Chapter 4
Image Enhancement in the Frequency Domain

Fourier Transform
Frequency Domain Filtering
Low-pass, High-pass, Butterworth, Gaussian
Laplacian, High-boost, Homomorphic
Properties of FT and DFT
Transforms

4.1

Chapter 4
Image Enhancement in the Frequency Domain

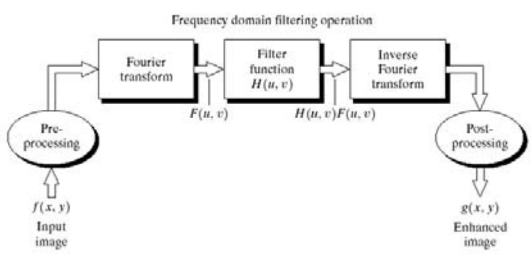


FIGURE 4.5 Basic steps for filtering in the frequency domain.

4.2

Chapter 4
Image Enhancement in the Frequency Domain

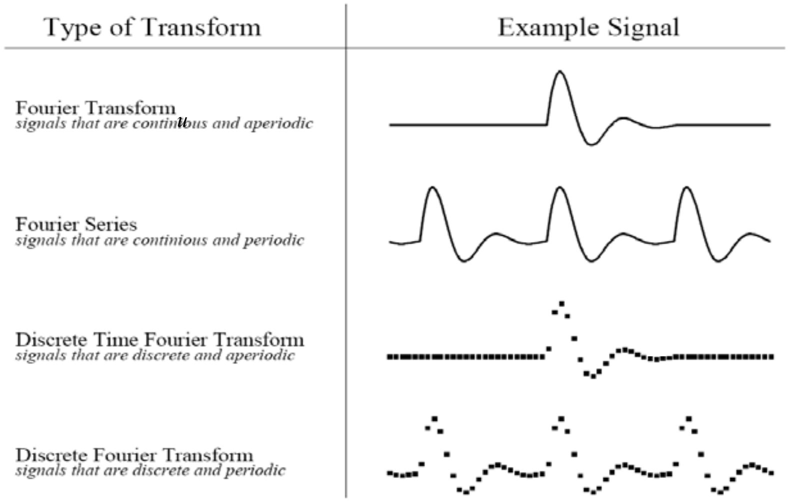
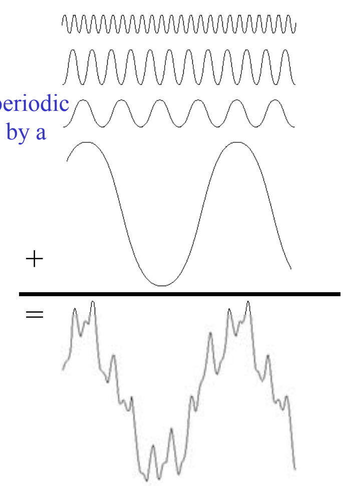


FIGURE 8-2
Illustration of the four Fourier transforms. A signal may be continuous or discrete, and it may be periodic or aperiodic. Together these define four possible combinations, each having its own version of the Fourier transform. The names are not well organized; simply memorize them.

Chapter 4
Image Enhancement in the Frequency Domain

Fourier series states that a periodic function can be represented by a weighted sum of sinusoids



Fourier, 1807

Periodic and non-periodic functions can be represented by an integral of weighted 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.

4.4

Chapter 4
Image Enhancement in the
Frequency Domain

4.2.1 The One-Dimensional Fourier Transform and its Inverse

The Fourier transform, $F(u)$, of a single variable, continuous function, $f(x)$, is defined by the equation

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

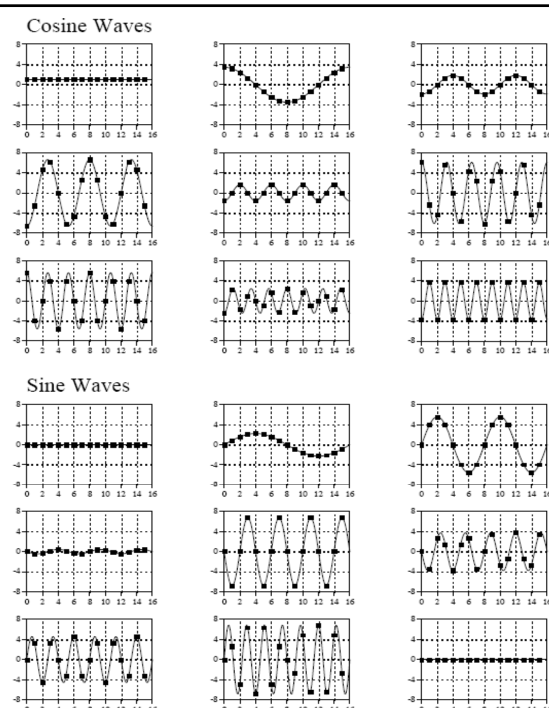
where $j = \sqrt{-1}$. Conversely, given $F(u)$, we can obtain $f(x)$ by means of the *inverse* Fourier transform

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

These two equations comprise the *Fourier transform pair*. They indicate the important fact mentioned in the previous section that a function can be recovered from its transform. These equations are easily extended to two variables, u and v:

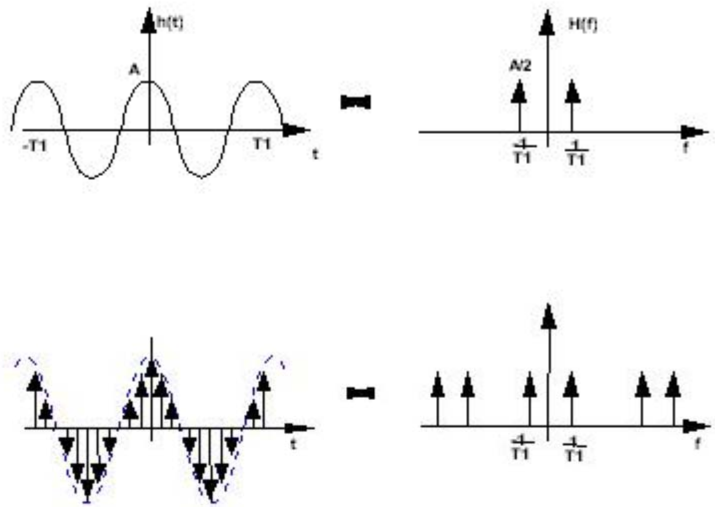
4.5

Chapter 4
Image
Enhancement in the
Frequency Domain



Chapter 4
Image Enhancement in the Frequency Domain

From the Continuous Fourier to the Discrete-time Fourier Transform



Chapter 4
Image Enhancement in the Frequency Domain

From the Continuous Fourier to the Discrete-time Fourier Transform

The frequency domain representation of continuous signals is given by

$$X(\omega) = \int_{-\infty}^{\infty} x(t) e^{-j\omega t} dt$$

If we consider a sampled signal $x_s(t)$, that is

$$x_s(t) = \sum_{n=-\infty}^{\infty} x(nT)\delta(t-nT)$$

then its F.T. is

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

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

4.8

Chapter 4
Image Enhancement in the Frequency Domain

The Discrete-Time Fourier Transform

The F.T can be also written as

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

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

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

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

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

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

4.9

Chapter 4
Image Enhancement in the Frequency Domain

The Inverse Discrete-time Fourier Transform

The Inverse DTFT is

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

A DTFT pair is denoted as

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

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

4.10

Chapter 4
Image Enhancement in the
Frequency Domain

Numerical Computation of the Fourier Transform
The DFT and the FFT

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

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

and the discrete spectrum is given by

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

4.11

Numerical Computation of the Fourier Transform
The DFT and the FFT

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

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

and the discrete spectrum is given by

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

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

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

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

4.12

Chapter 4
Image Enhancement in the Frequency Domain

The DFT Matrix

The DFT and the IDFT may be expressed in terms of matrices, i.e.,

$$\begin{bmatrix} X(0) \\ X(1) \\ X(2) \\ . \\ . \\ X(N-1) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & \zeta^{-1} & \zeta^{-2} & \dots & \zeta^{-(N-1)} \\ 1 & \zeta^{-2} & \zeta^{-4} & \dots & \zeta^{-2(N-1)} \\ . & . & . & \dots & . \\ . & . & . & \dots & . \\ 1 & \zeta^{-(N-1)} & \zeta^{-2(N-1)} & \dots & \zeta^{-(N-1)^2} \end{bmatrix} \begin{bmatrix} x(0) \\ x(1) \\ x(2) \\ . \\ . \\ x(N-1) \end{bmatrix}$$

where $\zeta^{-k} = e^{-j2\pi k/N}$ and $\underline{F}^{-1} = \frac{1}{N}\underline{F}^{H}$

4.13

Selected Properties of the DFT

Linearity: $\{\alpha x(n) + \beta y(n)\} \leftrightarrow \{\alpha X(k) + \beta Y(k)\}$

Shifting: $\{x(n-m) \bmod N\} \leftrightarrow e^{-j2\Pi km/N}\{X(k)\}$

Circular Convolution: $x(n) \otimes h(n) \leftrightarrow X(k)H(k)$

where $x(n) \otimes h(n) = \sum_{m=0}^{N-1} h(m)\, x\left((n-m)_{\mathrm{mod}\,N}\right)$

Freq. Circular Convolution: $x(n)w(n) \leftrightarrow \frac{1}{N} X(k) \otimes W(k)$

Parseval's Theorem: $\sum_{n=0}^{N-1} |x(n)|^2 = \frac{1}{N}\sum_{k=0}^{N-1} |X(k)|^2$

4.14

Chapter 4
Image Enhancement in the Frequency Domain

Notes on the DFT

The DFT transform is an exact one-to-one transform

The DFT can only approximate the continuous Fourier Transform

The DFT components correspond to N frequencies that are fs/N apart

The DFT of a real-valued signal gives symmetric frequency components

A fast algorithm, the FFT, is available for implementing the DFT

The FFT has several applications in spectral analysis, speech analysis-synthesis, fast convolution, etc

4.15

Chapter 4
Image Enhancement in the Frequency Domain

Frequency resolution of the DFT

The frequency resolution of the N-point DFT is

$$f_r = \frac{f_s}{N}$$

- The DFT can resolve exactly only the frequencies falling exactly at: k fs/N. There is spectral leakage for components falling between the DFT bins
- Typically we use an FFT that is as large as we can afford
- Zero-padding is often use to provide more resolution in the frequency components
- Zero padding is often combined with tapered windows

4.16

Chapter 4
Image Enhancement in the Frequency Domain

Spectral Estimates over Finite-time Data windows

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

The DFT, however, estimates the spectrum over finite time

The DFT essentially applies a window to truncate the data.

The simplest data window is the rectangular (boxcar).

Truncation in time is convolution in frequency

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

4.17

Chapter 4
Image Enhancement in the Frequency Domain

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

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

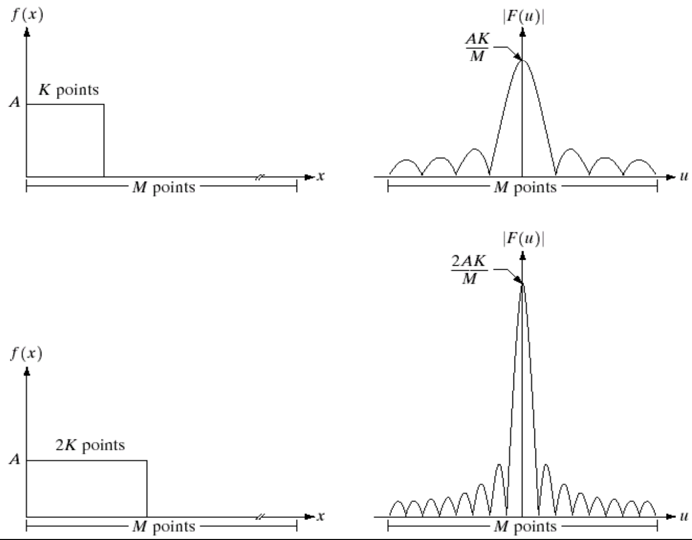
This *discrete Fourier transform* (DFT) is the foundation for most of the work in this chapter. Similarly, given $F(u)$, we can obtain the original function back using the inverse DFT:

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

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

4.18

Chapter 4
Image Enhancement in the Frequency Domain



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.

Chapter 4
Image Enhancement in the Frequency Domain

The concept of the frequency domain, mentioned numerous times in this chapter and in Chapter 3, follows directly from Euler's formula:

$$e^{j\theta} = \cos\theta + j\sin\theta. \tag{4.2-7}$$

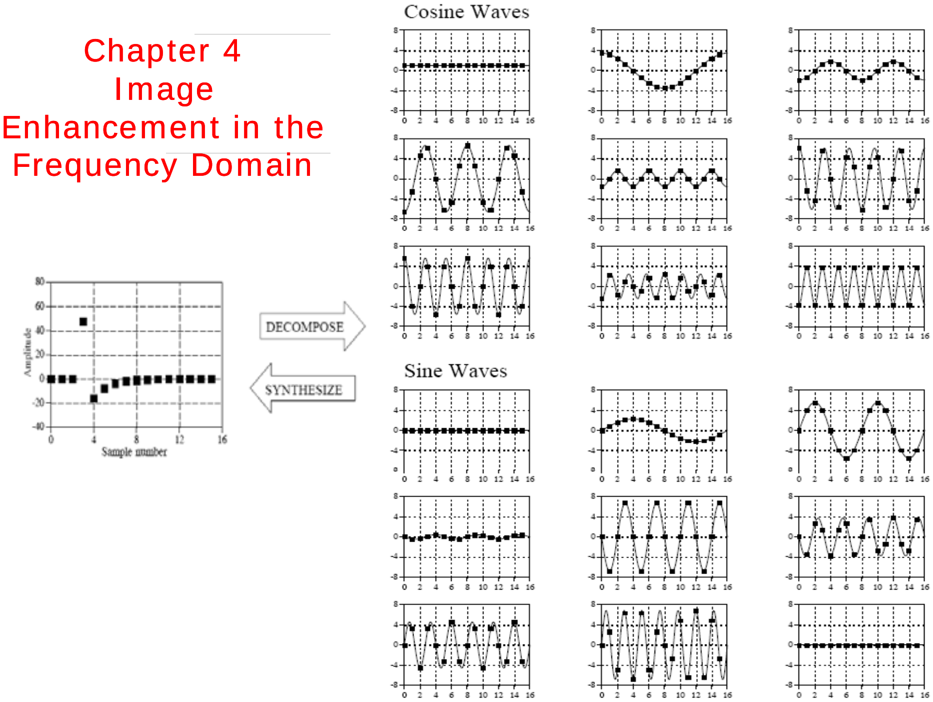
Substituting this expression into Eq. (4.2-5), and using the fact that $\cos(-\theta) = \cos\theta$, gives us

$$F(u) = \frac{1}{M}\sum_{x=0}^{M-1} f(x)[\cos 2\pi ux/M - j\sin 2\pi ux/M] \tag{4.2-8}$$

$$f(x) = \sum_{u=0}^{M/2} F(u)\cos(2\pi ux) + j\sum_{u=0}^{M/2} F(u)\sin(2\pi ux)$$

4.20

Chapter 4
Image Enhancement in the Frequency Domain



Chapter 4
Image Enhancement in the Frequency Domain

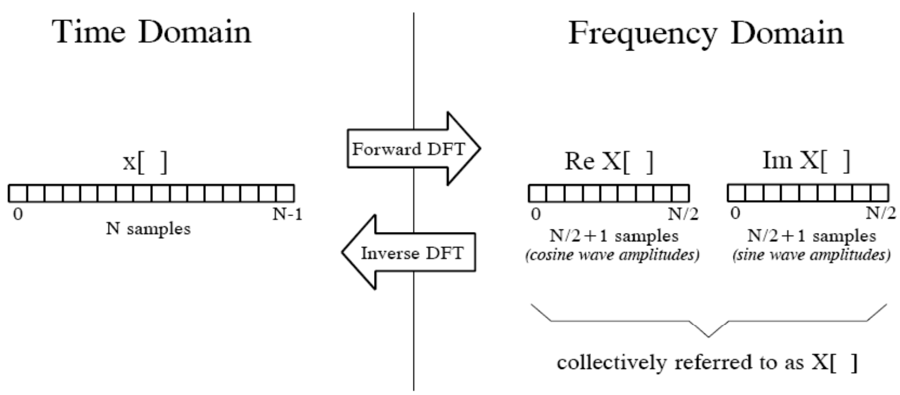
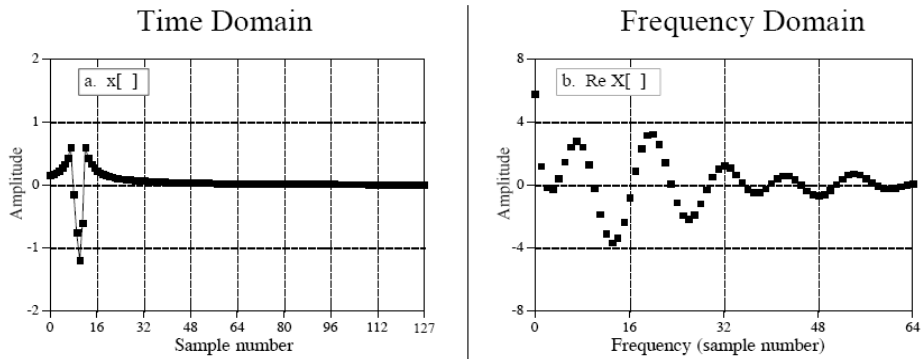


FIGURE 8-3
DFT terminology. In the time domain, $x[\ ]$ consists of N points running from 0 to $N-1$. In the frequency domain, the DFT produces two signals, the real part, written: $Re X[\ ]$, and the imaginary part, written: $Im X[\ ]$. Each of these frequency domain signals are $N/2+1$ points long, and run from 0 to $N/2$. The Forward DFT transforms from the time domain to the frequency domain, while the Inverse DFT transforms from the frequency domain to the time domain. (Take note: this figure describes the **real DFT**. The **complex DFT**, discussed in Chapter 31, changes N complex points into another set of N complex points).



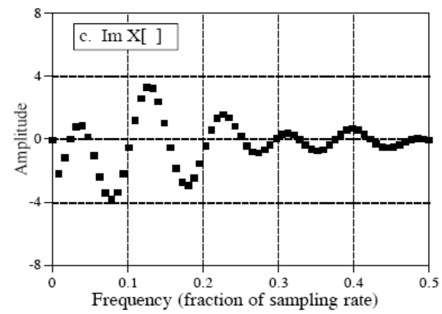


FIGURE 8-4
Example of the DFT. The DFT converts the time domain signal, *x*[ ], into the frequency domain signals, *ReX*[ ] and *Im X*[ ]. The horizontal axis of the frequency domain can be labeled in one of three ways: (1) as an array index that runs between 0 and *N*/2, (2) as a fraction of the sampling frequency, running between 0 and 0.5, (3) as a natural frequency, running between 0 and π. In the example shown here, (b) uses the first method, while (c) use the second method.

Chapter 4
Image Enhancement in the Frequency Domain

In general, we see from Eqs. (4.2-5) or (4.2-8) that the components of the Fourier transform are complex quantities. As in the analysis of complex numbers, we find it convenient sometimes to express $F(u)$ in polar coordinates:

$$F(u) = |F(u)|e^{-j\phi(u)} \tag{4.2-9}$$

where

$$|F(u)| = \left[R^2(u) + I^2(u)\right]^{1/2} \tag{4.2-10}$$

is called the *magnitude* or *spectrum* of the Fourier transform, and

$$\phi(u) = \tan^{-1}\left[\frac{I(u)}{R(u)}\right] \tag{4.2-11}$$

is called the *phase angle* or *phase spectrum* of the transform. In Eqs. (4.2-10) and (4.2-11), $R(u)$ and $I(u)$ are the real and imaginary parts of $F(u)$, respectively. In terms of image enhancement we are concerned primarily with properties of the spectrum. Another quantity that is used later in this chapter is the *power spectrum*, defined as the square of the Fourier spectrum:

$$\begin{aligned} P(u) &= |F(u)|^2 \\ &= R^2(u) + I^2(u). \end{aligned} \tag{4.2-12}$$

24

The term *spectral density* also is used to refer to the power spectrum.

Chapter 4
Image Enhancement in the Frequency Domain

These two equations comprise the *Fourier transform pair*. They indicate the important fact mentioned in the previous section that a function can be recovered from its transform. These equations are easily extended to two variables, u and v:

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

and, similarly for the inverse transform,

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

4.25

Chapter 4
Image Enhancement in the Frequency Domain

Extension of the one-dimensional discrete Fourier transform and its inverse to two dimensions is straightforward. The discrete Fourier transform of a function (image) $f(x,y)$ of size $M \times N$ is given by the equation

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

As in the 1-D case, this expression must be computed for values of $u = 0, 1, 2, \dots, M-1$, and also for $v = 0, 1, 2, \dots, N-1$. Similarly, given $F(u,v)$, we obtain $f(x,y)$ via the *inverse* Fourier transform, given by the expression

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

for $x = 0, 1, 2, \dots, M-1$ and $y = 0, 1, 2, \dots, N-1$. Equations (4.2-16) and (4.2-17) comprise the *two-dimensional, discrete Fourier transform (DFT) pair.*

4.26

Chapter 4
Image Enhancement in the Frequency Domain

$$|F(u,v)| = \left[R^2(u,v) + I^2(u,v)\right]^{1/2} \qquad (4.2\text{-}18)$$

$$\phi(u,v) = \tan^{-1}\left[\frac{I(u,v)}{R(u,v)}\right] \qquad (4.2\text{-}19)$$

and

$$P(u,v) = |F(u,v)|^2 = R^2(u,v) + I^2(u,v) \qquad (4.2\text{-}20)$$

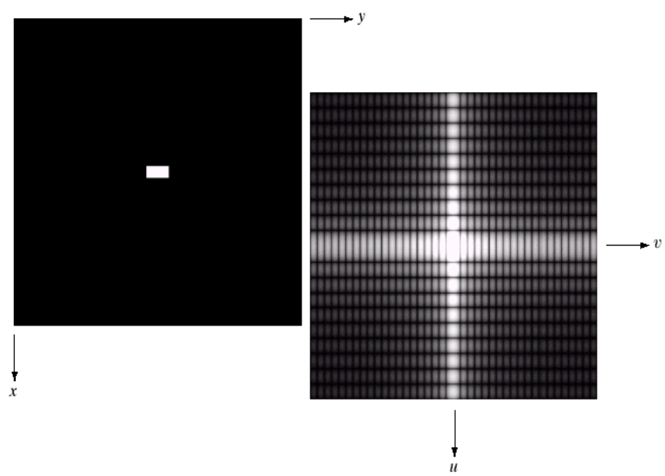
where $R(u,v)$ and $I(u,v)$ are the real and imaginary parts of $F(u,v)$, respectively.

It is common practice to multiply the input image function by $(-1)^{x+y}$ prior to computing the Fourier transform. Due to the properties of exponentials, it is not difficult to show (see Section 4.6) that

$$\Im\left[f(x,y)(-1)^{x+y}\right] = F(u - M/2, v - N/2) \qquad (4.2\text{-}21)$$

4.27

Chapter 4
Image Enhancement in the Frequency Domain

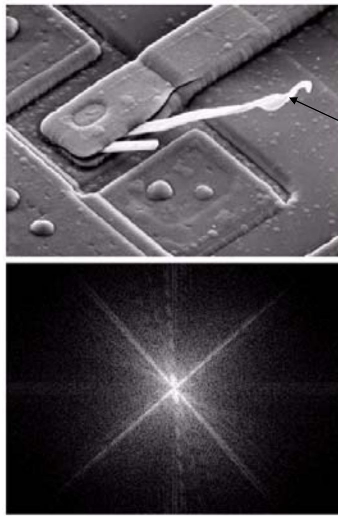


a b

FIGURE 4.3 (a) Image of a 20×40 white rectangle on a black background of size 512×512 pixels. (b) Centered Fourier spectrum shown after application of the log transformation given in Eq. (3.2-2). Compare with Fig. 4.2.

4.28

Chapter 4
Image Enhancement in the Frequency Domain



protrusions

a
b

SEM: scanning electron Microscope

FIGURE 4.4 (a) SEM image of a damaged integrated circuit. (b) Fourier spectrum of (a). (Original image courtesy of Dr. J. M. Hudak, Brockhouse Institute for Materials Research, McMaster University, Hamilton, Ontario, Canada.)

notice the ±45° components and the vertical component which is slightly off-axis to the left! It corresponds to the protrusion caused by thermal failure above.

4.29

Chapter 4
Image Enhancement in the Frequency Domain

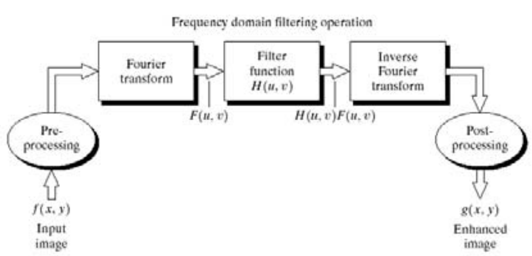


FIGURE 4.5 Basic steps for filtering in the frequency domain.

4.30

Chapter 4
Image Enhancement in the Frequency Domain

Basic Filtering Examples:

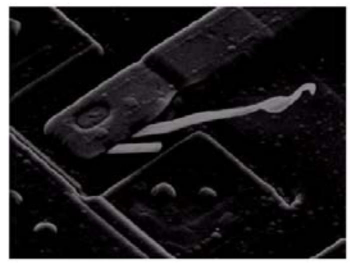
1. Removal of image average

- in time domain?
- in frequency domain: $H(u,v)=\begin{cases}0 & if\ (u,v)=(M/2,N/2)\\ 1 & otherwise\end{cases}$
- the output is:

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

This is called the notch filter, i.e. a constant function with a whole at the origin.

FIGURE 4.6 Result of filtering the image in Fig. 4.4(a) with a notch filter that set to 0 the $F(0,0)$ term in the Fourier transform.



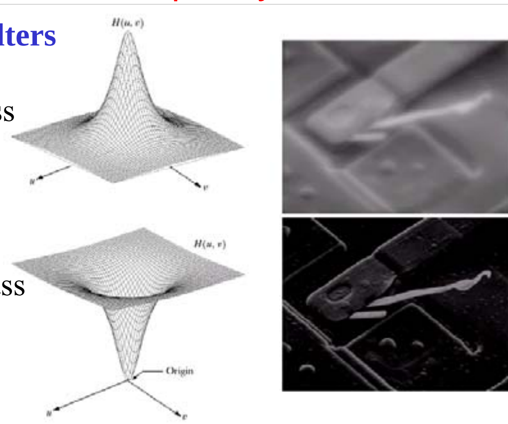
how is this image displayed if the average value is 0?!

Chapter 4
Image Enhancement in the Frequency Domain

2. Linear Filters

2.1. Low-pass

2.2. High-pass

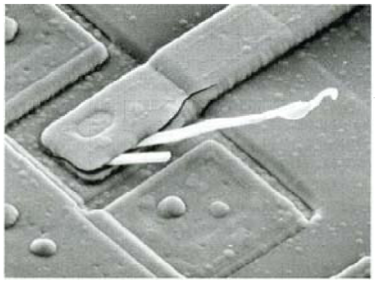


a b
c d

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).

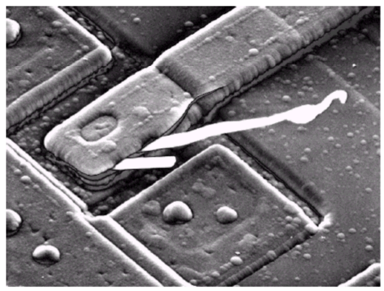
4.32

Chapter 4
Image Enhancement in the Frequency Domain



another result of high-pass filtering where a constant has been added
to the filter so as it will not completely eliminate F(0,0).

FIGURE 4.8 Result of highpass filtering the image in Fig. 4.4(a) with the filter in Fig. 4.7(c), modified by adding a constant of one-half the filter height to the filter function. Compare with Fig. 4.4(a).



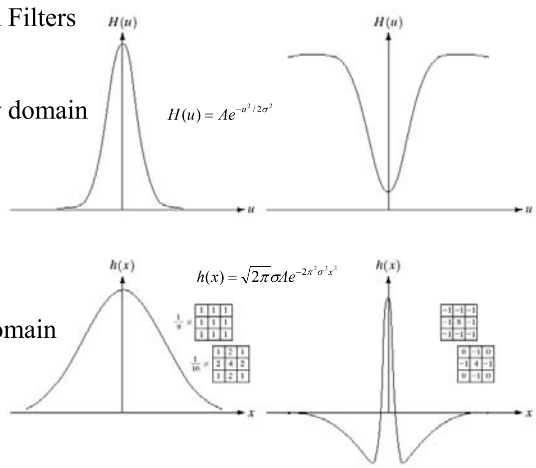
4.33

Chapter 4
Image Enhancement in the Frequency Domain

3. Gaussian Filters

frequency domain

spatial domain



Low-pass high-pass

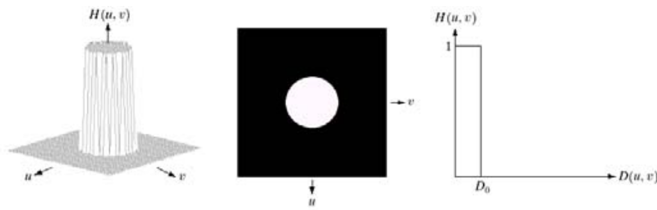
FIGURE 4.9 (a) Gaussian frequency domain lowpass filter. (b) Gaussian frequency domain highpass filter. (c) Corresponding lowpass spatial filter. (d) Corresponding highpass spatial filter. The masks shown are used in Chapter 3 for lowpass and highpass filtering.

4.34

Chapter 4
Image Enhancement in the Frequency Domain

4. Ideal low-pass filter
$$H(u,v)=\begin{cases}1 & if\ D(u,v)\le D_0\\ 0 & if\ D(u,v)\succ D_0\end{cases}$$

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



a b c

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

4.35

Chapter 4
Image Enhancement in the Frequency Domain

a b

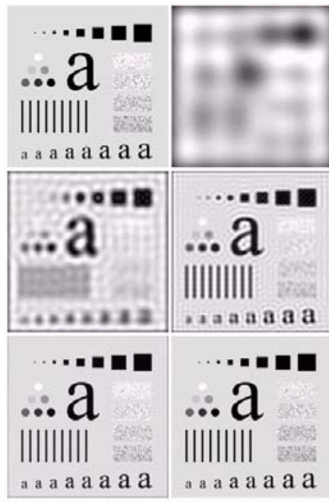
FIGURE 4.11 (a) An image of size 500 × 500 pixels and (b) its Fourier spectrum. The superimposed circles have radii values of 5, 15, 30, 80, and 230, which enclose 92.0, 94.6, 96.4, 98.0, and 99.5% of the image power, respectively.

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

4.36

Chapter 4
Image Enhancement in the Frequency Domain

Note that the narrowrer the filter in the freq. domain is the more severe are the blurring and ringing!



useless, even though only 8% of image power is lost!

notice both blurring and ringing!

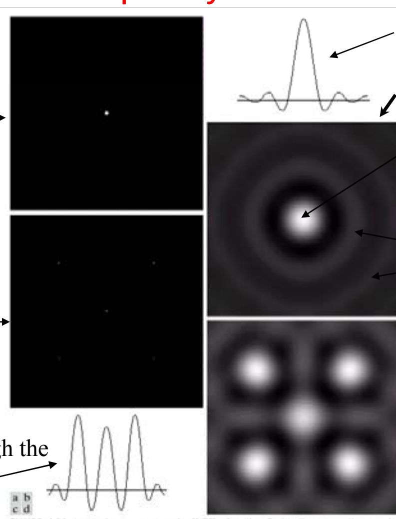
a b
c d
e f
FIGURE 4.12 (a) Original image. (b)–(f) Results of ideal lowpass filtering with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). The power removed by these filters was 8, 5.4, 3.6, 2, and 0.5% of the total, respectively.

4.37

Chapter 4
Image Enhancement in the Frequency Domain

H(u,v) of Ideal Low-Pass Filter (ILPF) with radius 5

input image containing 5 bright impulses



a greylevel profile of a horizontal scan line through the center

h(x,y) is the corresponding spatial filter

the center component is responsible for blurring

the concentric components are responbile for ringing

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

diagonal scan line through the filtered image center

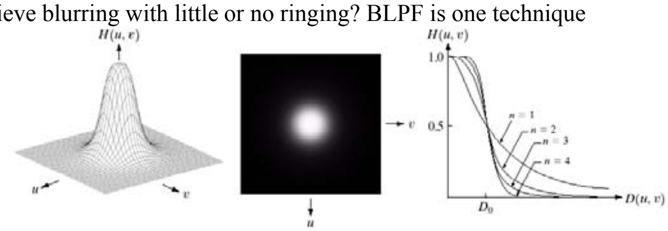
notice blurring and ringing!

a b
c d
FIGURE 4.13 (a) A frequency-domain ILPF of radius 5. (b) Corresponding spatial filter (note the ringing). (c) Five impulses in the spatial domain, simulating the values of five pixels. (d) Convolution of (b) and (c) in the spatial domain.

4.38

Chapter 4
Image Enhancement in the Frequency Domain

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



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.

Transfer function of a BLPF of order n and cut-off frequency at distance D_0 (at which H(u,v) is at ½ its max value) from the origin:

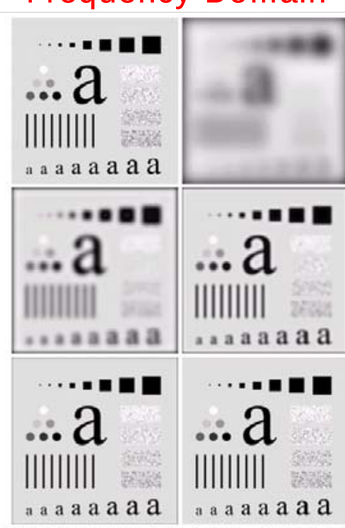
$$H(u,v) = \frac{1}{1+[D(u,v)/D_0]^{2n}} \quad \text{where} \quad D(u,v) = [(u-M/2)^2 + (v-N/2)^2]^{1/2}$$

D(u,v) is just the distance from point *(u,v)* to the center of the FT

4.39

Chapter 4
Image Enhancement in the Frequency Domain

Filtering with BLPF with **n=2** and increasing cut-off as was done with the Ideal LPF



a b c d e f **FIGURE 4.15** (a) Original image. (b)-(f) Results of filtering with BLPFs of order 2, with cutoff frequencies at radii of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Fig. 4.12.

note the smooth transition in blurring achieved as a function of increasing cutoff but no ringing is present in any of the filtered images with this particular BLPF (with n=2)

this is attributed to the smooth transition bet low and high frequencies

4.40

Chapter 4
Image Enhancement in the Frequency Domain

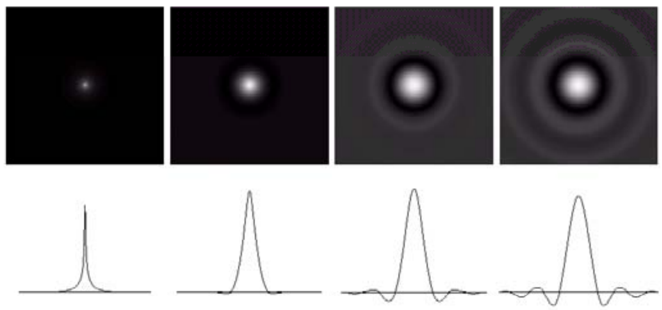


FIGURE 4.16 (a)–(d) Spatial representation of BLPFs of order 1, 2, 5, and 20, and corresponding gray-level profiles through the center of the filters (all filters have a cutoff frequency of 5). Note that ringing increases as a function of filter order.

no ringing for n=1, imperceptible ringing for n=2, ringing increases for higher orders (getting closer to Ideal LPF).

4.41

Chapter 4
Image Enhancement in the Frequency Domain

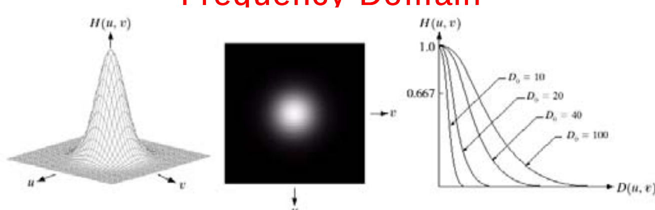


FIGURE 4.17 (a) Perspective plot of a GLPF transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections for various values of D_0.

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

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

σ is a measure of the spread of the Gaussian curve

recall that the inverse FT of the GLPF is also Gaussian, i.e. it has no ringing!

at the cutoff frequency D_0, H(u,v) decreases to 0.607 of its max value.

4.42

Chapter 4
Image Enhancement in the
Frequency Domain

Results of GLPFs

Remarks:

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

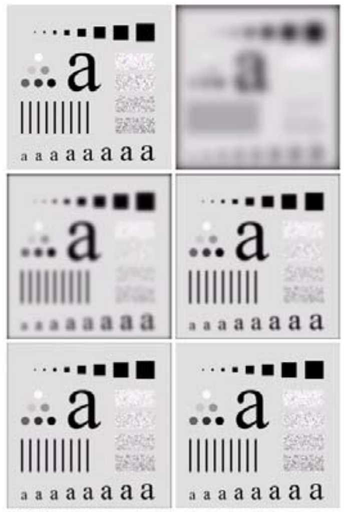


FIGURE 4.18 (a) Original image. (b)–(f) Results of filtering with Gaussian lowpass filters with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Figs. 4.12 and 4.15.

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

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

3. No ringing!

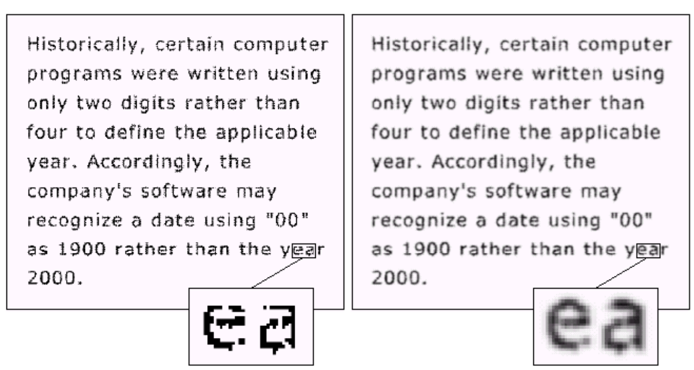
4.43

Chapter 4
Image Enhancement in the
Frequency Domain

Applications: fax transmission, duplicated documents and old records.

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).



GLPF with D_0=80 is used.

4.44

Chapter 4
Image Enhancement in the Frequency Domain

A LPF is also used in printing, e.g. to smooth fine skin lines in faces.

FIGURE 4.20 (a) Original image (1028 × 732 pixels). (b) Result of filtering with a GLPF with $D_0 = 100$. (c) Result of filtering with a GLPF with $D_0 = 80$. Note reduction in skin fine lines in the magnified sections of (b) and (c).

4.45

Chapter 4
Image Enhancement in the Frequency Domain

(a) a very high resolution radiometer (VHRR) image showing part of the Gulf of Mexico (dark) and Florida (light) taken from NOAA satellitle. Note horizontal scan lines caused by sensors.

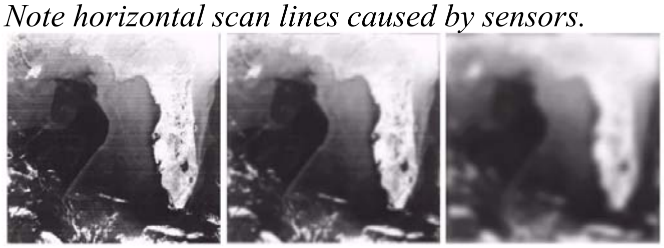


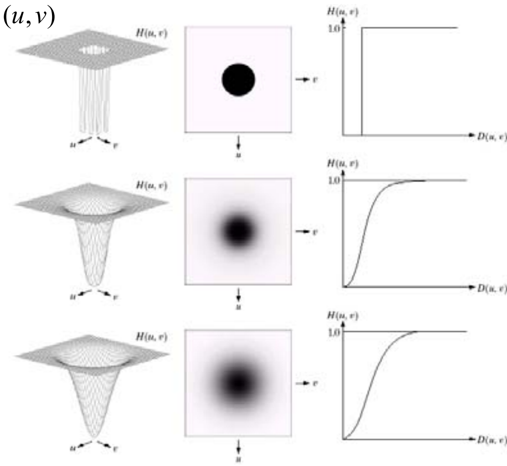
FIGURE 4.21 (a) Image showing prominent scan lines. (b) Result of using a GLPF with $D_0 = 30$. (c) Result of using a GLPF with $D_0 = 10$. (Original image courtesy of NOAA.)

(b) scan lines are removed in smoothed image by a GLP with D_0=30
(c) a large lake in southeast Florida is more visible when more agressive smoothing is applied (GLP with D_0=10).

4.46

Chapter 4
Image Enhancement in the Frequency Domain
Sharpening Frequency Domain Filters

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



ideal high-pass filter

Butterworth high-pass

Gaussian high-pass

a b c
d e f
g h i

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.

4.47

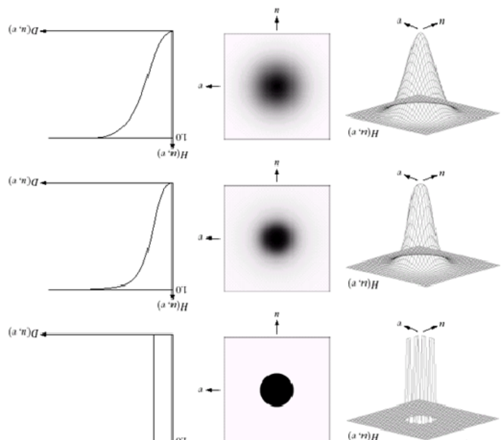


Image Enhancement in the Frequency Domain
Sharpening Frequency Domain Filters

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

ideal high-pass filter

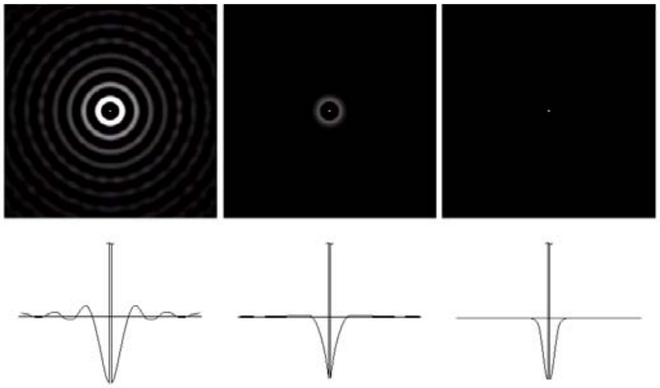
Butterworth high-pass

Gaussian high-pass

4.48

Chapter 4
Image Enhancement in the Frequency Domain

Ideal HPFs are expected to suffer from the same ringing effects as Ideal LPF, see part (a) below

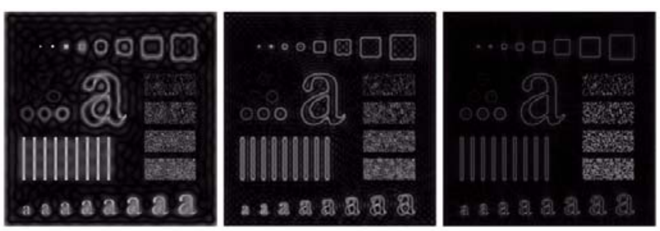


a b c

FIGURE 4.23 Spatial representations of typical (a) ideal, (b) Butterworth, and (c) Gaussian frequency domain highpass filters, and corresponding gray-level profiles.

4.49

Chapter 4
Image Enhancement in the Frequency Domain



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).

Ideal high-pass filters enhance edges but suffer from ringing artefacts, just like Ideal LPF.

4.50

Chapter 4
Image Enhancement in the Frequency Domain

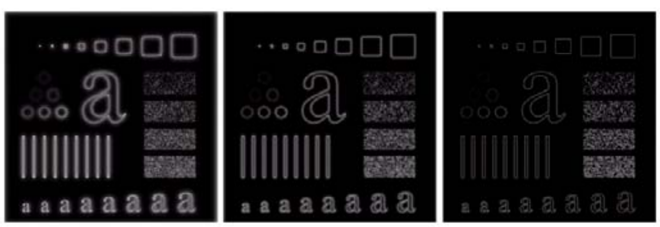
a b c

FIGURE 4.25 Results of highpass filtering the image in Fig. 4.11(a) using a BHPF of order 2 with $D_0 = 15$, 30, and 80, respectively. These results are much smoother than those obtained with an ILPF.

improved enhanced images with BHPFs

4.51

Chapter 4
Image Enhancement in the Frequency Domain

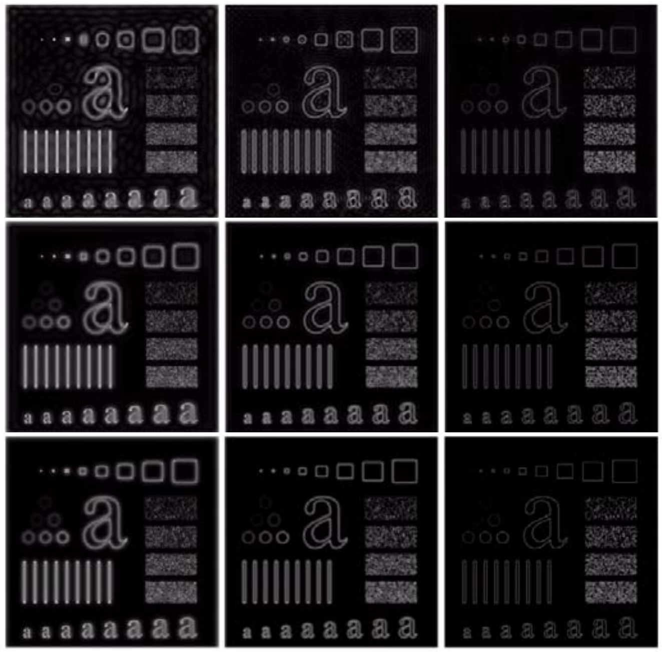


a b c

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.

even smoother results with GHPFs

4.52



Ideal HPF

BHPF

GHPF

4.53

a b c

FIGURE 4.26 Results of highpass filtering the image of Fig. 4.11(a) using a GHPF of order 2 with $D_0 = 15$.

Chapter 4
Image Enhancement in the Frequency Domain

Laplacian in the frequency domain

one can show that:

$$FT\left[\frac{d^n f(x)}{dx^n}\right] = (ju)^n F(u)$$

From this, it follows that:

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

Therefore, the Laplacian can be implemented in frequency by:

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

Recall that F(u,v) is centered if $F(u,v) = FT\left[(-1)^{x+y} f(x,y)\right]$

and thus the center of the filter must be shifted, i.e.

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

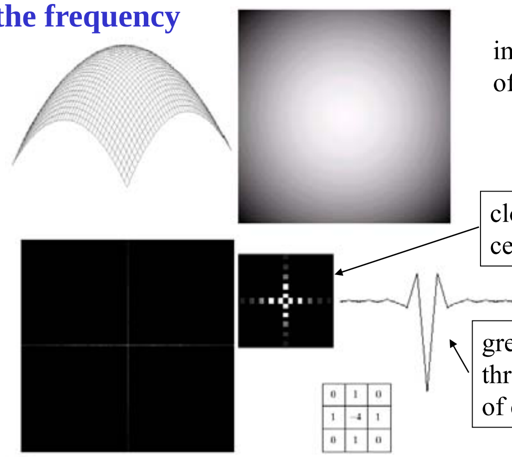
4.54

Chapter 4
Image Enhancement in the Frequency Domain

Laplacian in the frequency domain

H(u,v)

image representation of *H(u,v)*



close-up of the center part

IDFT of image of *H(u,v)*

grey-level profile through the center of close-up

0	1	0
1	−4	1
0	1	0

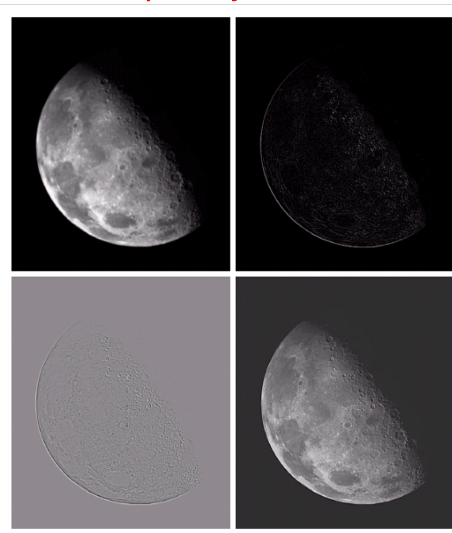
FIGURE 4.27 (a) 3-D plot of Laplacian in the frequency domain. (b) Image representation of (a). (c) Laplacian in the spatial domain obtained from the inverse DFT of (b). (d) Zoomed section of the origin of (c). (e) Gray-level profile through the center of (d). (f) Laplacian mask used in Section 3.7.

4.55

Chapter 4
Image Enhancement in the Frequency Domain

a b
c d

FIGURE 4.28 (a) Image of the North Pole of the moon. (b) Laplacian filtered image. (c) Laplacian image scaled. (d) Image enhanced by using Eq. (4.4-12). (Original image courtesy of NASA.)



result of filtering orig in frequency domain by Laplacian

previous result scaled

enhanced result obtained using

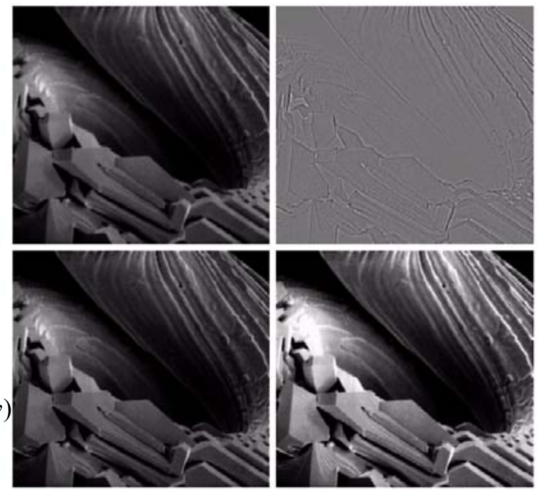
$$g(x,y) = f(x,y) - \nabla^2 f(x,y)$$

4.56

Chapter 4
Image Enhancement in the Frequency Domain: **high-boost filtering**

Scanning electron microscope image of tungsten filament

FIGURE 4.29 Same as Fig. 3.43, but using frequency domain filtering. (a) Input image. (b) Laplacian of (a). (c) Image obtained using Eq. (4.4-17) with $A = 2$. (d) Same as (c), but with $A = 2.7$. (Original image courtesy of Mr. Michael Shaffer, Department of Geological Sciences, University of Oregon, Eugene.)



$A=2$ $\quad$ $A=2.7$

$$f_{hb}(x,y) = (A-1)f(x,y) + f_{hp}(x,y)$$

or in frequency:

$$H_{hb}(u,v) = (A-1) + H_{hp}(u,v)$$

4.57

High Frequency Emphasis Filtering

- How to emphasise more the contribution to enhancement of high-frequency components of an image and still maintain the zero frequency?

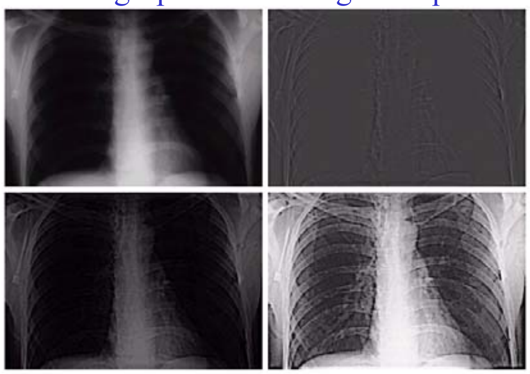
$$H_{hfe}(u,v) = a + bH_{hp}(u,v) \quad where \quad a \geq 0 \ and \ b \succ a$$

- Typical values of a range in 0.25 to 0.5 and b between 1.5 and 2.0.
- When $a=A-1$ and $b=1$ it reduces to high-boost filtering
- When $b>1$, high frequencies are emphasized.

4.58

Chapter 4
Image Enhancement in the Frequency Domain

Butterworth high-pass and histogram equalization



a b
c d

FIGURE 4.30 (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.)

X-ray images cannot be focused in the same manner as a lens, so they tend to produce slightly blurred images with biased (towards black) greylevels → complement freq dom filtering with spatial dom filtering!

4.59

Chapter 4
Image Enhancement in the Frequency Domain: **Homomorphic Filtering**

Recall that the image is formed through the multiplicative illumination-reflectance process:

$$f(x,y) = i(x,y)\,r(x,y)$$

where *i(x,y)* is the illumination and *r(x,y)* is the reflectance component

Question: how can we operate on the frequency components of illumination and reflectance?

Recall that: $FT[f(x,y)] = FT[i(x,y)]\,FT[r(x,y)]$ Correct? ***WRONG!***

Let's make this transformation:

$$z(x,y) = \ln(f(x,y)) = \ln(i(x,y)) + \ln(r(x,y))$$

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

$$Z(u,v) = F_i(u,v) + F_r(u,v)$$

Z(u,v) can then be filtered by a *H(u,v)*, i.e.

$$S(u,v) = H(u,v)Z(u,v) = H(u,v)F_i(u,v) + H(u,v)F_r(u,v)$$

4.60

Chapter 4 Image Enhancement in the Frequency Domain: Homomorphic Filtering

$$s(x,y) = \Im^{-1}\{S(u,v)\} = \Im^{-1}\{H(u,v)F_i(u,v)\} + \Im^{-1}\{H(u,v)F_r(u,v)\}. \tag{4.5-6}$$

By letting

$$i'(x,y) = \Im^{-1}\{H(u,v)F_i(u,v)\} \tag{4.5-7}$$

and

$$r'(x,y) = \Im^{-1}\{H(u,v)F_r(u,v)\}, \tag{4.5-8}$$

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

$$s(x,y) = i'(x,y) + r'(x,y). \tag{4.5-9}$$

Finally, as $z(x,y)$ was formed by taking the logarithm of the original image $f(x,y)$, the inverse (exponential) operation yields the desired enhanced image, denoted by $g(x,y)$; that is,

$$\begin{aligned} g(x,y) &= e^{s(x,y)} \\ &= e^{i'(x,y)} \cdot e^{r'(x,y)} \\ &= i_0(x,y)r_0(x,y) \end{aligned} \tag{4.5-10}$$

where

$$r_0(x,y) = e^{r'(x,y)} \qquad i_0(x,y) = e^{i'(x,y)} \tag{4.5-11}$$

4.61

Chapter 4 Image Enhancement in the Frequency Domain: Homomorphic filtering

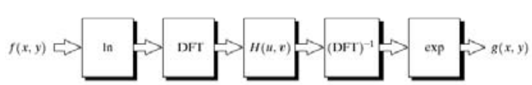


FIGURE 4.31 Homomorphic filtering approach for image enhancement.

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

$$\gamma_L \prec 1 \quad \text{and} \quad \gamma_H \succ 1$$

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

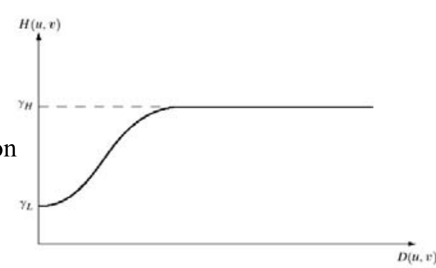


FIGURE 4.32 Cross section of a circularly symmetric filter function. $D(u,v)$ is the distance from the origin of the centered transform.

Net result: simultaneous dynamic range compression and contrast enhancement

4.62

Chapter 4
Image Enhancement in the Frequency Domain

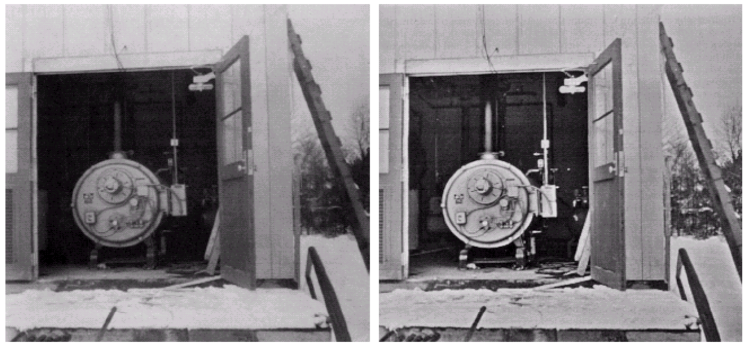


FIGURE 4.33 (a) Original image. (b) Image processed by homomorphic filtering (note details inside shelter). (Stockham.)

$\gamma_L = 0.5$ and $\gamma_H = 2.0$

details of objects inside the shelter which were hidden due to the glare from outside walls are now clearer!

4.63

Chapter 4

Image Enhancement in the Frequency Domain:

Implementation Issues of the FT: origin shifting

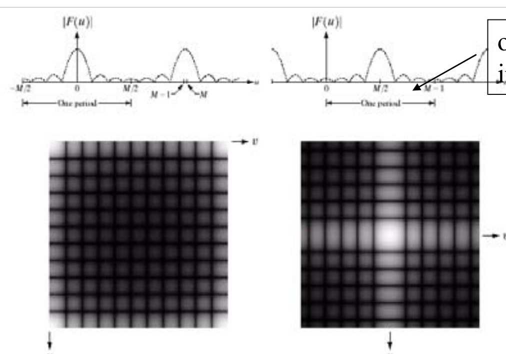


FIGURE 4.34 (a) Fourier spectrum showing back-to-back half periods in the interval $[0, M-1]$. (b) Shifted spectrum showing a full period in the same interval. (c) Fourier spectrum of an image, showing the same back-to-back properties as (a), but in two dimensions. (d) Centered Fourier spectrum.

one full period inside the interval

$$f(x, y)(-1)^{x+y} \Leftrightarrow F(u - M/2, v - N/2) \quad (4.6\text{-}3)$$

$$f(x - M/2, y - N/2) \Leftrightarrow F(u, v)(-1)^{(u+v)}. \quad (4.6\text{-}4)$$

4.64

Chapter 4
Image Enhancement in the Frequency Domain: Scaling

Distributivity and scaling

From the definition of the Fourier transform it follows that

$$\Im[f_1(x,y) + f_2(x,y)] = \Im[f_1(x,y)] + \Im[f_2(x,y)] \qquad (4.6\text{-}5)$$

and, in general, that

$$\Im[f_1(x,y) \cdot f_2(x,y)] \neq \Im[f_1(x,y)] \cdot \Im[f_2(x,y)]. \qquad (4.6\text{-}6)$$

In other words, the Fourier transform is distributive over addition, but not over multiplication. Identical comments apply to the inverse Fourier transform. Similarly, for two scalars a and b,

$$af(x,y) \Leftrightarrow aF(u,v) \qquad (4.6\text{-}7)$$

and

$$f(ax,by) \Leftrightarrow \frac{1}{|ab|}F(u/a, v/b). \qquad (4.6\text{-}8)$$

4.65

Chapter 4
Image Enhancement in the Frequency Domain: Periodicity & Conj. Symmetry

The discrete Fourier transform has the following periodicity properties:

$$F(u,v) = F(u+M,v) = F(u,v+N) = F(u+M,v+N). \qquad (4.6\text{-}10)$$

The inverse transform also is periodic:

$$f(x,y) = f(x+M,y) = f(x,y+N) = f(x+M,y+N). \qquad (4.6\text{-}11)$$

The idea of conjugate symmetry was introduced in Section 4.2, and is repeated here for convenience:

$$F(u,v) = F^*(-u,-v) \qquad (4.6\text{-}12)$$

from which it follows that the spectrum also is symmetric about the origin:

$$|F(u,v)| = |F(-u,-v)|. \qquad (4.6\text{-}13)$$

4.66

Chapter 4
Image Enhancement in the Frequency Domain: separability of 2-D FT

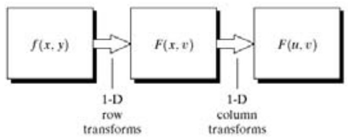


FIGURE 4.35 Computation of the 2-D Fourier transform as a series of 1-D transforms.

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

Separability

The discrete Fourier transform in Eq. (4.2-16) can be expressed in the separable form

$$\begin{aligned} F(u, v) &= \frac{1}{M} \sum_{x=0}^{M-1} e^{-j2\pi ux/M} \frac{1}{N} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi vy/N} \\ &= \frac{1}{M} \sum_{x=0}^{M-1} F(x, v) e^{-j2\pi ux/M} \end{aligned} \tag{4.6-14}$$

where

$$F(x, v) = \frac{1}{N} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi vy/N}. \tag{4.6-15}$$

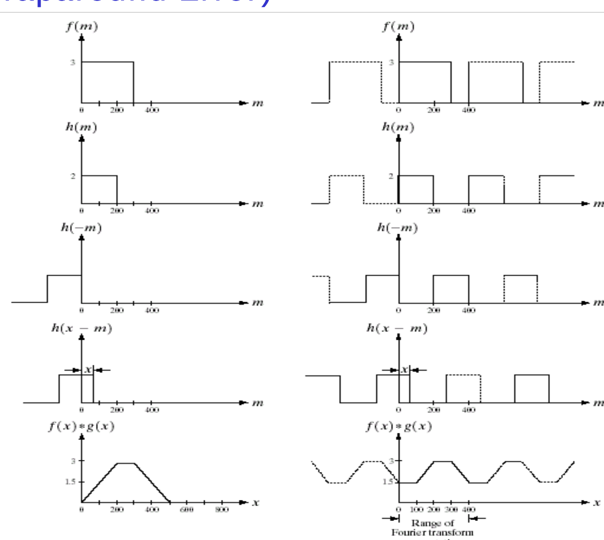
4.67

Chapter 4
Image Enhancement in the Frequency Domain: Convolution (Wraparound Error)

$$f(x) * h(x) = \frac{1}{M} \sum_{m=0}^{M-1} f(m) h(x - m).$$

a f
b g
c h
d i
e j

FIGURE 4.36 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.

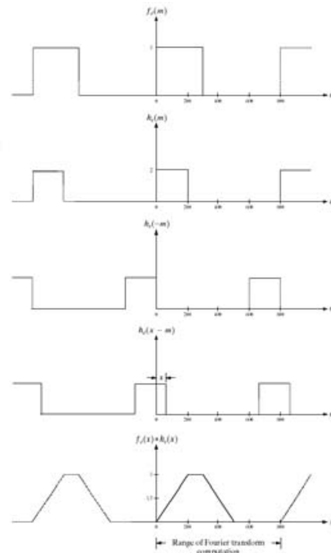


4.68

Chapter 4
Image Enhancement in the Frequency Domain: Convolution (Wraparound Error)

a b c d e

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



The solution to this problem is straightforward. Assume that f and h consist of A and B points, respectively. We append zeros to both functions so that they have identical periods, denoted by P. This procedure yields *extended*, or *padded*, functions given by

$$f_e(x) = \begin{cases} f(x) & 0 \le x \le A-1 \\ 0 & A \le x \le P \end{cases} \qquad (4.6\text{-}21)$$

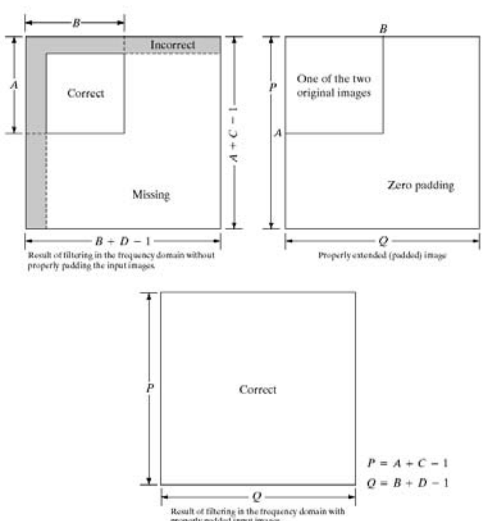
and

$$g_e(x) = \begin{cases} g(x) & 0 \le x \le B-1 \\ 0 & B \le x \le P. \end{cases} \qquad (4.6\text{-}22)$$

It can be shown (Brigham [1988]) that, unless we choose $P \ge A + B - 1$, the individual periods of the convolution will overlap. We already saw in Fig. 4.36 the result of this phenomenon, which is commonly referred to as *wraparound error*. If $P = A + B - 1$, the periods will be adjacent. If $P > A + B - 1$, the periods will be separated, with the degree of separation being equal to the difference between P and $A + B - 1$.

4.69

Chapter 4
Image Enhancement in the Frequency Domain: zero-padding in 2D



a b
c

FIGURE 4.38 Illustration of the need for function padding. (a) Result of performing 2-D convolution without padding. (b) Proper function padding. (c) Correct convolution result.

4.70

Chapter 4
Image Enhancement in the Frequency Domain: zero padding in convolution

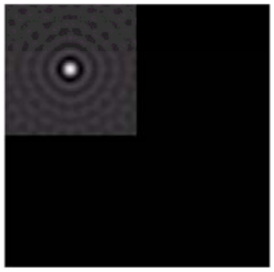


FIGURE 4.39 Padded lowpass filter is the spatial domain (only the real part is shown).

FIGURE 4.40 Result of filtering with padding. The image is usually cropped to its original size since there is little valuable information past the image boundaries.

Chapter 4
Image Enhancement in the Frequency Domain (Convolution & Correlation)

$$f(x, y) * h(x, y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) h(x - m, y - n). \quad (4.6\text{-}27)$$

$$f(x, y) * h(x, y) \Leftrightarrow F(u, v) H(u, v) \quad (4.6\text{-}28)$$

$$f(x, y) h(x, y) \Leftrightarrow F(u, v) * H(u, v). \quad (4.6\text{-}29)$$

4.72

Chapter 4
Image Enhancement in the
Frequency Domain (Convolution & Correlation)

The correlation of two functions $f(x, y)$ and $h(x, y)$ is defined as

$$f(x, y) \circ h(x, y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f^*(m, n) h(x + m, y + n) \quad (4.6\text{-}30)$$

$$f(x, y) \circ h(x, y) \Leftrightarrow F^*(u, v) H(u, v), \quad (4.6\text{-}31)$$

$$f^*(x, y) h(x, y) \Leftrightarrow F(u, v) \circ H(u, v).$$

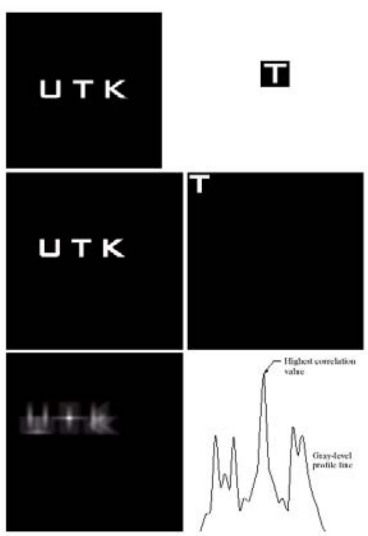
autocorrelation theorem,

$$f(x, y) \circ f(x, y) \Leftrightarrow |F(u, v)|^2. \quad (4.6\text{-}33)$$

$$|f(x, y)|^2 \Leftrightarrow F(u, v) \circ F(u, v). \quad (4.6\text{-}34)$$

4.73

Chapter 4
Image Enhancement in the
Frequency Domain (Matching Filter)



a b
c d
e f

FIGURE 4.41 (a) Image. (b) Template. (c) and (d) Padded images. (e) Correlation function displayed as an image. (f) Horizontal profile line through the highest value in (e), showing the point at which the best match took place.

4.74

Image Transforms

Introduction

- We shall consider mainly two-dimensional transformations.
- Transform theory has played an important role in image processing.
- Image transforms are used for image enhancement, restoration, encoding and analysis.

4.75

Image Transforms

Examples

1. In Fourier Transform,
   a) the average value (or "d.c." term) is proportional to the average image amplitude.
   b) the high frequency terms give an indication of the amplitude and orientation of image edges.
2. In transform coding, the image bandwidth requirement can be reduced by discarding or coarsely quantizing small coefficients.
3. Computational complexity can be reduced, e.g.
   a) perform convolution or compute autocorrelation functions via DFT,
   b) perform DFT using FFT.

4.76

Image Transforms

Unitary Transforms

Recall that

– a matrix A is orthogonal if $A^{-1}=A^T$

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

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

A unitary transformation:

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

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

4.77

Image Transforms

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

Consider an NxN image $u(m,n)$.

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

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

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

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

4.78

Image Transforms

Property 1: orthonormality:

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

Property 2: completeness:

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

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

4.79

Image Transforms

Remark:

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

4.80

Image Transforms

Separable Transforms

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

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

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

4.81

Image Transforms

Properties of Unitary Transforms

1. Energy conservation:

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

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

Therefore, a unitary transformation is simply a rotation!

4.82

Image Transforms

2. Energy compaction:

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

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

is transformed as

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

4.83

Image Transforms

The covariance of v is:

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

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

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

whereas in v:

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

The sum is still 2 (energy conservation), but if ρ=0.95, then

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

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

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

Image Transforms

Conclusions:

- In general, most unitary transforms tend to pack the image energy into few transform coefficients.
- This can be verified by evaluating the following quantities:

If $\mu_u = E[u]$ and $R_u = cov[u]$, then $\mu_v = E[v] = A\mu_u$ and

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

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

Remark:

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

4.85

Image Transforms:

1-D Discrete Fourier Transform (DFT)

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

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

where

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

The inverse transform is given by:

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

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

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

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

and

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

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

Properties of the DFT

a) The N-point DFT can be implemented via FFT in $O(N\log_2 N)$.

b) The DFT of an N-point sequence has N degrees of freedom and requires the same storage capacity as the sequence itself (even though the DFT has $2N$ coefficients, half of them are redundant because of the conjugate symmetry property of the DFT about $N/2$).

c) Circular convolution can be implemented via DFT; the circular convolution of two sequences is equal to the product of their DFTs ($O(N\log_2 N)$ compared with $O(N^2)$).

d) Linear convolution can also be implemented via DFT (by appending zeros to the sequences).

4.88

Image Transforms:

2-D Discrete Fourier Transform (DFT)

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

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

and the inverse transform is given by:

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

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

4.89

Chapter 4 Image Enhancement in the Frequency Domain

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

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

4.90

Chapter 4
Image Enhancement in the Frequency Domain

TABLE 4.1 *(continued)*

Conjugate symmetry	$F(u, v) = F^*(-u, -v)$ $\lvert F(u, v)\rvert = \lvert F(-u, -v)\rvert$
Differentiation	$\frac{\partial^n f(x, y)}{\partial x^n} \Leftrightarrow (ju)^n F(u, v)$ $(-jx)^n f(x, y) \Leftrightarrow \frac{\partial^n F(u, v)}{\partial u^n}$
Laplacian	$\nabla^2 f(x, y) \Leftrightarrow -(u^2 + v^2)F(u, v)$
Distributivity	$\Im[f_1(x, y) + f_2(x, y)] = \Im[f_1(x, y)] + \Im[f_2(x, y)]$ $\Im[f_1(x, y) \cdot f_2(x, y)] \neq \Im[f_1(x, y)] \cdot \Im[f_2(x, y)]$
Scaling	$af(x, y) \Leftrightarrow aF(u, v), f(ax, by) \Leftrightarrow \frac{1}{\lvert ab\rvert} F(u/a, v/b)$
Rotation	$x = r\cos\theta \quad y = r\sin\theta \quad u = \omega\cos\varphi \quad v = \omega\sin\varphi$ $f(r, \theta + \theta_0) \Leftrightarrow F(\omega, \varphi + \theta_0)$
Periodicity	$F(u, v) = F(u + M, v) = F(u, v + N) = F(u + M, v + N)$ $f(x, y) = f(x + M, y) = f(x, y + N) = f(x + M, y + N)$
Separability	See Eqs. (4.6-14) and (4.6-15). Separability implies that we can compute the 2-D transform of an image by first computing 1-D transforms along each row of the image, and then computing a 1-D transform along each column of this intermediate result. The reverse, columns and then rows, yields the same result.

Chapter 4
Image Enhancement in the Frequency Domain

TABLE 4.1 *(continued)*

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†	$f(x, y) \circ h(x, y) \Leftrightarrow F^*(u, v)H(u, v)$; $f^*(x, y)h(x, y) \Leftrightarrow F(u, v) \circ H(u, v)$

Chapter 4 Image Enhancement in the Frequency Domain

TABLE 4.1 *(continued)*

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	$\text{rect}[a, b] \Leftrightarrow ab\dfrac{\sin(\pi ua)}{(\pi ua)}\dfrac{\sin(\pi vb)}{(\pi vb)}e^{-j\pi(ua+vb)}$
Cosine	$\cos(2\pi u_0 x + 2\pi v_0 y) \Leftrightarrow \frac{1}{2}[\delta(u + u_0, v + v_0) + \delta(u - u_0, v - v_0)]$
Sine	$\sin(2\pi u_0 x + 2\pi v_0 y) \Leftrightarrow j\frac{1}{2}[\delta(u + u_0, v + v_0) - \delta(u - u_0, v - v_0)]$

[†] Assumes that functions have been extended by zero padding.

4.93

Chapter 4 Image Enhancement in the Frequency Domain

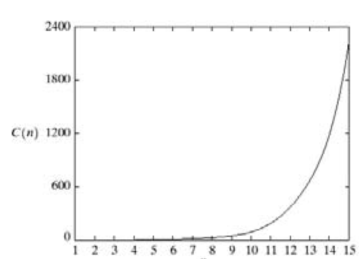


FIGURE 4.42 Computational advantage of the FFT over a direct implementation of the 1-D DFT. Note that the advantage increases rapidly as a function of *n*.

The computational advantage of FFT over direct implementation of the 1-D DFT is defined as: $C(n) = 2^n / n$

Note here that for n=15 ($M=2^n$ long sequence), FFT can be computed nearly 2200 times faster than direct DFT!

4.94

Image Transforms:
Discrete Fourier Transform (DFT)

Drawbacks of FT

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

4.95

Image Transforms:
Cosine and Sine Transforms

- Both are unitary transforms that use sinusoidal

basis functions as does the FT.

- Cosine and sine transforms are **NOT** simply the cosine and sine terms in the FT!

Cosine Transform

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

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

Q: How?

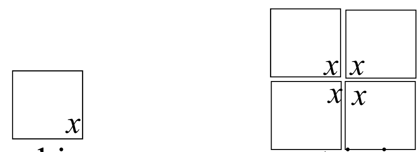
A: Easy!

4.96

Image Transforms:

Discrete Fourier Transform (DFT)

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



original image   symmetric image

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

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

4.97

Image Transforms

Remarks

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

Sine Transform

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

Properties:

- same as the DCT.

4.98

Image Transforms

Hadamard, Haar and Slant Transforms:

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

Hadamard Transform

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

Example:

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

Note that

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

4.99

Image Transforms

How to construct Hadamard matrices?

A: simple!

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

Example:

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

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

4.100

Image Transforms

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

Hadamard Transform Pairs:

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

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

where

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

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

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

4.101

Image Transforms

Properties of Hadamard Transform:

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

4.102

Image Transforms

Haar Transform

is also derived from the (Haar) matrix:

ex:

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

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

4.103

Image Transforms

Properties of the Haar Transform:

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

4.104

Image Transforms

The Slant Transform

is an orthogonal transform designed to possess these properties:

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

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

4.105

Image Transforms

The Karhunen-Loeve Transform (KL)

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

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

General form:

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

4.106

Image Transforms

where the kernel $\Psi(k,l;m,n)$

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

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

where R is the (N^2 x N^2) covariance matrix of the image mapped into an (N^2 x 1) vector and Ψ_i is the i'th column of Ψ

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

Then the KL kernel Ψ is also separable, i.e.,

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

4.107

Image Transforms

Advantage of separability:

- reduce the computational complexity from $O(N^6)$ to $O(N^3)$!

Recall that an NxN eigenvalue problem requires O(N^3) computations.

Properties of the KL Transform

1. Decorrelation: the KL transform coefficients are uncorrelated and have zero mean, i.e.,

$$E[v(k,l)] = 0 \quad for\ all\ k,l;\ and \quad E[v(k,l)v^*(m,n)] = \lambda(k,l)\delta(k-m, l-n)$$

2. It minimizes the mse for any truncated series expansion. Error vanishes in case there is no truncation.

3. Among all unitary transformations, KL packs the maximum average energy in the first few samples of v.

4.108

Image Transforms

Drawbacks of KL:

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

b) it is very computationally intensive.

4.109

Image Transforms

Singular Value Decomposition (SVD):

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

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

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

It is possible to find *r* orthogonal *Mx1* eigenvectors $\{\Phi_m\}$ of U^TU and *r* orthogonal *Nx1* eigenvectors $\{\Psi_m\}$of UU^T, i.e.

$$U^TU\,\Phi_m = \lambda_m\,\Phi_m, \quad m = 1, ..., r$$

and

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

4.110

Image Transforms: SVD Cont'd

The matrix U has the representation:

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

where Ψ and Φ are Nxr and Mxr matrices whose mth columns are the vectors Ψ_m and Φ_m, respectively.

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

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

where v_l are the transform coefficients.

4.111

Image Transforms: SVD Cont'd

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

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

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

Sub-Conclusions:

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

4.112

Image Transforms

Evaluation and Comparison of Different Transforms

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

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

where the variaces have been arranged in decreasing order.

4.113

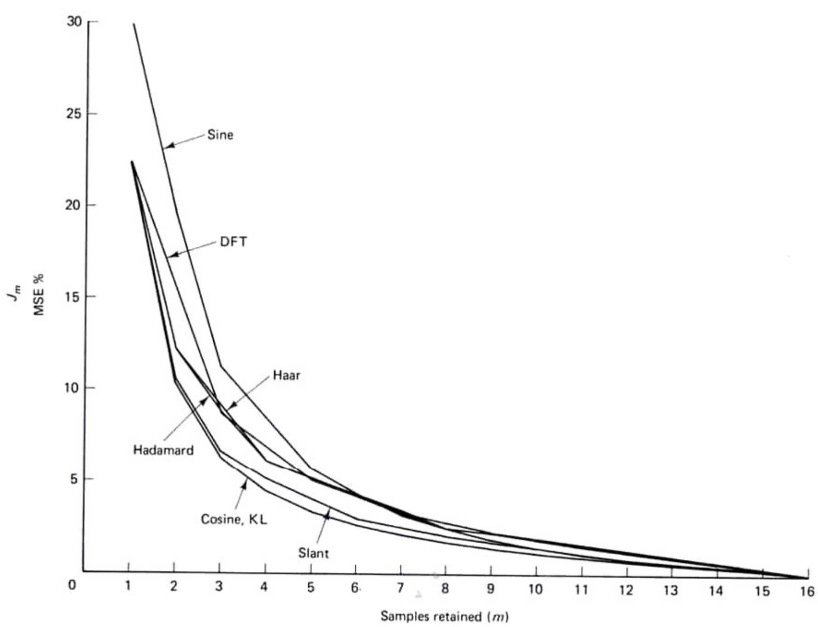


Figure 5.19 Performance of different unitary transforms with respect to basis restriction errors (J_m) versus the number of basis (m) for a stationary Markov sequence with $N = 16$, $\rho = 0.95$.

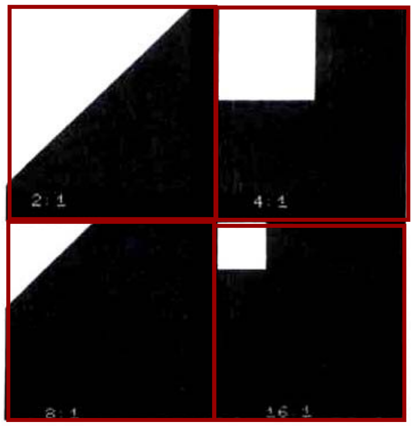
.114

Image Transforms

Evaluation and Comparison of Different Transforms

Zonal Filtering

Zonal Mask:



Define the normalized MSE:

$$J_s = \frac{\sum\limits_{k,l \in stopband}\sum |v_{k,l}|^2}{\sum\limits_{k,l=0}^{N-1}\sum |v_{k,l}|^2} = \frac{energy\ in\ stopband}{total\ energy}$$

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

4.115

Zonal Filtering with DCT transform

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

Figure Basis restriction zonal filtered images in cosine transform domain.

4.116

Basis restriction: Zonal Filtering with different transforms

(a) Cosine; (b) sine;

(c) unitary DFT; (d) Hadamard;

(e) Haar; (f) Slant.

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

4.117

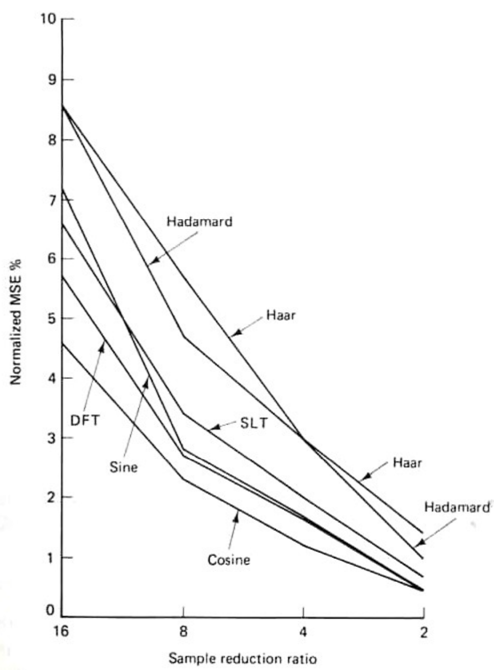


Figure 5.23 Performance comparison of different transforms with respect to basis restriction zonal filtering for 256 × 256 images.

.18

TABLE - Summary of Image Transforms

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

4.119

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

4.120

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

4.121

Image Transforms

Conclusions

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

4.122