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

---

**FIGURE 4.5** Basic steps for filtering in the frequency domain.
Chapter 4
Image Enhancement in the Frequency Domain

<table>
<thead>
<tr>
<th>Type of Transform</th>
<th>Example Signal</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fourier Transform</td>
<td></td>
</tr>
<tr>
<td>signals that are continuous and aperiodic</td>
<td></td>
</tr>
<tr>
<td>Fourier Series</td>
<td></td>
</tr>
<tr>
<td>signals that are continuous and periodic</td>
<td></td>
</tr>
<tr>
<td>Discrete Time Fourier Transform</td>
<td></td>
</tr>
<tr>
<td>signals that are discrete and aperiodic</td>
<td></td>
</tr>
<tr>
<td>Discrete Fourier Transform</td>
<td></td>
</tr>
<tr>
<td>signals that are discrete and periodic</td>
<td></td>
</tr>
</tbody>
</table>

FIGURE 4.2
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.

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

\[
\sum \text{periodic function} = \sum \text{weighted sum of sinusoids}
\]

Fourier, 1807

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

\[
F(u) = \int_{-\infty}^{\infty} f(x) e^{-2\pi i u x} \, dx
\]  

(4.2-1)

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

\[
f(x) = \int_{-\infty}^{\infty} F(u) e^{2\pi i u x} \, du.
\]  

(4.2-2)

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 \( v \).
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

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

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

\[
\Omega = \omega T = 2\pi f_s = \frac{2\pi}{f_s}
\]

and omitting the symbol \( T \) and the subscript \( s \) one can write

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

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 \( 2\pi \). 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 \( \Omega \) at regular intervals
\[
\Omega \Rightarrow \Omega_k = \frac{2\pi}{N} k
\]

and the discrete spectrum is given by
\[
X(e^{j\Omega_k}) = \sum_{n=0}^{N-1} x(n) e^{-j\Omega_k n}
\]

4.11

The inverse Discrete Fourier Transform (IDFT) of the sequence \( x(n) \)
\[
x(n) = \frac{1}{N} \sum_{k=0}^{N-1} X(k) e^{j2\pi kn/N} \quad \text{and} \quad n = 0, 1, \ldots, N-1
\]

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.,

\[
\begin{bmatrix}
    X(0) \\
    X(1) \\
    X(2) \\
    \vdots \\
    X(N-1)
\end{bmatrix} =
\begin{bmatrix}
    1 & 1 & 1 & \ldots & 1 \\
    1 & \zeta & \zeta^2 & \ldots & \zeta^{N-1} \\
    1 & \zeta^2 & \zeta^4 & \ldots & \zeta^{2(N-1)} \\
    \vdots & \vdots & \vdots & \ddots & \vdots \\
    1 & \zeta^{N-1} & \zeta^{2(N-1)} & \ldots & \zeta^{(N-1)(N-1)}
\end{bmatrix}
\begin{bmatrix}
    x(0) \\
    x(1) \\
    x(2) \\
    \vdots \\
    x(N-1)
\end{bmatrix}
\]

where \( \zeta^{-k} = e^{-j2\pi k/N} \) and

\[
F^{-1} = \frac{1}{N} F^H
\]

4.13

Selected Properties of the DFT

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

Shifting:
\[
\{ x(n-m) \mod N \} \leftrightarrow e^{-j2\pi km/N} \{ X(k) \}
\]

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

where
\[
x(n) \otimes h(n) = \sum_{m=0}^{N-1} h(m) x((n-m) \mod N)
\]

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.

---

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 used to provide more resolution in the frequency components.

- Zero padding is often combined with tapered windows.
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

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

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

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:

\[
f(x) = \sum_{u=0}^{M-1} F(u) e^{j2\pi xu/M} \quad \text{for } x = 0, 1, 2, \ldots, M - 1. \tag{4.2-6}
\]

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 \( 1/M \).
4.19

\[ \sum_{m=0}^{M-1} u_x F_j u_x F_x = \frac{2\pi}{M} \]

4.20

\[ f(x) = \sum_{u=0}^{M/2} F(u) \cos(2\pi u x) + j \sum_{u=0}^{M/2} F(u) \sin(2\pi u x) \]
Chapter 4
Image Enhancement in the Frequency Domain

FIGURE 4.3
DFT terminology. In the time domain, X[n] 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[n], and the imaginary part, written: Im X[n]. 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.)
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:

\[ F(u) = |F(u)|e^{-j\phi(u)} \quad (4.2.9) \]

where

\[ |F(u)| = \left( R^2(u) + I^2(u) \right)^{1/2} \quad (4.2.10) \]

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

\[ \phi(u) = \tan^{-1}\left( \frac{I(u)}{R(u)} \right) \quad (4.2.11) \]

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:

\[ P(u) = |F(u)|^2 = R^2(u) + I^2(u) \quad (4.2.12) \]

The term spectral density is also 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 \( v \):

\[
F(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) e^{-j2\pi(ux+vy)} \, dx \, dy \tag{4.2.3}
\]

and similarly for the inverse transform,

\[
f(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} F(u, v) e^{j2\pi(ux+vy)} \, du \, dv. \tag{4.2.4}
\]

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

\[
F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M+vy/N)}. \tag{4.2.16}
\]

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

\[
f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M+vy/N)} \tag{4.2.17}
\]

for \( x = 0, 1, 2, \ldots, M - 1 \) and \( y = 0, 1, 2, \ldots, N - 1 \). Equations (4.2.16) and (4.2.17) comprise the two-dimensional, discrete Fourier transform (DFT) pair.
Chapter 4

Image Enhancement in the Frequency Domain

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

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

and

\[ P(u, v) = |F(u, v)|^2 \]
\[ = R^2(u, v) + I^2(u, v) \] \hspace{1cm} (4.2-20)

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

\[ \Re\left[f(x, y)(-1)^{x+y}\right] = F(u - M/2, v - N/2) \] \hspace{1cm} (4.2-21)

4.27

---

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.
Chapter 4
Image Enhancement in the Frequency Domain

Chapter 4
Image Enhancement in the Frequency Domain

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

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.)

Chapter 4
Image Enhancement in the Frequency Domain

Figure 4.5 Basic steps for filtering in the frequency domain.
Chapter 4
Image Enhancement in the Frequency Domain

Basic Filtering Examples:
1. Removal of image average
   - in time domain?
   - in frequency domain: \( H(u,v) = \begin{cases} 0 & \text{if } (u,v) = (M/2, N/2) \\ 1 & \text{otherwise} \end{cases} \)
   - the output is:
     \[
     G(u,v) = H(u,v) \cdot F(u,v)
     \]

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

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

4.32
another result of high-pass filtering where a constant has been added to the filter so as it will not completely eliminate \( F(0,0) \).
4. Ideal low-pass filter

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

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

- 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)
Chapter 4
Image Enhancement in the Frequency Domain

Note that the narrower 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!

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

input image containing 5 bright impulses

diagonal scan line through the filtered image center

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

notice blurring and ringing!

h(x,y) is the corresponding spatial filter

A greylevel profile of a horizontal scan line through the center

the center component is responsible for blurring

the concentric components are responsible for ringing

4.37

4.38
Transfer function of a BLPF of order \( n \) and cut-off frequency at distance \( D_0 \) (at which \( H(u,v) \) is at \( \frac{1}{2} \) its max value) from the origin:

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

where

\[
D(u,v) = (\frac{u-M}{2})^2 + (\frac{v-N}{2})^2
\]

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

Filtering with BLPF with \( n=2 \) and increasing cut-off as was done with the Ideal LPF 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
no ringing for n=1, imperceptible ringing for n=2, ringing increases for higher orders (getting closer to Ideal LPF).

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

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

$\sigma$ 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.
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.

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

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

3. No ringing!

Applications: fax transmission, duplicated documents and old records.

GLPF with $D_0=80$ is used.
A LPF is also used in printing, e.g. to smooth fine skin lines in faces.

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

(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 aggressive smoothing is applied (GLP with \( D_0 = 10 \)).
Chapter 4
Image Enhancement in the Frequency Domain
Sharpening Frequency Domain Filters

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

- Ideal high-pass filter
- Butterworth high-pass
- Gaussian high-pass

Image Enhancement in the Frequency Domain Filters

$$(\lambda H) \cdot H - 1 = (\lambda H)^{\alpha} \cdot H$$
Ideal HPFs are expected to suffer from the same ringing effects as Ideal LPF, see part (a) below.

![Images of Ideal HPF and LPF effects](image)

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

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

**Figure 4.24** Results of ideal highpass filtering the image in Fig. 4.11(a) with $D_n = 15, 30, \text{ and } 80$, respectively. Problems with ringing are quite evident in (c) and (d).
improved enhanced images with BHPFs

even smoother results with GHPFs
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:

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

Therefore, the Laplacian can be implemented in frequency by:

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

Recall that \( F(u,v) \) is centered if

$$F(u,v) = FT[(-1)^x+y f(x,y)]$$

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

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

grey-level profile through the center of close-up

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

result of filtering orig in frequency domain by Laplacian

previous result scaled

enhanced result obtained using

\[ g(x,y) = f(x,y) - \nabla^2 f(x,y) \]
Chapter 4
Image Enhancement in the Frequency Domain: high-boost filtering

Scanning electron microscope image of tungsten filament

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

or in frequency:

\[ H_{hb}(u, v) = (A - 1) + H_{hp}(u, v) \]

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?

\[ H_{hfe}(u, v) = a + bH_{hp}(u, v) \quad \text{where} \quad a \geq 0 \text{ and } b > a \]

- 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.
Chapter 4  
Image Enhancement in the Frequency Domain

Butterworth high-pass and histogram equalization

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 \( \rightarrow \) complement freq dom filtering with spatial dom filtering!

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

\[
Z(u, v) = F_z(u, v) = F_i(u, v) + F_r(u, v)
\]

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

\[
S(u, v) = H(u, v) Z(u, v) = H(u, v) F_z(u, v) + H(u, v) F_i(u, v) + H(u, v) F_r(u, v)
\]

\[4.60\]
Chapter 4
Image Enhancement in the Frequency Domain: Homomorphic Filtering

\[ s(x, y) = \mathcal{F}^{-1}\{S(u, v)\} = \mathcal{F}^{-1}\{H(u, v)F(u, v)\} + \mathcal{F}^{-1}\{H(u, v)F(u, v)\} \]  (4.5.6)

By letting

\[ f(x, y) = \mathcal{F}^{-1}\{H(u, v)F(u, v)\} \]  (4.5.7)

and

\[ r(x, y) = \mathcal{F}^{-1}\{H(u, v)F(u, v)\} \]  (4.5.8)

Eq. (4.5.6) can be expressed in the form

\[ s(x, y) = f(x, y) + r(x, y). \]  (4.5.9)

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

\[ g(x, y) = e^{r(x, y)} = e^{e^{f(x, y)}}, e^{r(x, y)} = f(x, y) \]  (4.5.10)

where

\[ i_0(x, y) = e^{f(x, y)} \]

\[ i_0(x, y) = s(x, y) \]  (4.5.11)

4.61

Chapter 4
Image Enhancement in the Frequency Domain: Homomorphic filtering

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

\[ \gamma_L < 1 \quad \text{and} \quad \gamma_H > 1 \]

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

Net result: simultaneous dynamic range compression and contrast enhancement 4.62
Chapter 4
Image Enhancement in the Frequency Domain

4.63

\[ \gamma_L = 0.5 \quad \text{and} \quad \gamma_H = 2.0 \]

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

Chapter 4
Image Enhancement in the Frequency Domain:
Implementation Issues of the FT:
origin shifting

4.64

\[ f(x, y)(-1)^{xy} \Leftrightarrow F(u - M/2, v - N/2) \]  
\[ f(x - M/2, y - N/2) \Leftrightarrow F(u, v)(-1)^{xy}. \]
Chapter 4
Image Enhancement in the Frequency Domain: Scaling

Distributivity and scaling
From the definition of the Fourier transform it follows that
\[\mathcal{F}(f_1(x, y) + f_2(x, y)) = \mathcal{F}(f_1(x, y)) + \mathcal{F}(f_2(x, y))\]  \hspace{1cm} (4.6-5)
and, in general, that
\[\mathcal{F}(f_1(x, y) \cdot f_2(x, y)) \neq \mathcal{F}(f_1(x, y)) \cdot \mathcal{F}(f_2(x, y))\]  \hspace{1cm} (4.6-6)
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 \(b\),
\[af(x, y) \leftrightarrow aF(u, v)\]  \hspace{1cm} (4.6-7)
and
\[f(ax, by) \leftrightarrow \frac{1}{|ab|} F(u/a, v/b).\]  \hspace{1cm} (4.6-8)

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

The discrete Fourier transform has the following periodicity properties:
\[F(u, v) = F(u + M, v) = F(u, v + N) = F(u + M, v + N).\]  \hspace{1cm} (4.6-10)
The inverse transform also is periodic:
\[f(x, y) = f(x + M, y) = f(x, y + N) = f(x + M, y + N).\]  \hspace{1cm} (4.6-11)
The idea of conjugate symmetry was introduced in Section 4.2, and is repeated here for convenience:
\[F(u, v) = F^*(-u, -v)\]  \hspace{1cm} (4.6-12)
from which it follows that the spectrum also is symmetric about the origin:
\[|F(u, v)| = |F(-u, -v)|.\]  \hspace{1cm} (4.6-13)
Chapter 4
Image Enhancement in the Frequency Domain: separability of 2-D FT

4.67

FIGURE 4.65
Computation of the 2-D Fourier transform as a series of 1-D transforms.

\[ F(u, v) = \frac{1}{MN} \sum_{y=0}^{N-1} \sum_{x=0}^{M-1} f(x, y) e^{-j2\pi(ax/M + cy/N)} \]

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

\[ F(u, v) = \frac{1}{M} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi ay/N} \]

\[ - \frac{1}{N} \sum_{x=0}^{M-1} F(x, v) e^{-j2\pi bx/M} \]

where

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

4.67

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

4.68

\[ f(x) \ast h(x) = \frac{1}{M} \sum_{m=0}^{M-1} f(m)h(x - m). \]
Chapter 4
Image Enhancement in the Frequency Domain:
Convolution (Wraparound Error)

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

$$f(x) = \begin{cases} f[x] & 0 \leq x < A - 1 \\ 0 & A \leq x < P \\ 0 & P \leq x < 2P \end{cases} \quad (4.6.21)$$

and

$$g(x) = \begin{cases} g[x] & 0 \leq x < B - 1 \\ 0 & B \leq x < 2B \end{cases} \quad (4.6.22)$$

It can be shown (Brigham [1988]) that, unless we choose $P = A + B - 1$, 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 $A + B - 1$.

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

![Diagram](image1.png)

![Diagram](image2.png)
Chapter 4
Image Enhancement in the
Frequency Domain: zero padding in convolution

**Figure 4.39**: Padded lowpass filter in 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)

\[
f(x, y) \ast h(x, y) = \frac{1}{MN} \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} f(m, n) h(x - m, y - n). \tag{4.6-27}
\]

\[
f(x, y) \ast h(x, y) \iff F(u, v) H(u, v) \tag{4.6.28}
\]

\[
f(x, y) h(x, y) \iff F(u, v) \ast H(u, v). \tag{4.6-29}
\]
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

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

(4.6-30)

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

(4.6-31)

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

Autocorrelation theorem,

$$f(x, y) \ast f(x, y) \Leftrightarrow |F(u, v)|^2.$$  

(4.6-33)

$$|f(x, y)|^2 \Leftrightarrow F(u, v) \ast F(u, v).$$  

(4.6-34)

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

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.
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.

---

Image Transforms

In image processing, we deal with 2-D transforms. Consider an $N \times N$ 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 $\{a_{k,l}(m,n)\}$ is a set of complete orthonormal discrete basis functions satisfying the following two properties:
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 \(\{a_{k,l}(m,n)\}\) is the image transform.

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.
Image Transforms

Separable Transforms

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

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

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

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!
Image Transforms

2. Energy compaction:
Example: A zero-mean vector $u=[u(0), u(1)]$ with covariance matrix:

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

is transformed as

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

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 $\rho=0.95$, 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:

\[ \mu_u = \mathbb{E}[u] \text{ and } R_u = \text{cov}[u], \text{ then } \mu_v = \mathbb{E}[v] = A \mu_u \text{ and } \]

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

- 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.

---

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

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

\[ v(k) = \sum_{n=0}^{N-1} u(n) W_N^{kn} \quad k = 0,1,...,N-1 \]

where

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

The inverse transform is given by:

\[ u(n) = \frac{1}{N} \sum_{k=0}^{N-1} v(k) W_N^{-kn} \quad n = 0,1,...,N-1 \]
Image Transforms:
1-D Discrete Fourier Transform (DFT)

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

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

and

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

---

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).
### Image Transforms:
#### 2-D Discrete Fourier Transform (DFT)

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

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

and the inverse transform is given by:

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

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

---

### Chapter 4
#### Image Enhancement in the Frequency Domain

<table>
<thead>
<tr>
<th>Property</th>
<th>Expression(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fourier transform</td>
<td>$F(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-j2\pi(ux/M + vy/N)}$</td>
</tr>
<tr>
<td>Inverse Fourier transform</td>
<td>$f(x, y) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} F(u, v) e^{j2\pi(ux/M + vy/N)}$</td>
</tr>
<tr>
<td>Polar representation</td>
<td>$F(u, v) = |F(u, v)| e^{j\angle F(u, v)}$</td>
</tr>
<tr>
<td>Spectrum</td>
<td>$</td>
</tr>
<tr>
<td>Phase angle</td>
<td>$\phi(u, v) = \tan^{-1} \left( \frac{F(u, v)}{R(U, V)} \right)$</td>
</tr>
<tr>
<td>Power spectrum</td>
<td>$P(u, v) =</td>
</tr>
<tr>
<td>Average value</td>
<td>$f(x, y) = F(0, 0) + \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y)$</td>
</tr>
<tr>
<td>Translation</td>
<td>$f(x, y)e^{j2\pi(ux/M + vy/N)} \leftrightarrow F\left(u - \frac{m}{2}, v - \frac{n}{2}\right)$</td>
</tr>
<tr>
<td></td>
<td>$f(x - \frac{m}{2}, y - \frac{n}{2}) \leftrightarrow F(u, v)e^{-\frac{jm\pi}{M} - \frac{jn\pi}{N}}$</td>
</tr>
</tbody>
</table>

Where $x_u = \frac{m}{2}$ and $y_v = \frac{n}{2}$; then $f(x, y)(-1)^{mn} \leftrightarrow F\left(u - \frac{m}{2}, v - \frac{n}{2}\right)$.
Chapter 4
Image Enhancement in the Frequency Domain

Table 4.1 (continued)

<table>
<thead>
<tr>
<th>Property</th>
<th>Expression(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Conjugation of the inverse Fourier transform using a forward transform algorithm</td>
<td>$F^*(u, v) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} F(u', v') e^{-j2\pi \frac{ux'}{M} \frac{vy'}{N}}$</td>
</tr>
<tr>
<td>Convolution $f(x, y) * h(x, y) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x', y') h(x - x', y - y')$</td>
<td>$f(x, y) * h(x, y) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x', y') h(x - x', y - y')$</td>
</tr>
<tr>
<td>Correlation $f(x, y) \cdot h(x, y) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x', y') h(x - x', y - y')$</td>
<td>$f(x, y) \cdot h(x, y) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x', y') h(x - x', y - y')$</td>
</tr>
<tr>
<td>Convolution theorem $f(x, y) * h(x, y) = F(u, v) H(u, v)$</td>
<td>$f(x, y) * h(x, y) = F(u, v) H(u, v)$</td>
</tr>
<tr>
<td>Correlation theorem $f(x, y) \cdot h(x, y) = F(u, v) H(u, v)$</td>
<td>$f(x, y) \cdot h(x, y) = F(u, v) H(u, v)$</td>
</tr>
</tbody>
</table>
Chapter 4
Image Enhancement in the Frequency Domain

Some useful FT pairs:

<table>
<thead>
<tr>
<th>Function</th>
<th>FT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Impulse</td>
<td>$\delta(x, y) \iff 1$</td>
</tr>
<tr>
<td>Gaussian</td>
<td>$A \sqrt{\frac{M}{2\pi}} e^{-\frac{(x^2+y^2)}{2(M^2)}} \iff A e^{-\frac{(\omega_x^2+\omega_y^2)}{2(M^2)}}$</td>
</tr>
<tr>
<td>Rectangle</td>
<td>$\text{rect}(x, y) \iff e^{-\frac{\omega_x^2}{4(M^2)}} e^{-\frac{\omega_y^2}{4(M^2)}}$</td>
</tr>
<tr>
<td>Cosine</td>
<td>$\cos(2\pi f_x x + 2\pi f_y y) \iff \frac{1}{2} [\delta(f_x - f_0, f_y) + \delta(f_x + f_0, f_y)]$</td>
</tr>
<tr>
<td>Sine</td>
<td>$\sin(2\pi f_x x + 2\pi f_y y) \iff \frac{1}{2} [\delta(f_x - f_0, f_y) - \delta(f_x + f_0, f_y)]$</td>
</tr>
</tbody>
</table>

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

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

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

---

**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!
Image Transforms: Discrete Fourier Transform (DFT)

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

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

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

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.
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
\[ H_2 H_2^T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \]

How to construct Hadamard matrices?
A: simple!

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

\[ k = k_0 + 2k_1 + \cdots + 2^{m-1}k_{m-1} \]

\[ n = n_0 + 2n_1 + \cdots + 2^{n-1}n_{n-1} \]

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.
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.

Properties of the Haar Transform:
• it's real and orthogonal,
• very fast, $O(N)$ for $N$-point sequence!
• it has very poor energy compaction.
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:

\[
S_2 = \frac{1}{2} \begin{bmatrix}
1 & 1 & 1 & 1 \\
3a & a & -a & -3a \\
1 & -1 & -1 & 1 \\
a & -3a & 3a & a \\
\end{bmatrix}
\]

where \( a = \frac{1}{\sqrt{5}} \)

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 Hotelling in 1933 (the discrete version of the KL transform). Also known as Hotelling transform or method of principal component.

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

General form:

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

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 $(N^2 \times N^2)$ covariance matrix of the image mapped into an $(N^2 \times 1)$ vector and $\Psi_i$ is the $i$'th column of $\Psi$

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

Then the KL kernel $\Psi$ is also separable, i.e.,

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

**Image Transforms**

*Advantage of separability:*

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

Recall that an $N\times N$ 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.,

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

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 $v$.  

4.107
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.

---

**Image Transforms**

**Singular Value Decomposition (SVD):**

SVD does for one image exactly what KL does for a set of images. Consider an $N \times N$ image $U$. Let the image be real and $M \leq N$.

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

It is possible to find $r$ orthogonal $M \times 1$ eigenvectors $\{\Phi_m\}$ of $U^TU$ and $r$ orthogonal $N \times 1$ eigenvectors $\{\Psi_m\}$ of $UU^T$, i.e.

\[
U^TU \Phi_m = \lambda_m \Phi_m, \quad m = 1, \ldots, r
\]

and

\[
UU^T \Psi_m = \lambda_m \Psi_m, \quad m = 1, \ldots, r
\]
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 $\Psi$ and $\Phi$ are $N \times r$ and $M \times r$ matrices whose $m$th columns are the vectors $\psi_m$ and $\phi_m$, 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.

---

The energy concentrated in the transform coefficients $v_1, ..., v_k$ 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 $v_1, ..., v_k$, where the average is taken over an ensemble of images for which the autocorrelation function is constant. 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.
Image Transforms

Evaluation and Comparison of Different Transforms

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 \( N=16 \) and correlation coefficient 0.95.

\[
J_m = \frac{\sum_{k=m}^{N-1} \sigma_k^2}{\sum_{k=0}^{N-1} \sigma_k^2}, \quad m = 0, \ldots, N - 1
\]

where the variances have been arranged in decreasing order.

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 \( N = 16, \rho = 0.95 \).
Image Transforms

Evaluation and Comparison of Different Transforms

Zonal Filtering

Zonal Mask:

Define the normalized MSE:

\[
J_z = \frac{\sum_{k,l:_{stopband}} |v_{k,l}|^2}{\sum_{k,l:_{total}} |v_{k,l}|^2} = \frac{\text{energy in stopband}}{\text{total energy}}
\]

Figure 5.20 Zonal filters for 2:1, 4:1, 8:1, 16:1 sample reduction. White areas are passbands, dark areas are stopbands. 1.115

Zonal Filtering with DCT transform

Figure Basis restriction zonal filtered images in cosine transform domain. 4.116
Basis restriction:
Zonal Filtering with different transforms

Figure 4.117 Basis restriction zonal filtering using different transforms with 4:1 sample reduction.

Figure 5.23 Performance comparison of different transforms with respect to basis restriction zonal filtering for 256 x 256 images.
### TABLE - Summary of Image Transforms

<table>
<thead>
<tr>
<th>Transform</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>DFT/unitary DFT</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Cosine</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Sine</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Hadamard</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Haar</strong></td>
<td>Very fast transform. Useful in feature extraction, image coding, and image analysis problems. Energy compaction is fair.</td>
</tr>
<tr>
<td><strong>Slant</strong></td>
<td>Fast transform. Has “image-like basis”; useful in image coding. Has very good energy compaction for images.</td>
</tr>
</tbody>
</table>
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.

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).