Basic (Low-Level) DIP -- Part II

Image Enhancement in the Frequency Domain (Fourier Transform and Wavelet Transform)

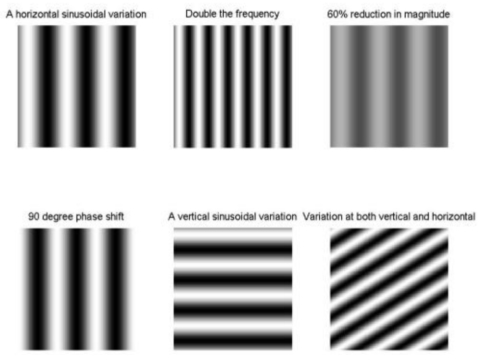
Xiaojun Qi

1

Concepts

- Enhance: To make greater (as in value, desirability, or attractiveness).
- Frequency: The number of times that a periodic function repeats the same sequence of values during a unit variation of the independent variable.
- Frequency Domain Enhancement: Image enhancement is carried out in frequency domain via certain transforms (such as Fourier Transform and Wavelet Transform).

2



Fourier Series

Any function that **periodically repeats** itself can be expressed as the **sum of sines and/or cosines of different frequencies**, each multiplied by a different **coefficient**.

$$f(t) = a_0 + \sum_{n=1}^{\infty} a_n \cos(n f_0 t) + \sum_{n=1}^{\infty} b_n \sin(n f_0 t)$$

where f_0 is called the *fundamental frequency*.

4

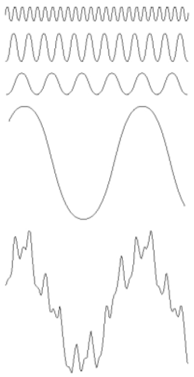


FIGURE 4.1 The function at the bottom is the sum of the four functions above it. Fourier's idea in 1807 that periodic functions could be represented as a weighted sum of sines and cosines was met with skepticism.

5

Fourier Transform

- Any function that is not periodic (but whose area under the curve is finite) can be expressed as the integral of sines and/or cosines multiplied by a weighting function.
- Images can be considered as functions of finite duration. ➔ Therefore, the Fourier transform is the tool in which we are interested.

6

Perfect Reconstruction Feature

- Both Fourier Series and Fourier Transform share the important characteristic that a function, expressed in either a Fourier series or transform, can be reconstructed (recovered) completely via an inverse process, with no loss of information.
- This feature (Perfect Reconstruction Feature) allows us to work in the “Fourier domain” and then return to the original domain of the function without losing any information.

7

1D Continuous Fourier Transform (CFT) and Its Inverse

Let f(x) be a continuous function. Let F(u) represent its Fourier transform. The Fourier transform is defined as:

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

where j = sqrt(-1). The inverse Fourier transform is defined as:

$$f(x) = \int_{-\infty}^{\infty} F(u) e^{j2\pi ux} du$$

8

2D CFT and Its Inverse

Let f(x, y) be a 2D continuous function. Let F(u, v) represent its Fourier transform.

$$F(u,v) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x,y) e^{-j2\pi(ux+vy)} dxdy$$

The inverse Fourier Transform is defined as:

$$f(x,y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} F(u,v) e^{j2\pi(ux+vy)} dudv$$

9

1D Discrete Fourier Transform (DFT)

- Let f(n) be a discrete function, n = 0, 1, 2, …, N-1, resulted from sampling the continuous function f(x) N times at a constant interval delta(x)=1/N. The DFT applied to f(n) is:

$$F(u) = \frac{1}{N}\sum_{n=0}^{N-1} f(n) e^{-j2\pi un/N}$$
$$= \frac{1}{N}\sum_{n=0}^{N-1} f(n)[\cos 2\pi un/N - j\sin 2\pi un/N]$$

  where u =0, 1, 2, …, N-1.
- The domain (values of u) over which the values of F(u) range is appropriately called the ***frequency domain***. Each of the N terms of F(u) is called a ***frequency component*** of the transform.

How many summations and multiplications are needed to compute the DFT?

$$F(u) = R(u) + jI(u)$$

As in the analysis of complex numbers, F(u) can be expressed in polar coordinates:

$$F(u) = |F(u)| e^{j\phi(u)}$$

where |F(u)| is called ***Fourier spectrum*** of f(x) and Φ(u) is the ***phase spectrum***. They are defined as:

$$|F(u)| = [R^2(u) + I^2(u)]^{0.5}$$

$$\phi(u) = \tan^{-1}\frac{I(u)}{R(u)}$$

The square of the Fourier spectrum is called ***power spectrum.***

11

Example – 1D DFT

Given a 1D continuous function, the sampling result is:

f(0.5) = 2; f(0.5 + 1* 0.25) = 3; f(0.5 + 2 * 0.25) = 4; f(0.5 + 3 * 0.25) = 4;

➔ f(0) = 2 ; f(1) = 3 ; f(2) = 4 ; f(3) = 4 ;

$$F(u) = \frac{1}{4}\sum_{n=0}^{3} f(n) e^{-j2\pi un/4}$$

$$= \frac{1}{4}\sum_{n=0}^{3} f(n) e^{-j\pi un/2} = \frac{1}{4}\sum_{n=0}^{3} f(n)[\cos(\pi un/2) - j\sin(\pi un/2)]$$

Are the components of the Fourier transform are complex quantities or the real quantities?

12

$$F(0) = 1/4[f(0)*1 + f(1)*1 + f(2)*1 + f(3)*1]$$
$$= 1/4[2+3+4+4] = 13/4$$
$$F(1) = 1/4\sum_{n=0}^{3} f(n)\,[\cos\frac{n\pi}{2} - j\sin\frac{n\pi}{2}]$$
$$= 1/4[2*1 + 3*(-j) + 4*(-1) + 4*j] = \frac{-2+j}{4}$$
$$F(2) = 1/4\sum_{n=0}^{3} f(n)[\cos n\pi - j\sin n\pi]$$
$$= 1/4[2*1 + 3*(-1) + 4*1 + 4*(-1)] = -1/4$$
$$F(3) = 1/4\sum_{n=0}^{3} f(n)[\cos\frac{3n\pi}{2} - j\sin\frac{3n\pi}{2}]$$
$$= 1/4[2*1 + 3*j + 4*(-1) + 4*(-j)] = \frac{-2-j}{4}$$

13

$$F(0) = 13/4;\ F(1) = \frac{-2+j}{4};$$
$$F(2) = -1/4;\ F(3) = \frac{-2-j}{4}.$$

In the spatial domain, we have $\Delta x = 1/4$;

In the frequency domain, we have $\Delta u = \frac{1}{N\Delta x} = 1.$

It means that : Magnitude, Phase

$$F(0+0*\Delta u) = F(0) = 13/4; \Rightarrow 13/4,\quad 0^\circ$$
$$F(0+1*\Delta u) = F(1) = \frac{-2+j}{4}; \Rightarrow \frac{\sqrt{5}}{4},\quad -26^\circ$$
$$F(0+2*\Delta u) = F(2) = -1/4; \Rightarrow 1/4,\quad 0^\circ$$
$$F(0+3*\Delta u) = F(3) = \frac{-2-j}{4}. \Rightarrow \frac{\sqrt{5}}{4},\quad 26^\circ$$

14

Inverse 1D DFT

- At a particular location n, f(n) can be reconstructed (or approximated) from the weighted sum of a basis function at different frequencies (u = 0, 1, …, N-1).
- The weight is |F(u)|, the magnitude of a frequency component and the basis is sinusoid wave at different frequencies.

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

15

$$f(0) = \sum_{u=0}^{3} F(u)[\cos 0 + j\sin 0]$$
$$= 1/4[F(0)*1 + F(1)*1 + F(2)*1 + F(3)*1]$$
$$= 1/4[13 - 2 + j - 1 - 2 - j] = 2$$
$$f(1) = 1/4\sum_{u=0}^{3} F(u)\,[\cos\frac{u\pi}{2} + j\sin\frac{u\pi}{2}]$$
$$= 1/4[13*1 + (-2+j)*j + (-1)*(-1) + (-2-j)*(-j)] = 3$$
$$f(2) = 1/4\sum_{u=0}^{3} f(u)[\cos u\pi + j\sin u\pi]$$
$$= 1/4[13*1 + (-2+j)*(-1) + (-1)*1 + (-2-j)*(-1)] = 4$$
$$f(3) = 1/4\sum_{u=0}^{3} f(u)[\cos\frac{3u\pi}{2} + j\sin\frac{3u\pi}{2}]$$
$$= 1/4[13*1 + (-2+j)*(-j) + (-1)*(-1) + (-2-j)*j] = 4$$

16

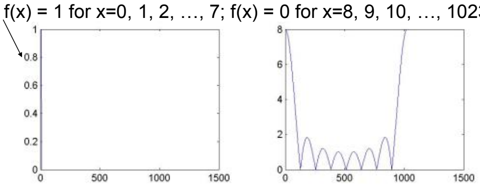
The following is the result obtained from using Matlab

```
% A represents the 1D Discrete Signal
A= [2 3 4 4] ;
% Call the DFT of A
B = fft(A)
B =
13.0000        -2.0000 + 1.0000i
-1.0000        -2.0000 - 1.0000i
C = ifft(B)
C =
    2    3    4    4
```

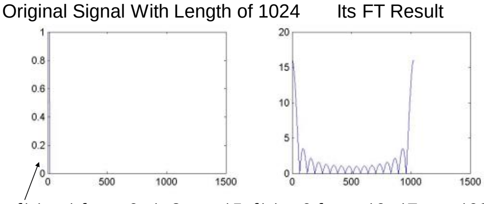
What is the difference between our calculated result and Matlab's result?

17

f(x) = 1 for x=0, 1, 2, …, 7; f(x) = 0 for x=8, 9, 10, …, 1023

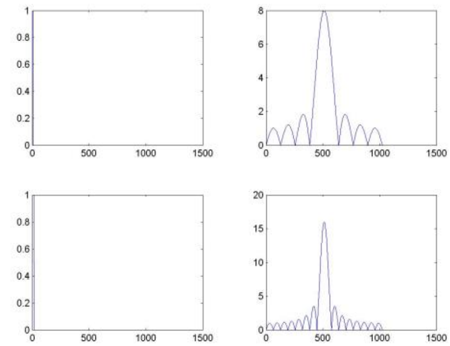


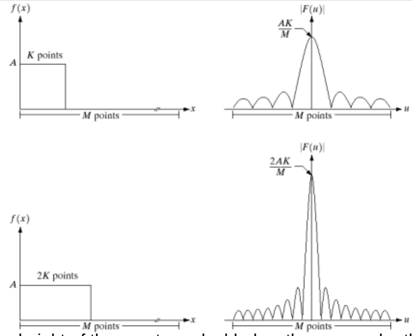
Original Signal With Length of 1024 Its FT Result



f(x) = 1 for x=0, 1, 2, …, 15; f(x) = 0 for x=16, 17, …, 1023

Shift the frequency component to center at zero by using Matlab:
fout = fft(fin) ; fout = fftshift(fout) ;





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.

$$\Delta u = \frac{1}{N \Delta x}$$

The height of the spectrum doubled as the area under the curve in the x-domain doubled.

The number of zeros in the spectrum in the same interval doubled as the length of the function doubled.

Another Way to Get 1D DFT

The DFT can be obtained from sampling the CFT at discrete frequency intervals

$k = 0, \Delta u, 2\Delta u, ..., (N-1)\Delta u$. where $\Delta u = \frac{1}{N \Delta x}$.

That is:

F(k) can be obtained by sampling F(u) at

$$k = 0, \Delta u, 2\Delta u, ..., (N-1)\Delta u.$$

2D DFT

Let f(n,m) be a discrete 2D function resulted from uniformly sampling the continuous function f(x,y) at intervals delta(x) and delta(y). The DFT of f(n,m) is defined as:

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

where u =0, 1, 2, …, N-1 and v = 0, 1, 2, …, M-1.

$$\Delta u = \frac{1}{N \Delta x} \qquad \Delta v = \frac{1}{M \Delta y}$$

Similar to 1D DFT notation, 2D DFT can also be represented in the following notation:

$$F(u,v) = R(u,v) + jI(u,v)$$

$$F(u,v) = |F(u,v)| e^{j\phi(u,v)}$$

where |F(u,v)| is called *Fourier spectrum* of f(x,y) and Φ(u,v) is the *phase spectrum*. They are defined as:

$$|F(u,v)| = [R^2(u,v) + I^2(u,v)]^{0.5}$$

$$\phi(u,v) = \tan^{-1} \frac{I(u,v)}{R(u,v)}$$

Example – 2D DFT

A = [3 6 9 10 ; 2 4 9 0 ; 10 3 9 8 ; 20 4 8 14] ;

Results from 1D DFT applied on each column, **B = fft(A)**

B =

35.00	17.00	35.00	32.00
-7.00 +18.00i	3.00	0 - 1.00i	2.00 +14.00i
-9.00	1.00	1.00	4.00
-7.00 -18.00i	3.00	0 + 1.00i	2.00 -14.00i

Results from 1D DFT applied on each row, **C=(fft(B'))'**

C =

1.0e+002 *

1.19	0 + 0.15i	0.21	0 - 0.15i
-0.02 + 0.31i	-0.21 + 0.18i	-0.12 + 0.03i	0.07 + 0.20i
-0.03	-0.10 + 0.03i	-0.13(Center)	-0.10 - 0.03i
-0.02 - 0.31i	0.07 - 0.20i	-0.12 - 0.03i	-0.21 - 0.18i

A = [3 6 9 10 ; 2 4 9 0 ;   10 3 9 8 ;   20 4 8 14] ;

Results from 2D DFT by calling Matlab function: **B = fft2(A)**

1.0e+002 *

1.19	0 + 0.15i	0.21	0 - 0.15i
-0.02 + 0.31i	-0.21 + 0.18i	-0.12 + 0.03i	0.07 + 0.20i
-0.03	-0.10 + 0.03i	-0.13	-0.10 - 0.03i
-0.02 - 0.31i	0.07 - 0.20i	-0.12 - 0.03i	-0.21 - 0.18i

Center frequency (0,0) by calling Matlab function: **fftshift(B)**

1.0e+002 *

-0.13	-0.10 - 0.03i	-0.03	-0.10 + 0.03i
-0.12 - 0.03i	-0.21 - 0.18i	-0.02 - 0.31i	0.07 - 0.20i
0.21	0 - 0.15i	1.19	0 + 0.15i
-0.12 + 0.03i	0.07 + 0.20i	-0.02 + 0.31i	-0.21 + 0.18i

25

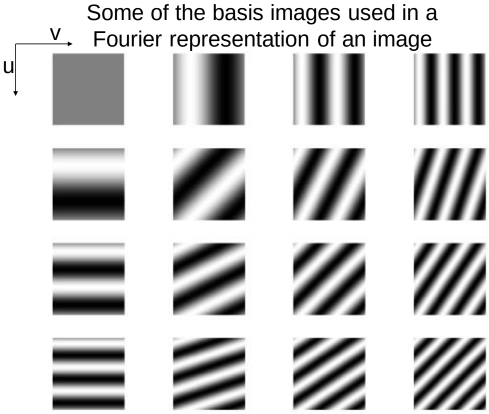
- The value of the transform at (u, v) = (0, 0) is:

$$F(0,0) = \frac{1}{MN}\sum_{n=0}^{N-1}\sum_{m=0}^{M-1} f(n,m)$$

- If f(x, y) is an image, the value of the Fourier transform at the origin is equal to the average gray level of the image.
- F(0, 0) is called **dc (i.e., direct current) component of the spectrum**.

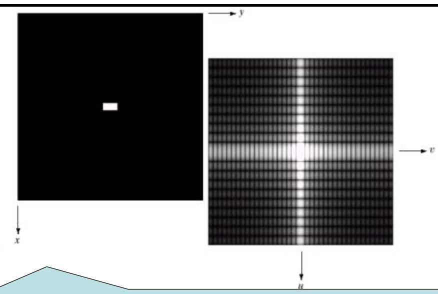
26

Some of the basis images used in a Fourier representation of an image

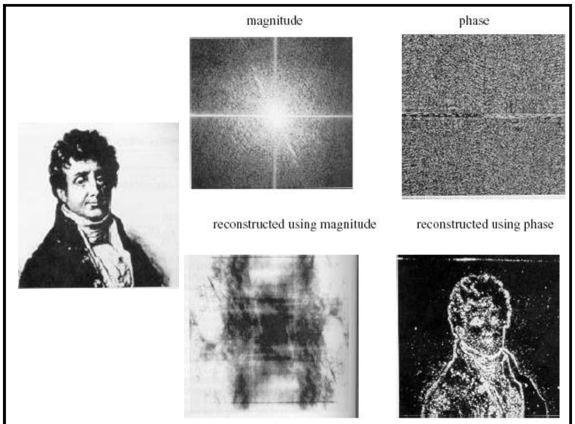


a b

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.



1. What is the origin in the spatial domain?
2. What is the origin in the frequency domain?
3. What is the difference between the number of zeros in the spectrum of u and v at the same interval?
4. Which features in the spatial domain leads to this difference?



Which is more important? Fourier Spectrum (Magnitude) or Phase Spectrum?

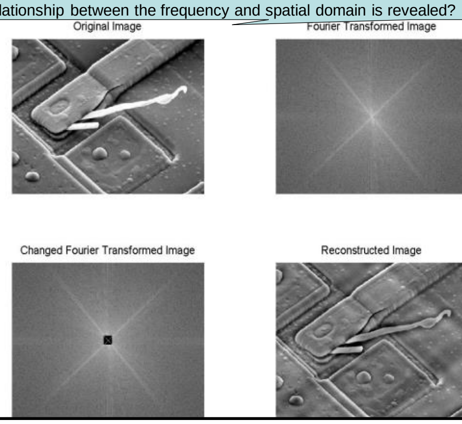
- It seems that the magnitude spectrum contains all the useful information, whereas the phase spectrum appears to be somewhat random and noise.

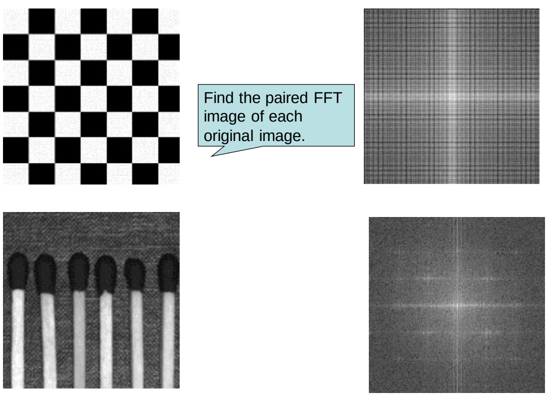
30

- Magnitude determines the contribution of each sinusoidal component
- Phase determines where each of the sinusoidal components resides. That is, it encodes the location of the frequency in the image.
- ➔ Without phase information, the spatial coherence of the image is disrupted and it becomes impossible to recognize feature of interest;
- ➔ Without magnitude information, we can no longer determine the relative brightness of those features, but we can at least see the boundaries between them, which aids recognition.

31

What relationship between the frequency and spatial domain is revealed?





Inverse 2D DFT

- At a particular location (n, m), f(n, m) can be reconstructed (or approximated) from the weighted sum of a basis function at different frequencies (u = 0, 1, …, N-1, and v = 0, 1, …, M-1).
- The weight is |F(u, v)|, the magnitude of a frequency component and the basis is sinusoid wave at different frequencies.

$$f(n,m) = \sum_{u=0}^{N-1}\sum_{v=0}^{M-1} F(u,v)e^{j2\pi(un/N+vm/M)}$$

34

Another Way to Get 2D DFT

The 2D DFT can be obtained from sampling 2D CFT at intervals

$$u = 0,\ \Delta u, 2\Delta u, \ldots, (N-1)\Delta u.$$

and

$$v = 0,\ \Delta v, 2\Delta v, \ldots, (M-1)\Delta v.$$

That is:

F(k1, k2) can be obtained by sampling F(u, v).

35

DFT Properties

- Separability
- Translation
- Periodicity and Conjugate Symmetry
- Rotation
- Distributivity and Scaling
- Convolution and Correlation

36

Separability

- Since Fourier transform involves orthogonal basis, DFT is separable so that it can be written as a separable transform in n and m respectively.

$$F(u,v) = \frac{1}{N}\sum_{n=0}^{N-1}\left[\frac{1}{M}\sum_{m=0}^{M-1} f(n,m)e^{-j2\pi\frac{vm}{M}}\right]e^{-j2\pi\frac{un}{N}}$$

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

37

Translation

$$F(u-u_0, v-v_0) \Leftrightarrow f(n,m)e^{2\pi j(u_0 n/N + v_0 m/M)}$$

$$f(n-n_0, m-m_0) \Leftrightarrow F(u,v)e^{-2\pi j(un_0/N + vm_0/M)}$$

Translation by (n0, m0) in spatial domain does not affect Fourier spectrum but leads to a phase shift in frequency domain.

1. What is the corresponding spatial function after translating in frequency domain by *u0 =N/2* and *v0 = M/2*?
2. What is the new phase angle in frequency domain after translating in spatial domain by (n0, m0)?

Periodicity and Conjugate Symmetry

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

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

To observe a complete period, the origin of the transform is usually shifted to the center point (N/2, M/2) by multiplying f(n, m) by $(-1)^{n+m}$.

39

Rotation

- Rotating f(n, m) leads to a rotation of F(u, v) by the same angle.

$$x = r\cos\theta,\ y = r\sin\theta,$$

$$u = w\cos\varphi,\ v = w\sin\varphi$$

$$f(r, \theta+\theta_0) \Leftrightarrow F(w, \varphi+\theta_0)$$

40

Distributivity and Scaling

$$F(f_1(n,m) + f_2(n,m)) \Leftrightarrow F(f_1(n,m)) + F(f_2(n,m))$$

$$af(n,m) \Leftrightarrow aF(u,v)$$

$$f(an, bm) \Leftrightarrow \frac{1}{|ab|}F(u/a, v/b)$$

41

Convolution Theorem

$$f(n,m) * h(n,m) \Leftrightarrow F(u,v)H(u,v)$$

$$f(n,m)h(n,m) \Leftrightarrow F(u,v) * H(u,v)$$

Using the convolution theorem, convolution in spatial domain can be achieved in frequency domain.

To convolve f(n,m) with h(n,m), do the following:

1. Take DFT on f(n, m) and h(n, m), yielding F(u, v) and H(u, v)
2. Calculate G(u, v) = F(u, v) H(u,v)
3. Perform inverse DFT on G(u, v), yielding g(n, m)

42

Summary

- **An Image f in the Spatial Domain:**

1. (n, m) represents the spatial coordinates.
2. f(n, m) represents intensity at (n, m).

- **An Image f in the Frequency Domain:**

1. (u, v) is the spatial frequency that represents intensity change with respect to spatial distance in a particular direction.
2. F(u, v) is composed of the magnitude and phase of the spatial frequency at (u, v).
3. Low frequencies in the Fourier transform are responsible for the general gray-level appearance of an image over smooth areas, while high frequencies are responsible for detail, such as edges and noise.

43

Filtering and Enhancement in the Frequency Domain

- FT transforms images from the spatial domain to the frequency domain.
- The usefulness of the FT
  - Remove undesirable frequencies from a signal
  - Easier and faster to perform certain operations in the frequency domain than in the spatial domain.
- Enhancement in frequency domain is performed via frequency filtering.
  - Low-pass filtering: Attenuates high frequencies while "passing" low frequencies.
  - High-pass filtering: Attenuates low frequencies while "passing" high frequencies.

44

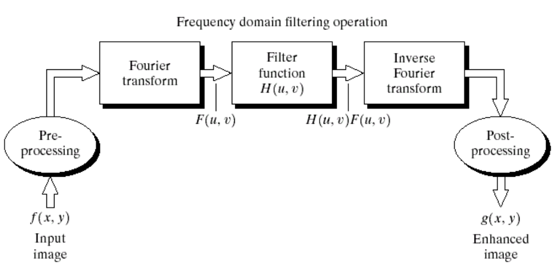


FIGURE 4.5 Basic steps for filtering in the frequency domain.

The important point here is that the filtering process is based on modifying the transform of an image in some way via a filter function, and then taking the inverse of the result to obtain the processed output image.

45

Steps for Filtering in the Frequency Domain

1. Multiply the input image by $(-1)^{x+y}$ to center the transform
2. Compute F(u,v), the DFT of the image from 1.
3. Pixelwisely multiply F(u,v) by a filter function H(u,v).
4. Compute the inverse DFT of the result in 3.
5. Obtain the real part of the result in 4.
6. Multiply the result in 5 by $(-1)^{x+y}$.

46

Given the following:

1. An image f(n, m);
2. Its DFT F(u, v);
3. A filter h(n, m);
4. Its DFT H(u, v). H(u, v) is called a filter (The term filter transfer function also is used commonly). It suppresses certain frequencies in the transform while leaving others unchanged.

Filtering in frequency domain can be defined as

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

where G(u, v) is a filtered image in frequency domain. The filtered image in spatial domain can be obtained via an inverse DFT on G(u, v).

47

Correspondence Between Filtering in the Spatial and Frequency Domains

$f(x,y) \rightarrow \boxed{h(x,y)} \rightarrow g(x,y)$ $\qquad$ $F(u,v) \rightarrow \boxed{H(u,v)} \rightarrow G(u,v)$

g(x,y) = f(x,y) * h(x,y) $\qquad \longleftrightarrow \qquad$ G(u,v) = F(u,v)H(u,v)

$(g(x,y) = F^{-1}(F(u,v)H(u,v)))$

h(x,y): spatial filter

In certain cases, h(x,y) can be chosen by specifying H(u,v)

H(u,v): Transfer function

48

The benefits of performing filtering in the frequency domain:

- It is convenient to perform a filter design. That is: Designing an appropriate filter to achieve the given goal is convenient since filtering is more intuitive in the frequency domain.
- Implementation may be more efficient with FFT.
- Convolution in the spatial domain reduces to multiplication in the frequency domain, and vice versa.

49

Filter Design

- The goal is to construct a spatial filter h(x, y) given the frequency filter H(u, v).

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

- Estimate h(x, y) by minimizing

$$\varepsilon^2=\sum_{u=0}^{N-1}\sum_{v=0}^{M-1} |H(u,v)-\hat{H}(u,v)|$$

- In practice, spatial convolution generally is simplified by using small masks that attempt to capture the salient features of their frequency domain counterparts.

50

Some Basic Filters and Their Properties

- **Notch Filter**: It is a constant function with a hole (notch) at the origin. → The filtered image would have the zero as the average pixel intensity. Consequently, notch filters are exceptionally useful tools when it is possible to identify spatial image effects caused by specific, localized frequency domain components.
- **Lowpass Filter**: It attenuates high frequencies while "passing" low frequencies. → The filtered image would have less sharp detail than the original image.
- **Highpass Filter**: It attenuates low frequencies while "passing" high frequencies. → The filtered image would have less gray level variations in smooth areas and emphasized transitional gray level detail.

51

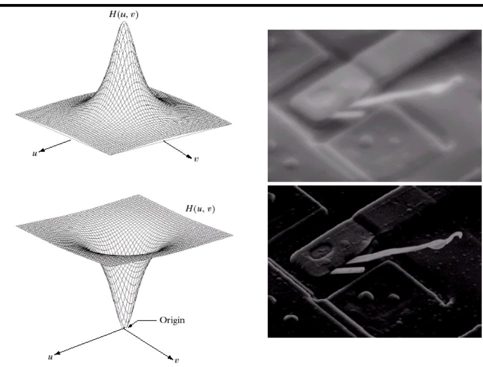
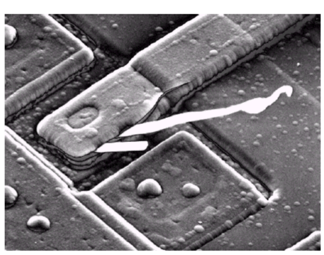


FIGURE 4.7 (a) A two-dimensional lowpass filter function. (b) Result of lowpass filtering the image in Fig. 4.4(a). (c) A two-dimensional highpass filter function. (d) Result of highpass filtering the image in Fig. 4.4(a).

52

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).



For a high-pass filtering, a procedure often followed is to add a constant to the filter so that it will not completely eliminate F(0, 0)

53

Ideal Lowpass Filter (ILPF)

- In the frequency domain, the ideal low-pass filter is defined as:

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

where $D(u,v)=\sqrt{[u-(M/2+1)]^2+[v-(N/2+1)]^2}$

(A distance from point (u,v) to the origin of the frequency rectangle) and D_0 (a specified nonnegative quantity) is called ***cutoff frequency***.

54

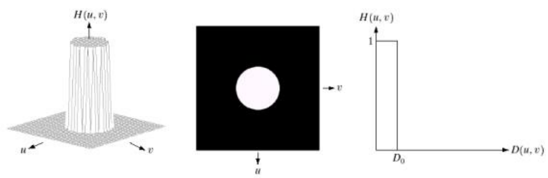


FIGURE 4.10 (a) Perspective plot of an ideal lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross section.

One way to establish a set of standard cutoff frequency loci is to compute circles that enclose specified amounts of total **image power PT.**

$$P_T = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} P(u,v) = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} |F(u,v)|^2 = \sum_{u=0}^{M-1}\sum_{v=0}^{N-1} (R^2(u,v) + I^2(u,v))$$

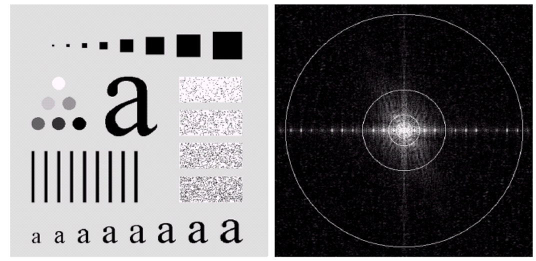
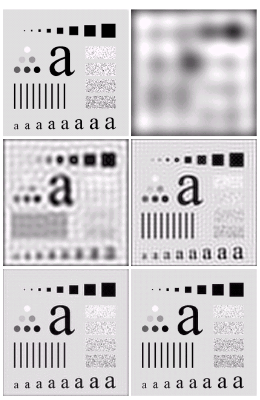
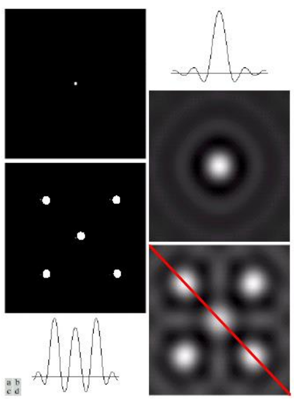


FIGURE 4.11 (a) An image of size 500 × 500 pixels and (b) its Fourier spectrum. The superimposed circles have radii values of 5, 15, 30, 80, and 230, which enclose 92.0, 94.6, 96.4, 98.0, and 99.5% of the image power, respectively.



Results of ideal lowpass filtering with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in previous slide. The power removed by these filters was 8%, 5.4%, 3.6%, 2%, and 0.5% of the total image power, respectively.



Problems with Ideal LPF

1. Difficult to implement the sharp cutoff frequency with electronic component.
2. Severe blurring and ringing

→ In practice, we use filters which attenuate the high frequencies smoothly (no ringing effect).

Butterworth Lowpass Filter (BLPF)

- The transfer function of a Butterworth lowpass filter of order n, and with cutoff frequency at a distance D_0 from the origin, is defined as:

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

where n is an integer.

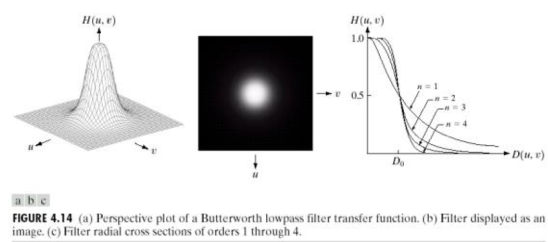
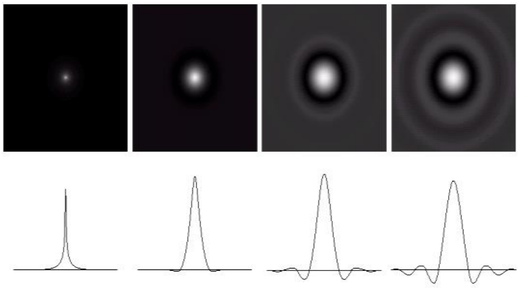


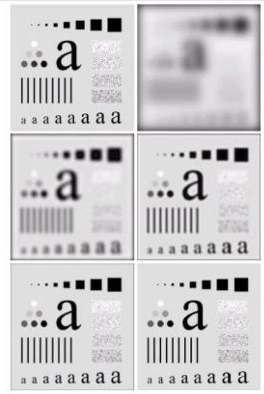
FIGURE 4.14 (a) Perspective plot of a Butterworth lowpass filter transfer function. (b) Filter displayed as an image. (c) Filter radial cross sections of orders 1 through 4.

- It does not have a sharp transition from passed and filtered frequencies. Instead, it has a smooth transition.
- As n increases, the transition goes towards sharper.
- In general, the order 2 is a good compromise between effective lowpass filtering and acceptable ringing characteristics.



a b c d

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



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.

Results of Butterworth lowpass of order 2, filtering with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b).

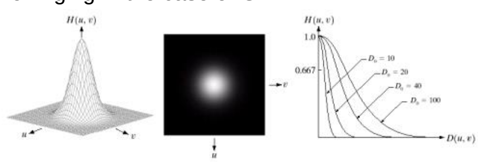
There is no ringing effect due to the filter's smooth transition between low and high frequencies.

62

Gaussian Lowpass Filter (GLPF)

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

- The variance is a measure of the spread of the Gaussian curve. The larger the variance, the larger the cutoff frequency and the milder the filtering.
- No ringing in the case of GLPF.



a b c

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.

63

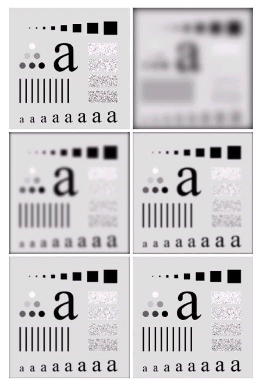


FIGURE 4.18 (a) Original image. (b)–(f) Results of filtering with Gaussian lowpass filters with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b). Compare with Figs. 4.12 and 4.15. a b c d e f

Results of Gaussian lowpass filtering with cutoff frequencies set at radii values of 5, 15, 30, 80, and 230, as shown in Fig. 4.11(b).

There is no ringing effect due to the filter's smooth transition between low and high frequencies.

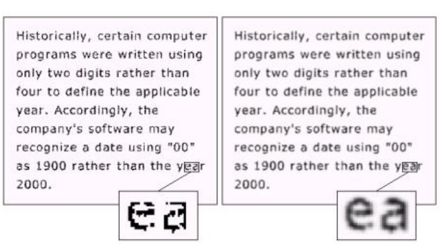
It does achieve as much smoothing as the BLPF of order 2.

64

Ex1: Character Recognition

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).



(a) Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.

(b) Historically, certain computer programs were written using only two digits rather than four to define the applicable year. Accordingly, the company's software may recognize a date using "00" as 1900 rather than the year 2000.

D0 = 80

65

Ex2: Printing and Publishing

a b c

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).

Ex3: Process Satellite Images

a b c

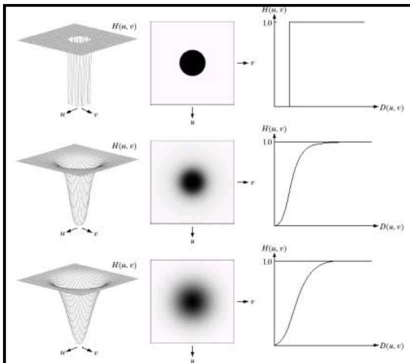
FIGURE 4.21 (a) Image showing prominent scan lines. (b) Result of using a GLPF with $D_0 = 30$. (c) Result of using a GLPF with $D_0 = 10$. (Original image courtesy of NOAA.)

67

Sharpening Frequency Domain Filters

- Because edges and other abrupt changes in gray levels are associated with high-frequency components, image sharpening can be achieved in the frequency domain by a highpass filtering process, which attenuates the low-frequency components without disturbing high-frequency information in the Fourier transform.

68

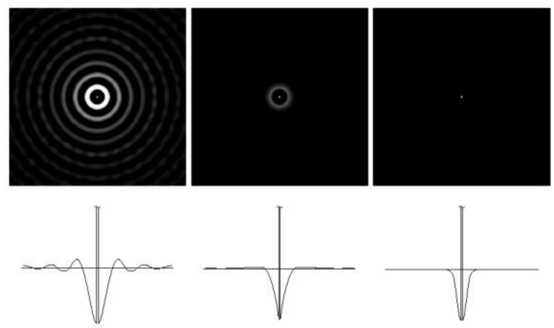


The Butterworth filter represents a transition between the sharpness of the ideal filter and the total smoothness of the Gaussian Filter

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.

69



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.

70

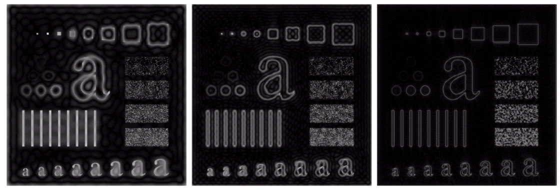
Ideal High-Pass Filter (IHPF)

In frequency domain, an IHPF is defined as:

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

It is the opposite of the ILPF in the sense that it sets to zero all frequencies inside a circle of radius D_0 while passing, without attenuation, all frequencies outside the circle.

71



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).

72

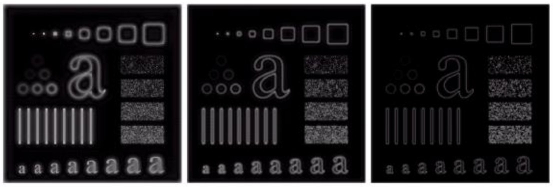
Butterworth High-Pass Filter (BHPF)

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

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

where n is an integer. It does not have a sharp transition from passed and filtered frequencies. Instead, it has a smooth transition.

73



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.

74

Gaussian High-Pass Filter (GHPF)

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

- The variance is a measure of the spread of the Gaussian curve.
- The results obtained are smoother than the other two filters. Even the filtering of the smaller objects and thin bars is cleaner with the Gaussian filter. See Fig. 4.26

75

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.

76

The Laplacian in the Frequency Domain

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

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

$$\nabla^2 f(x,y) \Leftrightarrow H(u,v)F(u,v) = -[(u - M/2)^2 + (v - N/2)^2] F(u,v)$$

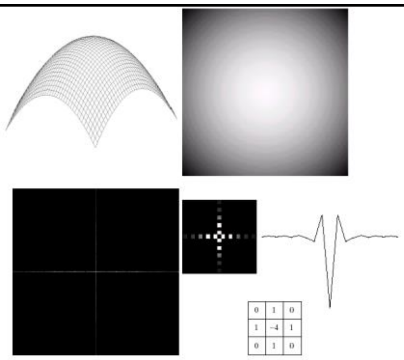
- The spatial domain Laplacian filter function is obtained by taking the inverse FT of

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

- The enhanced image g(x,y) can be obtained by:

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

77



a b
c d e
f

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.

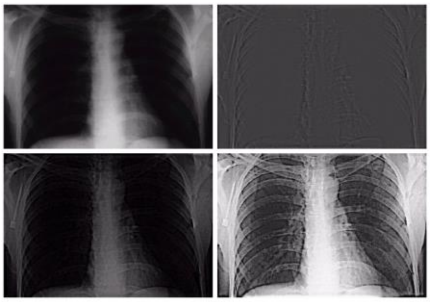
78

High Frequency Emphasis

- To alleviate the problem that low frequency components are severely attenuated, a constant is usually added to the HPF to preserve the low frequency components. That is: Add a constant to H(u,v) to preserve low-frequencies.
- It emphasizes edges but fine detail in the image (low frequencies) are lost.
- Since high-frequency components are further emphasized, histogram equalization is needed.

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

79



a b
c d

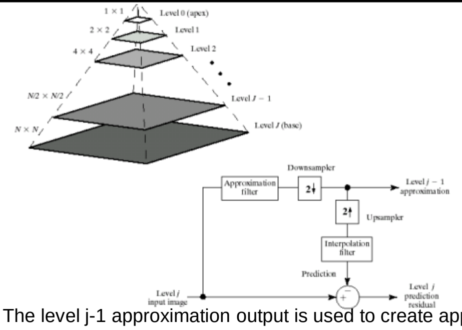
FIGURE 4.30 (a) A chest X-ray image. (b) Result of Butterworth highpass filtering. (c) Result of high-frequency emphasis filtering. (d) Result of performing histogram equalization on (c). (Original image courtesy Dr. Thomas R. Gest, Division of Anatomical Sciences, University of Michigan Medical School.)

80

Wavelet Coding

- Wavelet coding is a type of image pyramid, which is a powerful, but conceptually simple structure for representing images at more than one resolution. That is, image pyramid is a collection of decreasing resolution images arranged in the shape of a pyramid.
- The basic idea is to represent a given function as a combination of "basis" functions belonging to a specified set, whose analytic properties are readily accessible. With luck, the given function is well approximated by just a few basis functions. In that case, it is easy to work with the function. In particular, its description can be reduced from a painstaking point-by-point report to a handful of nonzero coefficients.

81



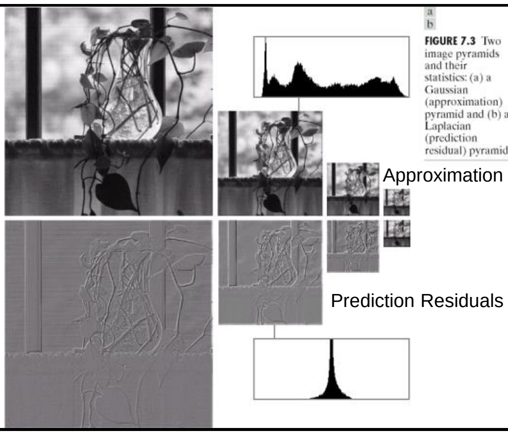
a
b

FIGURE 7.2 (a) A pyramidal image structure and (b) system block diagram for creating it.

The level j-1 approximation output is used to create approximation pyramids, which contain one or more approximations of the original image.

The level j prediction residual output is used to build prediction residual pyramids.

82



a
b

FIGURE 7.3 Two image pyramids and their statistics: (a) a Gaussian (approximation) pyramid and (b) a Laplacian (prediction residual) pyramid.

Wavelet Coding: Multi-Resolution Nature

- In general, a pyramid's lower-resolution levels can be used for the analysis of large structures or overall image context; its high-resolution images are appropriate for analyzing individual object characteristics.
- Such a coarse to fine analysis strategy is particularly useful in pattern recognition.

84

Wavelet Coding: Mechanism

If one has a filtering equation of the form [G]=[H][F], then one has an output for every input value x of the function F

Here: H is the approximation filter.

$$\begin{bmatrix} G(0) \\ G(1) \\ . \\ . \\ G(N-1) \end{bmatrix}_{N\times 1} = \begin{bmatrix} h(0) & h(1) & \dots\ \dots & & & \\ \dots & h(0) & h(1) & \dots\ \dots & & \\ \dots. & & h(0) & h(1) & \dots & \\ \dots.. & & & & & \\ h(1) \dots. & \dots\dots\dots & & \dots\dots\dots & \dots.. & h(0) \end{bmatrix}_{N\times N} \begin{bmatrix} F(0) \\ F(1) \\ . \\ . \\ F(N-1) \end{bmatrix}_{N\times 1}$$

Represent the convolution procedure by a matrix multiplication

The act of decimation (downsampling) is to produce an output for only G(0), G(2), G(4), etc. One can accomplish this with the decimation matrix

$$[\downarrow 2] = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & \dots\ \dots & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & \dots\ \dots & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & \dots\ \dots & 0 \\ \dots & & & & & & & \dots \\ \dots & & & & & & & \dots \end{bmatrix}_{\frac{N}{2}, N}$$

The filtering equation becomes $[G] = [\downarrow 2][H][F]$

The matrix $[\downarrow 2]$ serves to delete every other output function value shown below for N=8. The output function G' is the output from the filter without decimation.

$$\begin{bmatrix} G(0) \\ G(2) \\ G(4) \\ G(6) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} h(0) & h(1) & \dots\ \dots & & & \\ \dots & h(0) & h(1) & \dots\ \dots & & \\ \dots. & & h(0) & h(1) & \dots & \\ \dots.. & & & & & \\ h(1) \dots. & \dots\dots\dots & & \dots\dots\dots & \dots.. & h(0) \end{bmatrix}_{8\times 8} \begin{bmatrix} F(0) \\ F(1) \\ . \\ . \\ F(7) \end{bmatrix}_{8\times 1}$$

$$= \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} G'(0) \\ G'(1) \\ \dots \\ \dots \\ G'(7) \end{bmatrix}$$

The transpose of the decimation matrix is the upsampling matrix. Upsampling restores the size of the function

$$[\uparrow 2] \begin{bmatrix} F(0) \\ F(2) \\ F(4) \\ F(6) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} F(0) \\ F(2) \\ F(4) \\ F(6) \end{bmatrix} = \begin{bmatrix} F(0) \\ 0 \\ F(2) \\ 0 \\ F(4) \\ 0 \\ F(6) \\ 0 \end{bmatrix} = \begin{bmatrix} F'(0) \\ F'(1) \\ F'(2) \\ F'(3) \\ F'(4) \\ F'(5) \\ F'(6) \\ F'(7) \end{bmatrix}$$

This shows that upsampling takes a function with even values of f(x) and reproduces it with zeros for f'(x) for odd x

- The difference between the upsampling restored function and the original function forms the prediction residuals (i.e., errors).
- We can also design a high-pass filter to generate these prediction residuals.

This is the Haar Wavelet Transform

$$L = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \qquad H = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

What are the downsampling results of the low-pass and high-pass filter?

We can combine these operations in one matrix where the low-pass and high-pass filters alternate as:

$$\begin{bmatrix} G_L(0) \\ G_H(0) \\ G_L(1) \\ G_H(1) \\ G_L(2) \\ G_H(2) \\ G_L(3) \\ G_H(3) \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \\ F(4) \\ F(5) \\ F(6) \\ F(7) \end{bmatrix}$$

Haar Wavelet: Combining Low-Pass and High-Pass Filters

We can combine the filters in the following way to consolidate the low-pass and high-pass parts.

$$\begin{bmatrix} G_L(0) \\ G_L(2) \\ G_L(4) \\ G_L(6) \\ G_H(0) \\ G_H(2) \\ G_H(4) \\ G_H(6) \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \begin{bmatrix} F(0) \\ F(1) \\ F(2) \\ F(3) \\ F(4) \\ F(5) \\ F(6) \\ F(7) \end{bmatrix}$$

Low-pass Filter

High-pass Filter

91

Intuitive Description of Wavelet

- Suppose that we apply wavelets to 1-D signals or time series. A time series is simply a sample of a signal or a record of something, like temperature, water level or market data (like equity close price).
- Wavelets allow a time series to be viewed in multiple resolutions. Each resolution reflects a different frequency. The wavelet technique takes averages and differences of a signal, breaking the signal down into spectrum. Most the wavelet algorithms work on time series a power of two values (e.g., 64, 128, 256...).

92

Wavelet Coefficients (Differences)

- Each step of the wavelet transform produces two sets of values: a set of averages and a set of differences (the differences are referred to as wavelet coefficients), that is half the size of the input data:
  - For example, if the time series contains 256 elements, the first step will produce 128 averages and 128 coefficients. The averages then become the input for the next step (e.g., 128 averages resulting in a new set of 64 averages and 64 coefficients). This continues until one average and one coefficient is calculated.

93

Spatial and Frequency Localization Property

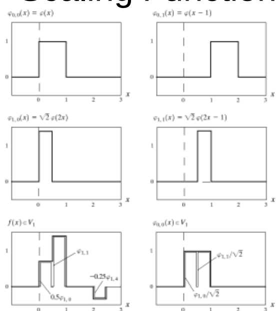
- The average and difference of the time series is made across a window of values. Most wavelet algorithms calculate each new average and difference by shifting this window over the input data.
  - For example, if the input time series contains 256 values, the window will be shifted by two elements if Haar Wavelet is utilized, 128 times, in calculating the averages and differences. The next step of the calculation uses the previous set of averages, also shifting the window by two elements. This has the effect of averaging across a four element window. Logically, the window increases by a factor of two each time.
- In the wavelet literature, this tree structured recursive algorithm is referred to as a pyramidal algorithm.

94

- The power of two coefficient (difference) spectrum generated by a wavelet calculation reflect change in the time series at various resolutions.
  - The first coefficient band generated reflects the highest frequency changes. Each later band reflects changes at lower and lower frequencies.
- There are an infinite number of wavelet basis functions. The more complex functions (like the Daubechies wavelets) produce overlapping averages and differences that provide a better average than the Haar wavelet at lower resolutions. However, these algorithms are more complicated.

95

Wavelet Mathematical View: Scaling Functions (Average)



a b
c d
e f

FIGURE 7.9 Haar scaling functions in V_0 in V_1.

$$\varphi_{j,k}(x) = 2^{j/2}\varphi(2^j x - k)$$

96

Wavelet Mathematical View: Wavelet Functions (Difference)

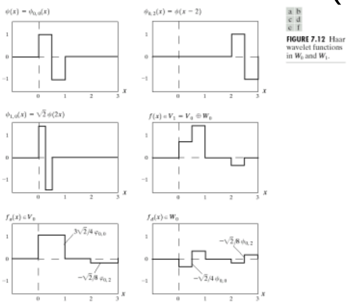


FIGURE 7.12 Haar wavelet functions in W_0 and W_1.

$$\psi_{j,k}(x) = 2^{j/2}\psi(2^j x - k)$$

97

Scaling and Wavelet Functions

- Translation k determines the position of the 1-D functions along the x-axis.
- Scale j determines the width – how broad or narrow the 1-D functions are along the x-axis.
- $2^{j/2}$ controls the height or amplitude.
- The associated expansion functions are binary scaling and integer translates of mother wavelet $\Psi(x) = \Psi_{0,0}(x)$ and scaling function $\varphi(x) = \varphi_{0,0}(x)$

98

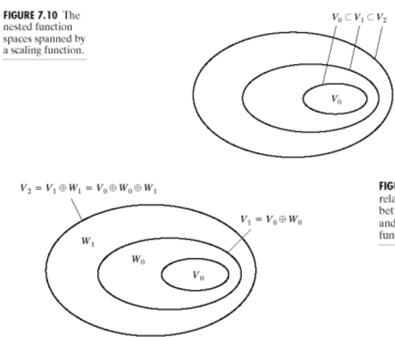


FIGURE 7.10 The nested function spaces spanned by a scaling function.

FIGURE 7.11 The relationship between scaling and wavelet function spaces.

The wavelet function Ψ together with its integer translates and binary scalings, spans the difference between any two adjacent scaling subspaces.

99

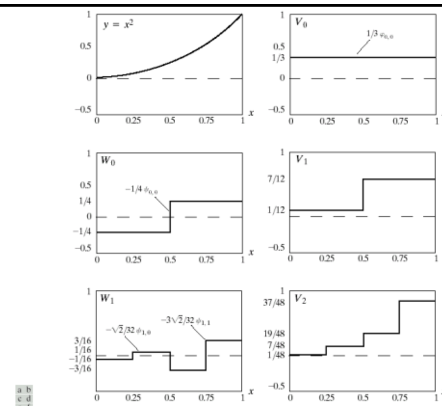


FIGURE 7.13 A wavelet series expansion of $y = x^2$ using Haar wavelets.

100

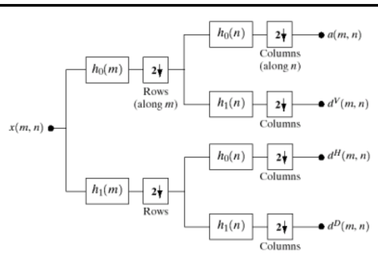


FIGURE 7.5 A two-dimensional, four-band filter bank for subband image coding.

$$\varphi(x, y) = \varphi(x)\varphi(y)$$

$$\psi^H(x, y) = \psi(x)\varphi(y)$$

$$\psi^V(x, y) = \varphi(x)\psi(y)$$

$$\psi^D(x, y) = \psi(x)\psi(y)$$

101

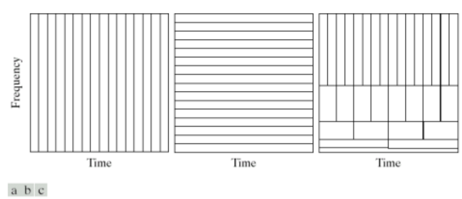
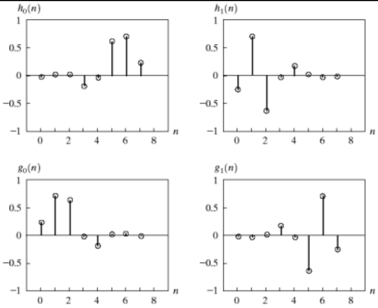


FIGURE 7.21 Time-frequency tilings for (a) sampled data, (b) FFT, and (c) FWT basis functions.

The sampled data is a delta function. It emphasizes the instants when events occur but provides no frequency information.
The FFT domain emphasizes the frequency that are present in events that occur over long periods but provides no time resolution.
In DWT domain, each title represents an equal portion of the time-frequency plane. At low frequencies, the tiles are shorter but are wider. At high frequencies, tile width is smaller and the tile height is greater.
➔This fundamental difference between the FFT and DWT makes the DWT a good candidate for the analysis of non-stationary functions whose frequencies varies in time.

102

FIGURE 7.6 The impulse responses of four 8-tap Daubechies orthonormal filters.



[f1, f2] = wfilters('db4', 'd')

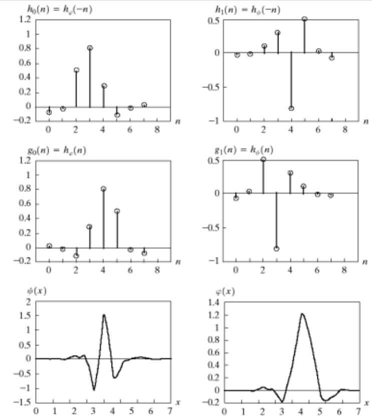
f1 = -0.0106 0.0329 0.0308 -0.1870 -0.0280 0.6309 0.7148 0.2304
f2 = -0.2304 0.7148 -0.6309 -0.0280 0.1870 0.0308 -0.0329 -0.0106

[f1, f2] = wfilters('db4', 'r')

f1 = 0.2304 0.7148 0.6309 -0.0280 -0.1870 0.0308 0.0329 -0.0106
f2 = -0.0106 -0.0329 0.0308 0.1870 -0.0280 -0.6309 0.7148 -0.2304

103

FIGURE 7.24 Fourth-order symlets: (a)–(b) decomposition filters; (c)–(d) reconstruction filters; (e) the one-dimensional wavelet; (f) the one-dimensional scaling function; and (g) one of three two-dimensional wavelets, $\psi^{H}(x, y)$.



104

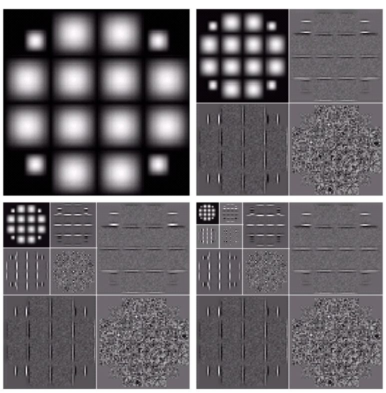


FIGURE 7.23 A three-scale FWT.

105

Matlab Wavelet Related Functions

help waveinfo

Available family short names are:
'haar': Haar wavelet.
'db': Daubechies wavelets.
'sym': Symlets.
'coif': Coiflets.
'bior': Biorthogonal wavelets.
'rbio': Reverse biorthogonal wavelets.
'meyr': Meyer wavelet.
'dmey': Discrete Meyer wavelet.
'gaus': Gaussian wavelets.
'mexh': Mexican hat wavelet.
'morl': Morlet wavelet.
'cgau': Complex Gaussian wavelets.
'cmor': Complex Morlet wavelets.
'shan': Complex Shannon wavelets.
'fbsp': Complex Frequency B-spline wavelets.
'fk': Fejer-Korovkin orthogonal wavelets

106

Matlab Wavelet Related Functions

waveinfo('haar')

HAARINFO Information on Haar wavelet.
General characteristics: Compactly supported wavelet, the oldest and the simplest wavelet.

scaling function phi = 1 on [0 1] and 0 otherwise.
wavelet function psi = 1 on [0 0.5), = -1 on [0.5 1] and 0 otherwise.

Family	Haar
Short name	haar
Examples	haar is the same as db1
Orthogonal	yes
Biorthogonal	yes
Compact support	yes
DWT	possible
CWT	possible

Support width	1
Filters length	2
Regularity	haar is not continuous
Symmetry	yes
Number of vanishing moments for psi	1

Reference: I. Daubechies, Ten lectures on wavelets, CBMS, SIAM, 61, 1994, 194-202.

107

Matlab Wavelet Related Functions

- dwtmode('status') or dwtmode: return the current active extension mode.
- help dwtmode will show all the padding modes.
- dwtmode('per') is suggested to ensure the proper dimension is returned in each decomposition.

108

```
>>dwtmode('per') ;
>>f = magic(8) ;
>>[c1, s1] = wavedec2(f, 3, 'db2') ;
>>size(c1)
ans =
    1   64
>>s1 =
    1    1
    1    1
    2    2
    4    4
    8    8
>>approx = appcoef2(c1, s1, 'db2')
approx =  260.0000
>>horizdet2 = detcoef2('h', c1, s1, 2)
horizdet2 =
   -1.7321    1.7321
   -1.7321    1.7321
>>newc1 = wthcoef2('h', c1, s1, 2);
>>newhorzdet2 = detcoef2('h', newc1, s1, 2)
newhorzdet2 =
    0    0
    0    0
```

109

Questions?

- What does the reconstruction image look like after the following operations:

1) Set all the approximation coefficients as 0's
2) Set the first level detail coefficients as 0's
3) Set the second level detail coefficients as 0's

- How to get a refined approximation incorporating only the 4-th level details?

110

- Practice on "wavemenu" command in Matlab
- Type "help wavelet" in Matlab → List all possible wavelet related functions provided by Matlab
- "wavedec2" and "dwt2" are two functions used for wavelet decomposition → What is the difference between these two functions?
- "waverec2" and "idwt2" are two functions used for wavelet reconstruction. → What is the difference between these two functions?

111

U =

1	3	3	1	0	0	0	0	Low-pass Filter
0	0	1	3	3	1	0	0	
0	0	0	0	1	3	3	1	
3	1	0	0	0	0	1	3	
1	3	-3	-1	0	0	0	0	High-pass Filter
0	0	1	3	-3	-1	0	0	
0	0	0	0	1	3	-3	-1	
-3	-1	0	0	0	0	1	3	

112

U^{-1} = 1/16

-1	0	0	3	-1	0	0	-3
3	0	0	-1	3	0	0	1
3	-1	0	0	-3	-1	0	0
-1	3	0	0	1	3	0	0
0	3	-1	0	0	-3	-1	0
0	-1	3	0	0	1	3	0
0	0	3	-1	0	0	-3	-1
0	0	-1	3	0	0	1	3

113

Input Signal F^T =

0	0	1	2	3	4	4	4

G = UF =

5
20
31
16
-5
-6
-1
16

Note that G is scaled up from the values of F.

The top half of G reflects the low-pass operation and the bottom half of G represents the high-pass operation.

To reconstruct F from G one can compute $F = U^{-1}G$

114

Input Image

F(x,y) =

0	0	0	2	3	4	4	4
0	0	0	2	3	4	4	4
0	0	0	2	3	4	4	4
0	0	0	2	3	4	4	4
0	0	0	2	3	4	4	4
0	0	0	2	3	4	4	4
0	0	0	2	3	4	4	4
0	0	0	2	3	4	4	4

115

F(x, y) U^T =

2	19	31	16	-2	-7	-1	16
2	19	31	16	-2	-7	-1	16
2	19	31	16	-2	-7	-1	16
2	19	31	16	-2	-7	-1	16
2	19	31	16	-2	-7	-1	16
2	19	31	16	-2	-7	-1	16
2	19	31	16	-2	-7	-1	16
2	19	31	16	-2	-7	-1	16

Low-pass Part **High-pass Part**

x-dir **x-dir**

116

X

Y

Low-pass x dir High-pass x dir

Low-pass y dir Low-pass y dir

G(x, y) =

UF(x, y) U^T =

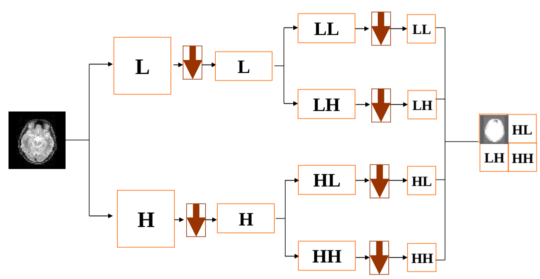
16	152	248	128	-16	-56	-8	128
16	152	248	128	-16	-56	-8	128
16	152	248	128	-16	-56	-8	128
16	152	248	128	-16	-56	-8	128
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Low-pass x dir High-pass x dir

High-pass y dir High-pass y dir

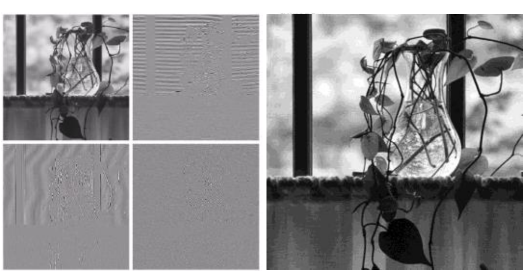
117

Wavelet Coding



118

Wavelet Coding



119

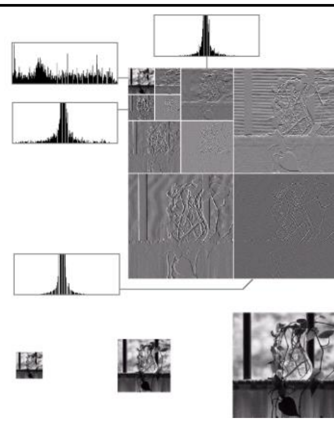


FIGURE 7.8 (a) A discrete wavelet transform using Haar basis functions. Its local histogram variations are also shown; (b)–(d) Several different approximations (64 × 64, 128 × 128, and 256 × 256) that can be obtained from (a).

120

Wavelet Coding Applications -- Compression

Compression Ratio: 34:1

Compression Ratio: 67:1

Difference between the original image

Difference between the original image

The precise nature shown in the zoomed images

The precise nature shown in the zoomed images

121

Wavelet Coding Applications -- Compression (Cont.)

Compression Ratio: 108:1

Compression Ratio: 167:1

Difference between the original image

Difference between the original image

The precise nature shown in the zoomed images

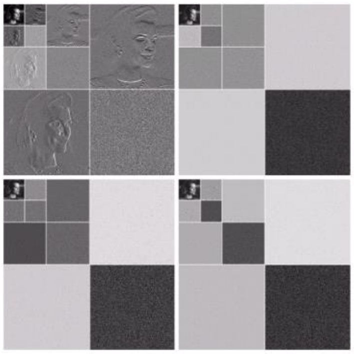
The precise nature shown in the zoomed images

122

Compression Standard for Fingerprint

- It is based on scalar quantization of a 64-subband discrete wavelet transform. Compression takes place in the quantization step, where the coefficients of the transform within each subband are, in effect, assigned to integer-valued "bins." (The information is further compressed by Huffman coding, which uses strings of variable length to represent data.)
- The wavelet/scalar quantization standard is not locked in to a particular wavelet basis. A digitized fingerprint file will contain not only the compressed image, but also tables specifying the wavelet transform, scalar quantizer, and Huffman code.
- At compression ratios of about 20:1, the new standard will facilitate the rapid transmission of information that is crucial for effective police work.

123



a b
c d

FIGURE 8.42 Wavelet transforms of Fig. 8.23 with respect to (a) Haar wavelets, (b) Daubechies wavelets, (c) symlets, and (d) Cohen-Daubechies-Feauveau biorthogonal wavelets.

124

Wavelet Coding Applications -- Noise Removal

- The technique starts with the application of a wavelet transform to the noisy signal or data set.
- It then "shrinks" each of the wavelet coefficients toward zero , using a soft-threshold nonlinearity, so that suitably small coefficients are set precisely to zero.
- Finally, the altered coefficients are inverted to produce a "denoised" signal.

  → Proposed by Dr. David Donoho at Stanford University

125

Wavelet Coding Applications -- PDEs

- John Weiss and Sam Qian have been investigating the use of wavelets in the numerical solution of partial differential equations (PDEs).
  - One of the things that wavelets seem to be very useful for are situations where there have strong gradients. That includes problems involving shock waves and turbulence.
- What they are trying to do is see if they can make the wavelet method as universal as the finite element method but obtain a better rate of convergence.

126

Wavelet Coding Applications -- Others

- Image/Video Retrieval
  - Color
  - Texture
  - Shape
- Pattern Recognition
  - Medical Disease Diagnosis
  - Pedestrian Detection
  - Vehicle Detection
- Digital Watermarking
- Data Mining
- Data Analysis

127