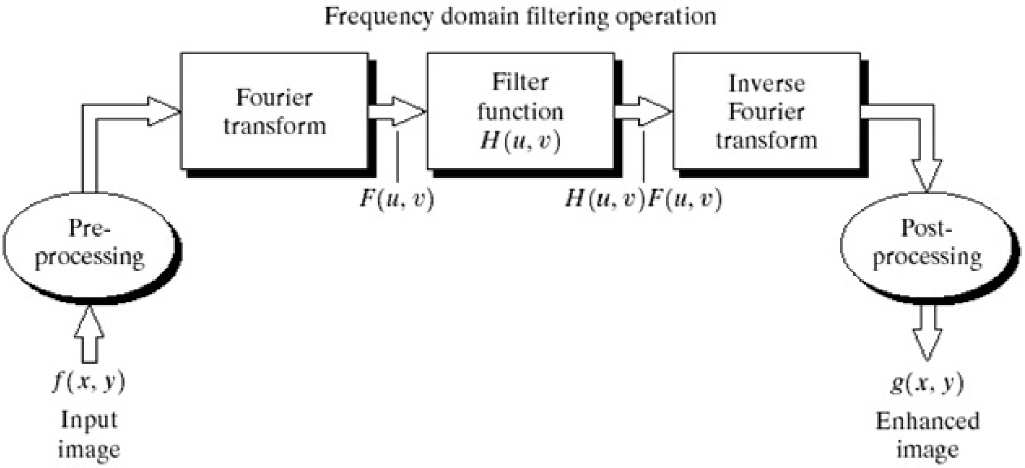
a b

FIGURE (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.

Energy Compaction (II)



Filtering in the Frequency Domain



Low Pass and High Filtering Example

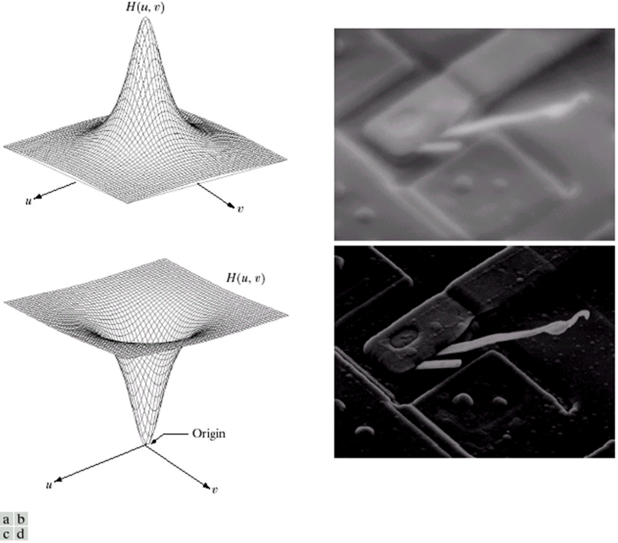
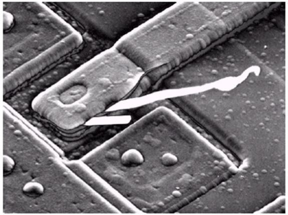


FIGURE 4.7 (a) A two-dimensional lowpass filter function. (b) Result of lowpass filtering the image in Fig. 4.4(a). (c) A two-dimensional highpass filter function. (d) Result of highpass filtering the image in Fig. 4.4(a).



DC shifted highpass filter output

Low Pass Filters

Lowpass filters. D_0 is the cutoff frequency and n is the order of the Butterworth filter.

Ideal	Butterworth	Gaussian
$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \le D_0 \\ 0 & \text{if } D(u,v) > D_0 \end{cases}$	$H(u,v) = \dfrac{1}{1 + [D(u,v)/D_0]^{2n}}$	$H(u,v) = e^{-D^2(u,v)/2D_0^2}$

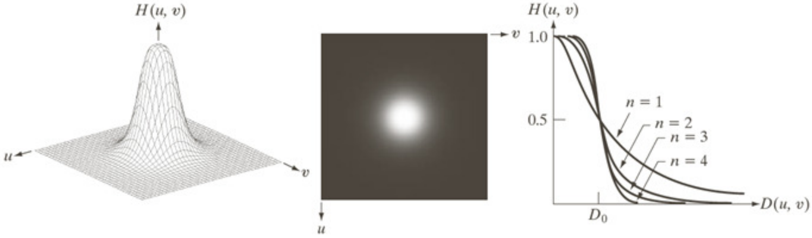
- $D(u,v)$ is the distance from point (u,v) to the origin
- Ideal filter can be implemented digitally but has undesired effects
- Butterworth filter is a smooth approximation to ideal filter
- Gaussian filter is a smooth function both in space and frequency domains

Butterworth Low Pass Filter

Frequency response:

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

- Order: n, Cutoff frequency: D_0
- Smooth transfer function
  - Minimizes ringing
  - Order controls transition bandwidth

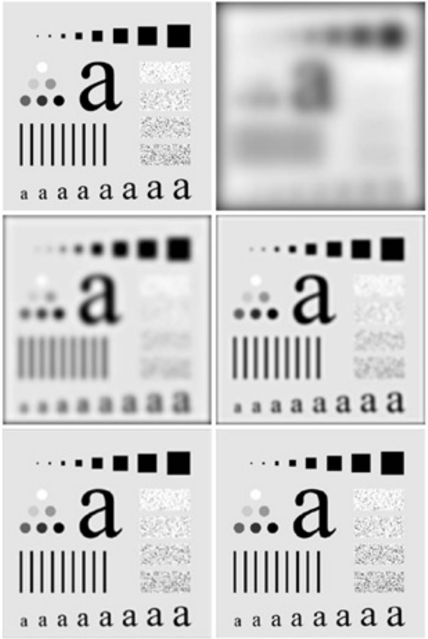


a b c

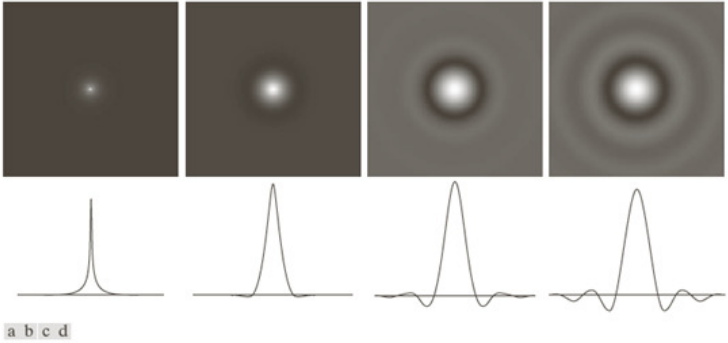
FIGURE (a) Perspective plot of a Butterworth lowpass-filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.

Butterworth Filter Example

- Size 500×500
- Filter order:2
- $D_0 = 5, 15, 30, 80$ and 230
- Significantly reduced ringing compared to ILPF



Spatial Domain Representation of Butterworth Filter



- Cutoff frequency:5
- Increasing filter order: $1, 2, 5$ and 20
  - Impulse response spreads, oscillations introduced
  - Smoothing and ringing introduced

Gaussian Low Pass Filter

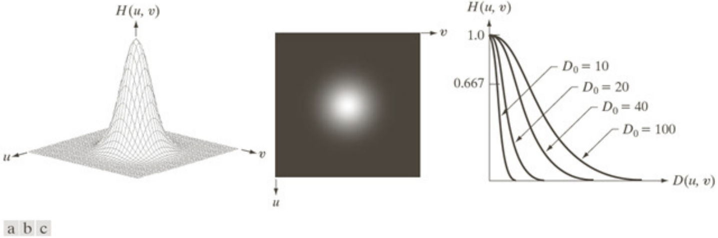


FIGURE (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of D_0.

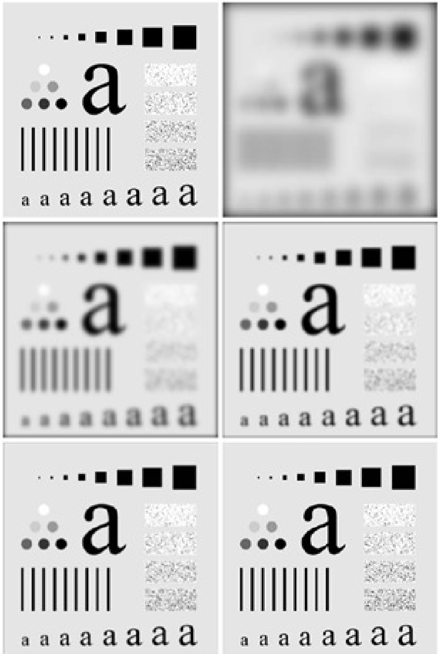
Frequency response:

$$H(u,v) = \exp - D^2(u,v)/2D_0^2$$

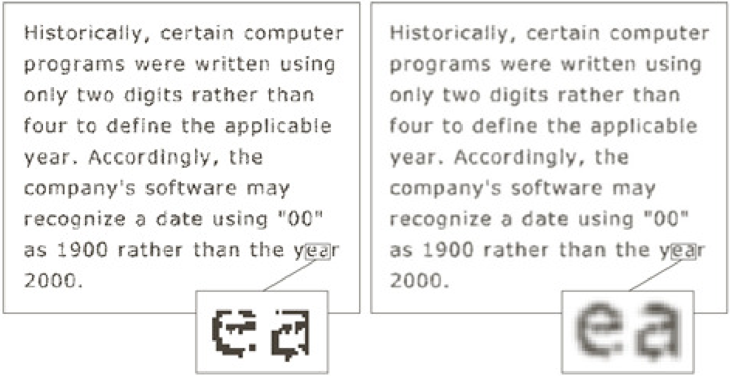
- Spatial domain also a gaussian function
- No ringing
- Less cutoof/transition control

Gaussian Low Pass Filter Example

- $D_0 = 5, 15, 30, 80$ and 230
- Not as much smoothing
- More gradual transition band
- No ringing

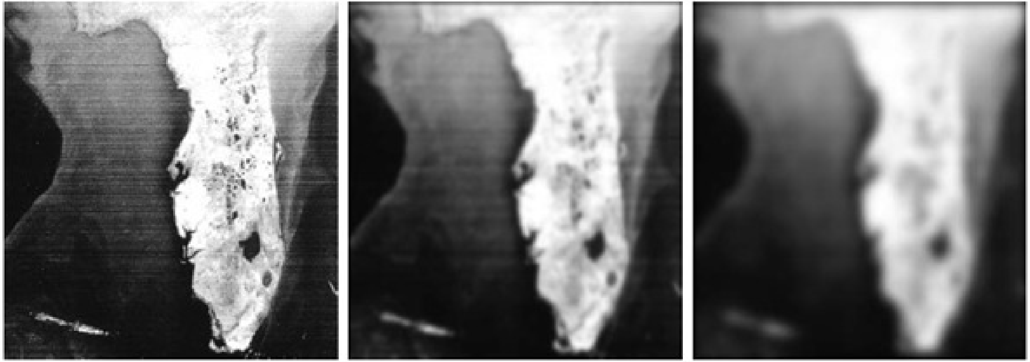


Application Example



- Poor resolution sampled text
  - Scanned material, faxes
  - Broken text
- Result of Gaussian low pass filtering: broken character segments are joined

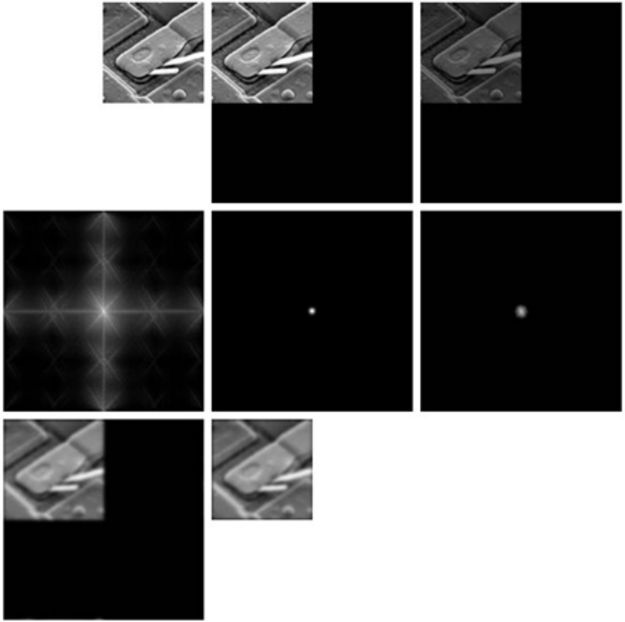
Enhancement of Poorly Acquired Images



a b c

FIGURE (a) Image showing prominent horizontal scan lines. (b) Result of filtering using a GLPF with $D_0 = 50$. (c) Result of using a GLPF with $D_0 = 20$. (Original image courtesy of NOAA.)

Gaussian Filter with Zero Padding Example

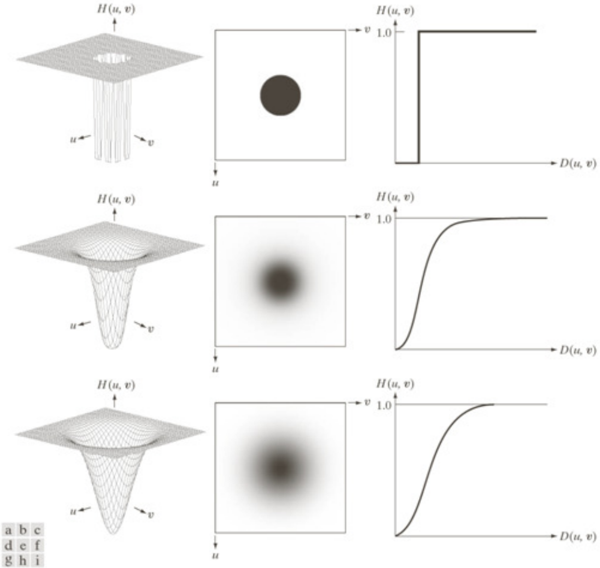


Spectral Representations of Sharpening Filters

- Simple highpass representation

$$H_{hp}(u,v) = 1 - H_{lp}(u,v)$$

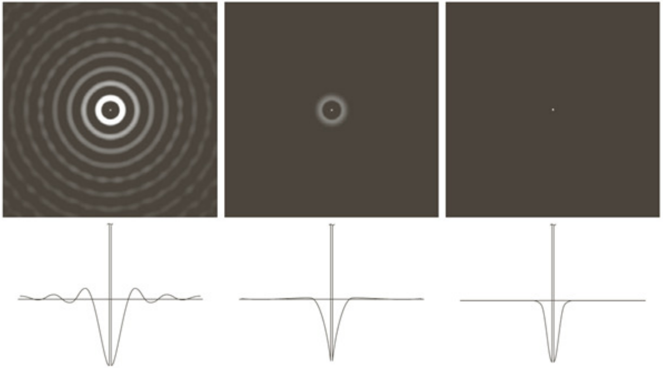
- Spectrally centered examples
  - Ideal
  - Butterworth
  - Gaussian



High Pass Filters

Highpass filters. D_0 is the cutoff frequency and n is the order of the Butterworth filter.

Ideal	Butterworth	Gaussian
$H(u,v) = \begin{cases} 1 & \text{if } D(u,v) \leq D_0 \\ 0 & \text{if } D(u,v) > D_0 \end{cases}$	$H(u,v) = \dfrac{1}{1 + [D_0/D(u,v)]^{2n}}$	$H(u,v) = 1 - e^{-D^2(u,v)/2D_0^2}$



a b c

FIGURE Spatial representation of typical (a) ideal, (b) Butterworth, and (c) Gaussian frequency domain highpass filters, and corresponding intensity profiles through their centers.

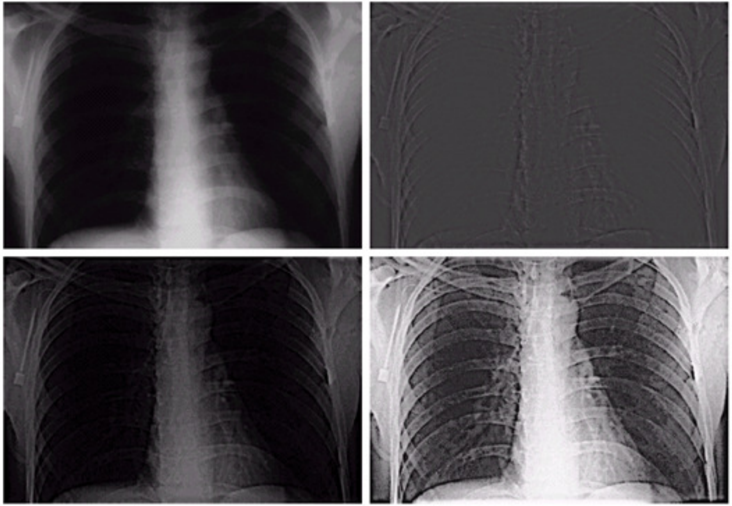
High Pass Filtering Example

Results of highpass filtering the image in Fig. 4.41(a) using an IHPF with $D_0 = 30, 60$, and 160.

Results of highpass filtering the image in Fig. 4.41(a) using a BHPF of order 2 with $D_0 = 30, 60$, and 160.

High Frequency Emphasis Example

High frequency emphasis, $a = 0.5$, $b = 2.0$



a	b
c	d

(a) A chest X-ray image. (b) Result of Butterworth highpass filtering. (c) Result of high-frequency emphasis filtering. (d) Result of performing histogram equalization on (c). (Original image courtesy Dr. Thomas R. Gest, Division of Anatomical Sciences, University of Michigan Medical School.)

Correlation Example

- Correlation measures statistical similarity
- Common application: template matching
- Zero pad image and template
- Multiply DFTs (conjugate image DFTs)
- Invert results
- Find peaks location