Image Processing and Analysis
Lecture 9、Wavelets and Multiresolution Processing

Weiqiang Wang
School of Computer Science and Technology, UCAS

November 6, 2023

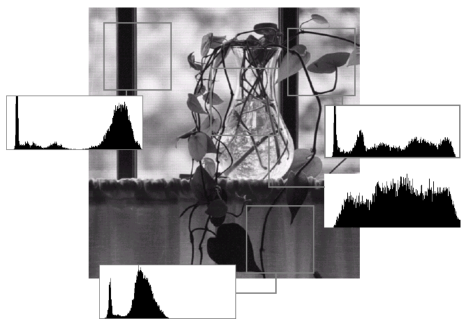
Outline

Background

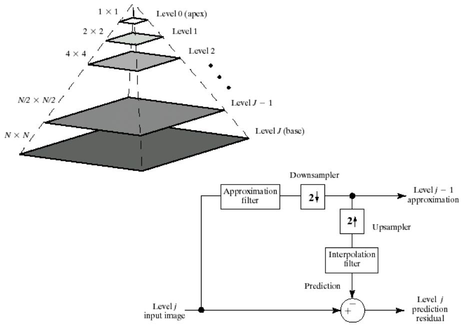
- Although the Fourier transform has been the mainstream in image analysis since 1950s, the wavelet transform becomes very popular since 1990s.
- Unlike the Fourier transform, whose basis function is sinusoids, wavelet transforms are based on small waves, called wavelet, of varying frequency and limited duration.
- Provided a musical score for an image, wavelet tranforms can reveal not only what notes (or frequency) to play but also when to play them; while, the Fourier transform can only provide the notes or frequency information.
- wavelet tranforms are often associated with the multiresolution theory. Multiresoluton theory incorporates and unifies techniques from a variety of disciplines, including subband coding from signal processing, quadrature mirror filtering from digital speech recognition, and pyramidal image processing.

Background

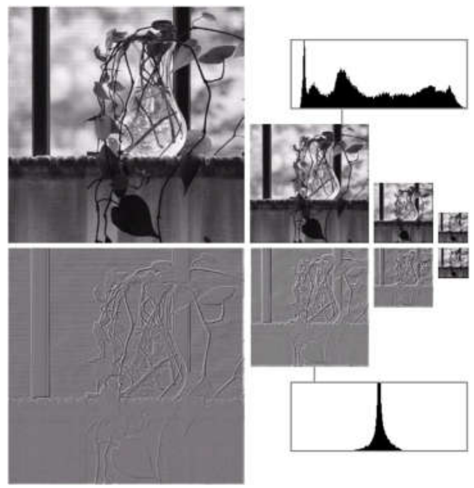
FIGURE 7.1 A natural image and its local histogram variations.



Background(cont.)



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

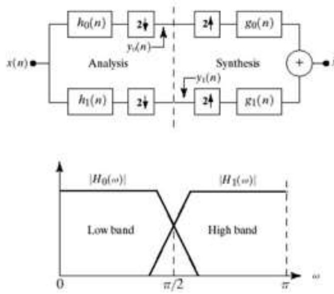


a b
FIGURE 7.3 Two image pyramids and their statistics: (a) a Gaussian (approximation) pyramid and (b) a Laplacian (prediction residual) pyramid.

Subband Coding(cont.)

- Subband coding

a b
FIGURE 7.4 (a) A two-band filter bank for one-dimensional subband coding and decoding, and (b) its spectrum splitting properties.



Subband Coding(cont.)

- *Z-transform*

  $X(z) = \sum_{-\infty}^{\infty} x(n) z^{-n}$ where z is a complex variable.
- Downsampling

  $$x_{down}(n) = x(2n) \Leftrightarrow X_{down}(z) = \frac{1}{2}[X(z^{\frac{1}{2}}) + X(-z^{\frac{1}{2}})]$$
- Upsampling

  $$x^{up}(n) = \begin{cases} x(n/2), n = 0, 2, 4, \ldots \\ 0 \quad\quad , otherwise \end{cases} \Leftrightarrow X^{up}(z) = X(z^2) \tag{1}$$
- Combine downsample and upsample,we get:

  $$\hat{X}(z) = \frac{1}{2}[X(z) + X(-z)]$$

  where

  $$\begin{cases} \hat{x}(n) = Z^{-1}[\hat{X}(z)] \\ Z^{-1}[\hat{X}(-z)] = (-1)^n \hat{x}(n) \end{cases} \tag{2}$$

Subband Coding(cont.)

- We can express the subband coding and decoding system as:

$$\hat{X}(z) = \frac{1}{2}G_0(z)[H_0(z)X(z) + H_0(-z)X(-z)] + \frac{1}{2}G_1(z)[H_1(z)X(z) + H_1(-z)X(-z)]$$

- where the output of filter $h_0(n)$ is defined by the transform pair

$$h_0(n) * x(n) = \sum_k h_0(n-k)x(k) \Leftrightarrow H_0(z)X(z)$$

- Rearrange the terms in the above Eq, we get

$$\hat{X}(z) = \frac{1}{2}[H_0(z)G_0(z) + H_1(z)G_1(z)]X(z) + \frac{1}{2}[H_0(-z)G_0(z) + H_1(-z)G_1(z)]X(-z)$$

Subband Coding(cont.)

- For error-free reconstruciton of the input, we impose the following conditions:

$$H_0(-z)G_0(z) + H_1(-z)G_1(z) = 0$$
$$H_0(z)G_0(z) + H_1(z)G_1(z) = 2$$

- They can incorporated into the single matrix expression

$$[G_0(z)\ \ G_1(z)]\mathbf{H_m}(z) = [2\ \ 0]$$

where the analysis modulation matirx $\mathbf{H}_m(z)$ is

$$\mathbf{H_m}(z) = \begin{bmatrix} H_0(z) & H_0(-z) \\ H_1(z) & H_1(-z) \end{bmatrix}$$

- Assuming the $\mathbf{H}_m(z)$ is nonsingular, we can get

$$\begin{bmatrix} G_0(z) \\ G_1(z) \end{bmatrix} = \frac{2}{det(\boldsymbol{H_m}(z))}\begin{bmatrix} H_1(-z) \\ -H_0(-z) \end{bmatrix}$$

Characteristics of Perfect Construction Filter Banks(PCFB)

$$\begin{bmatrix} G_0(z) \\ G_1(z) \end{bmatrix} = \frac{2}{det(\boldsymbol{H_m}(z))}\begin{bmatrix} H_1(-z) \\ -H_0(-z) \end{bmatrix}$$

- The above equation tells us $G_0(z)$ is a function of $H_1(-z)$, and $G_1(z)$ is a function of $H_0(-z)$. So the analysis and synthesis filters are cross-modulated
- For finite pulse response (FIR) filters, the determinate of the analysis modulation matrix $\mathbf{H}_m(z)$ is a pure delay, (see Vetterli and Kovacevic [1995]), i.e.,

$$det(\boldsymbol{H_m}(z)) = \alpha z^{-(2k+1)}$$

- The $z^{-(2k+1)}$ term can be considered arbitrary since it is a shift that only changes the overall delay of the filter. Ignoring the delay, letting $\alpha = 2$ and taking the reverse z-tranform, we get

$$g_0(n) = (-1)^n h_1(n)$$
$$g_1(n) = (-1)^{n+1} h_0(n)$$

Characteristics of PCFB, Cross-Modulation

- if $\alpha = -2$, the resulting expressions are sign reversed, i.e.,

$$g_0(n) = (-1)^{n+1} h_1(n)$$

$$g_1(n) = (-1)^n h_0(n)$$

- Thus, **FIR synthesis filters are cross-modulated copies of the analysis filters with one and only one being sign reversed**.

Characteristics of PCFB, Biorthogonality

- Now we demonstrate another important property of analysis and synthesis filters, biorthogonality.
- Let $P(z)$ be the product of the lowpass analysis and synthesis filer transfer functions, and we get

$$P(z) = G_0(z)H_0(z) = \frac{2}{\det(\mathbf{H}_m(z))} H_0(z)\ H_1(-z)$$

- Since $det(\mathbf{H}_m(z)) = -det(\mathbf{H}_m(-z))$, product $G_1(z)H_1(z)$ can be similarly defined as

$$G_1(z)H_1(z) = \frac{-2}{\det(\mathbf{H}_m(z))} H_0(-z)\ H_1(z) = P(-z)$$

- Thus, $G_1(z)H_1(z)$ $=P(-z)$ $=G_0(-z)H_0(-z)$ .if we substitute it into the second constraint of perfect construction equations, we have

$$G_0(z)H_0(z) + G_0(-z)H_0(-z) = 2$$

Characteristics of PCFB, Biorthogonality (cont.)

- For both sides of $G_0(z)H_0(z) + G_0(-z)H_0(-z) = 2$, we take the reverse z-transform and get

$$\sum_k g_0(k)h_0(n-k) + (-1)^n \sum_k g_0(k)h_0(n-k) = 2\delta(n)$$

- Since the odd indexed terms cancel, additional simplification yields

$$\sum_k g_0(k)h_0(2n-k) = < g_0(k), h_0(2n-k) > = \delta(n)$$

- In the similar way, we can show that

$$< g_1(k), h_1(2n-k) > = \delta(n)$$
$$< g_0(k), h_1(2n-k) > = 0$$
$$< g_1(k), h_0(2n-k) > = 0$$

- We can establish the more general expression

$$< h_i(2n-k), g_j(k) > = \delta(i-j)\delta(n) \qquad i,j = \{0,1\}$$

Characteristics of PCFB, Biorthogonality (cont.)

- Filter banks satisfy this condition, i.e.,

$$< h_i(2n-k), g_i(k) > = \delta(i-j)\delta(n) \qquad i,j = \{0,1\}$$

  are called *biorthogonality*
- The analysis and synthesis filter impulse response of all two-band, real-coefficient,perfect construction filter banks are subject to the biothogonality constraint. are called *biorthogonality*
- Examples of biorthogonal, FIR filters include the biorthogonal spline family (Cohen,Daubechies and Feauveau[1992]) and the biorthogonal coiflet family (Tian and wells[1995])

The Solutions of Perfect Construction Filter Banks

- Three general solution to PCFB are given in the following table,

TABLE 7.1 Perfect reconstruction filter families.

Filter	QMF	CQF	Orthonormal
$H_0(z)$	$H_0^2(z) - H_0^2(-z) = 2$	$H_0(z)H_0(z^{-1}) + H_0^2(-z)H_0(-z^{-1}) = 2$	$G_0(z^{-1})$
$H_1(z)$	$H_0(-z)$	$z^{-1}H_0(-z^{-1})$	$G_1(z^{-1})$
$G_0(z)$	$H_0(z)$	$H_0(z^{-1})$	$G_0(z)G_0(z^{-1}) + G_0(-z)G_0(-z^{-1}) = 2$
$G_1(z)$	$-H_0(-z)$	$zH_0(-z)$	$-z^{-2K+1}G_0(-z^{-1})$

- While each satisfies the biothogonality requirement, they are generated in different ways and define unique classes of perfect reconstruction filters.
- For each class, a ”prototype” filter is designed to a particular specification and the remaining filtrs are computed from the prototype.

The Solutions of Perfect Construction Filter Banks (cont.)

- Table 7.1 are classic results from the filter bank literature- namely, *quadrature mirrror filter* (QMFs) and *conjugate quadrature filters* (CQFs), and *orthonormal filters* which are used to develop the fast wavelet transform.
- Besides biorthogonality, the orthonormality for perfect reconstruction filter banks are defined by

$$< g_i(n), g_j(n+2m) >= \delta(i-j)\delta(m) \qquad i,j = \{0,1\}$$

- As can be seen, G_1 is related to the lowpass synthesis filter G_0 by modulation, time reversal and odd shift. In addtiion, both H_1 and H_0 are time reversed versions of the corresponding synthesis filters, G_1 and G_0, repectiveley.

$$x(-n) \Leftrightarrow X(z^{-1}), \qquad x(n-k) \Leftrightarrow z^{-k}X(z)$$

The Solutions of Perfect Construction Filter Banks (cont.)

- Taking the inverse z-transform of the appropriate entries from colomn 3 of Table 7.1, we get that

$$g_1(n) = (-1)^n g_0(2K-1-n)$$

$$h_i(n) = g_i(2K-1-n), i = \{0,1\}$$

  where h_0, h_1, g_0, g_1 are the impulse responses of the defined orthonormal filters
- The examples of the orthonormal filters include the smith and Barnwell filter, Daubechies filters and the Vaidyanathan and hoang filter

Two-dimensional Subband Filters for Images

- The one-dimensional filters in Table 7.1 can be used as two-dimensional separable filters for the processing of images.

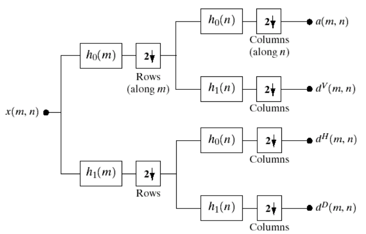
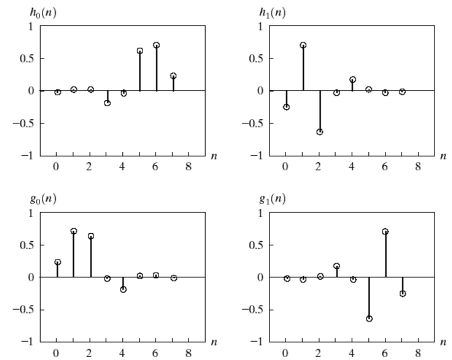


FIGURE 7.5 A two-dimensional, four-band filter bank for subband image coding.

- The resulting filtered outputs, denoted $a(m,n), d^V(m,n), d^H(m,n)$ and $d^D(m,n)$ in Fig 7.5, are called theapproximation, vertical detail , horizontal detail , and diagonal detail subbands of the images.
- One or more of these subbands can be split into four smaller subbands, which can be split again and so on.

Example of PCFBs (I)

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



Example of PCFBs (II)

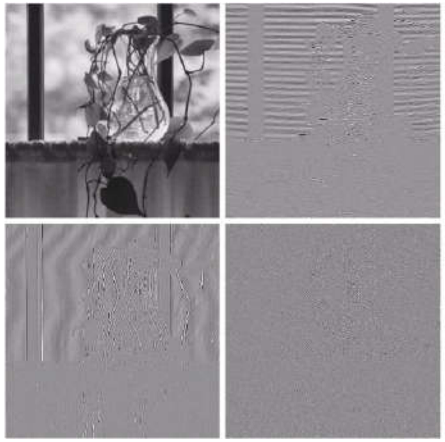


FIGURE 7.7 A four-band split of the vase in Fig. 7.1 using the subband coding system of Fig. 7.5.

Haar Transform

- The basis function of Haar Transform (Haar [1910]) is the oldest and simplest known orthonormal wavelet.
- The Haar Transform itself is both separable and symmetric and can be expressed as the matrix form,

  $\mathbf{T} = \mathbf{HFH}^T$,

  where $\mathbf{F}$ is an $N \times N$ matrix, $\mathbf{H}$ is an $N \times N$ tranform matrix, and $\mathbf{T}$ is a resulting $N \times N$ transform.
- Transform matrix $\mathbf{H}$ contains the Haar basis functions $h_k(z)$ defined over the interval $z \in [0,1]$, for $k = 0, ..., N-1$, where $N = 2^n$. To
- generate $\mathbf{H}$, we define the integer k, $k = 2^p + q - 1$, where $0 \leq p \leq n-1$, q=0 or 1 for p=0, and $1 \leq q \leq 2^p$ for $p \neq 0$ .Then the Haar basis functions are

$$h_0(z) = h_{00}(z) = \frac{1}{\sqrt{N}}, z\epsilon[0,1]$$

$$h_k(z) = h_{pq}(z) = \frac{1}{\sqrt{N}} \begin{cases} 2^{p/2}, & (q-1)/2^p \leq z \leq (q-0.5)/2^p \\ -2^{p/2}, & (q-0.5)/2^p \leq z \leq q/2^p \\ 0, & otherwise, z\epsilon[0,1] \end{cases}$$

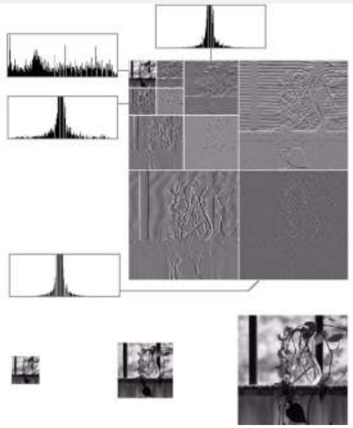
Haar Transform(cont.)

- When N=8, the corresponding k, q, p values

K	0	1	2	3	4	5	6	7
P	0	0	1	1	2	2	2	2
Q	0	1	1	2	1	2	3	4

- $$\mathbf{H}_8 = \frac{1}{\sqrt{8}} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & \sqrt{2} & \sqrt{2} & -\sqrt{2} & -\sqrt{2} \\ 2 & -2 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & -2 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 2 & -2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & -2 \end{bmatrix}$$

A Discrete Wavelet Transform Using the Haar Basis Functions



a
b c d

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).

Outline

Multiresolution Expansions

- In Multi-resolution analysis (MRA), a scaling function is used to create a series of approximations of a function or image. Additional function, called *wavelet*, are then used to encode the difference in information between adjacent approximations
- A signal or function $f(x)$ can often be better analyzed as a linear combination of expansion functions, i.e.,

$$f(x) = \sum_k \alpha_k \varphi_k(x)$$

- If the expansion is unique, i.e., there is only one set of α_k for any given $f(x)$, the $\phi_k(x)$ are called *basis functions* and the expansion set, $\{\phi_k(x)\}$, is called a basis for the class of functions that can be so expressed.
- The expressible functions form a functional space, called the *closed span* of the expansion set, denoted

$$V = \overline{Span_k\{\varphi_k(x)\}}$$

Series Expansions

- For any function space V and corresponding expansion set $\{\phi_k(x)\}$, there is a set of dual functions, denoted $\tilde{\varphi}_k(x)$,that can be used to compute the $\{\alpha_k\}$ coefficients by taking the integral inner products of dual $\tilde{\varphi}_k(x)$ and $f(x)$, i.e.,.

$$\alpha_k =< \tilde{\varphi}_k(x), f(x) >= \int \tilde{\varphi}_k^*(x) f(x) dx$$

- Three cases are involved in computing the coefficients α_k
- Case I: if the expansion functions form an orthonormal basis for V, i.e.,

$$< \varphi_j(x), \varphi_k(x) >= \delta_{jk} = \begin{cases} 0 & j \neq k \\ 1 & j = k \end{cases}$$

the basis and its dual are equivalent. Thus

$$\alpha_k =< \varphi_k(x), f(x) >$$

Series Expansions(cont.)

- Case II: if the expansion functions are not orthonormal, but are an orthogonal basis for V, that is,

$$< \varphi_j(x), \varphi_k(x) >= 0, j \neq k$$

and the bais functions and their duals are called biorthogonal, i.e.,

$$< \varphi_j(x), \tilde{\varphi}_k(x) >= \delta_{jk} = \begin{cases} 0 & j \neq k \\ 1 & j = k \end{cases}$$

Then

$$\alpha_k < \tilde{\varphi}_k(x), f(x) >$$

Series Expansions(cont.)

- Case III: if the expansion set is not a basis for V, but supports the expansion, it is a spanning set in which there is more than one set of α_k for any $f(x) \in V$. The expansion functions and their duals are said to be overcomplete or redundant. They form a frame in which

$$A||f(x)||^2 \leq \sum_k |< \varphi_k(x), f(x) >|^2 \leq B||f(x)||^2$$

for some $A > 0, B < \infty$ and $f(x) \in V$.

- Both $\alpha_k = < \tilde{\varphi}_k(x), f(x) >$ and $\alpha_k = < \varphi_k(x), f(x) >$ can be used to find the expansion coeefficients for frames
- if $A = B$, the expansion set is called a tight frame and it can be shown that

$$f(x) = \frac{1}{A} \sum_k < \varphi_k(x), f(x) > \varphi_k(x)$$

Multiresolution Expansions-Scaling Function

- Now consider the set of expansion functions composed of integer translations and binary scalings of the real, square-integrable function $\varphi_{j,k}(x)$,i.e.,

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

for all $j, k \in \mathbf{Z}$ and $\varphi_{j,k}(x) \in L^2(R)$.

- Here k determines the position of $\varphi_{j,k}(x)$ along the x-axis, j determines $\varphi_{j,k}(x)$'s width and $2^{j/2}$ controls its height or amplitude.
- $\varphi_{j,k}(x)$ is called *scaling function*. By choose $\varphi(x)$ wisely, $\{\varphi_{j,k}(x)\}$ can be made to span $L^2(R)$, the set of all measurable , square-integrable functions.

Multiresolution Expansions-Scaling Function (cont.)

- For a specific value, $j = j_0$, the expansion set $\{\varphi_{j_0,k}(x)\}$ is a subset of $\{\varphi_{j,k}(x)\}$. We can define that subspace as

$$V_{j_0} = \overline{Span\{\varphi_{j_0,k}(x)\}}$$

- Since V_{j_0} is the span of $\varphi_{j_0,k}(x)$ over k, if $f(x) \in V_{j_0}$, it can be written

$$f(x) = \sum_k \alpha_k \varphi_{j_0,k}(x)$$

- Generally, we will denote the subspace spanned over k for any j as

$$V_j = \overline{Span\{\varphi_{j,k}(x)\}}$$

- As we will see in the following example, increasing j increases the size of V_j, allowing functions with smaller variations or finer detail to be included in the subspace.

Example

Consider the unit-height,unit-width scaling function (Haar [1910])

$$\varphi(x), = \begin{cases} 1 & ,0 \leq x \leq 1 \\ 0 & ,otherwise \end{cases}$$

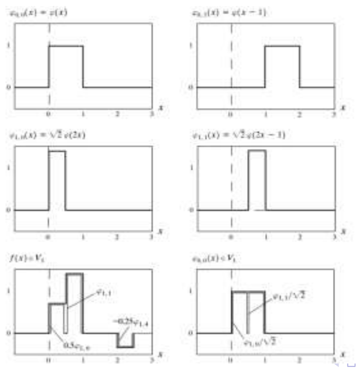


FIGURE 7.9 Haar scaling functions in V_0 in V_1.

MRA Requirements

The scaling function obeys the four fundamental requirements of multiresolution analysis (Mallat[1989a]):

1. The scaling function is orthogonal to its integer translates.
   - The Haar scaling function is said to have compact support, which means that is 0 everywhere outside a finite interval called the support.
   - the requirement for orthogonal integer translates becomes harder to satisfy as the support of the scaling function becomes larger than 1
2. The subspaces spanned by the scaling function at low scales are nested within those spanned at higher scales.
   $$V_{-\infty} \subset \ \dots \ \subset V_{-1} \subset V_0 \subset V_1 \subset V_3 \subset \dots \ \subset V_\infty$$
3. The only function that is common to all V_j is $f(x) = 0$.
   $$V_{-\infty} = 0$$
4. Any function can be represented with arbitrary precision.
   $$V_\infty \ = \ L^2(R)$$

Scaling Function Coefficients

- The expansion function of subspace V_j can be expressed as a weighted sum of the expansion functions of subspace V_{j+1}, i.e.,
  $$\varphi_{j,k}(x) = \sum_n \alpha_n \varphi_{j+1,n}(x)$$
- Changing variable α to $h_\varphi(n)$, we further have
  $$\varphi_{j,k}(x) = \sum_n h_\varphi(n) 2^{(j+1)/2} \varphi\,(2^{(j+1)} x - n)$$
- For $\varphi(x) \ = \ \varphi_{0,0}(x)$, we obtain the simpler expression
  $$\varphi\,(x) = \sum_n h_\varphi(n) \sqrt{2} \varphi\,(2x - n)$$

The $h_\varphi(n)$ is called *scaling function coefficients*, and h_φ is called a *scaling vector*. The equation is called the refinement equation , the *MRA equation* , or the *dilation equation*. It states that the expansion function of any subspace can be built from double resolution copies of themselves.

Example of Scaling Function

- The scaling function coefficients for the Haar function are $h_\varphi(0) = h_\varphi(1) = \frac{1}{\sqrt{2}}$, so the MRA equation is
  $$\varphi\,(x) = \frac{1}{\sqrt{2}}[\sqrt{2}\varphi\,(2x)] + \frac{1}{\sqrt{2}}[\sqrt{2}\varphi\,(2x-1)]$$
- After simplification, we obtain
  $$\varphi\,(x) = \ \varphi\,(2x) + \varphi\,(2x-1)$$

Wavelet Function

- Given a scaling function, we can define a *wavelet function* $\psi(x)$, together with its integer translates and binary scalings, spans the difference between any two adjacent scaling subspaces, V_j and V_{j+1}.
  We can define the set $\psi_{j,k}(x)$ of wavelets
  $$\psi_{j,\,k}(x) \ = \ 2^{j/2}\psi(2^j x \ - \ k)$$

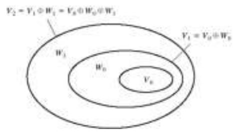


FIGURE 7.11 The relationship between scaling and wavelet function spaces.

- As with the scaling function, we define the wavelet subspace
  $$W_j \ = \ \overline{Span_k\{\psi_{j,k}(x)\}}$$
- Note that if $f(x) \in W_j$, we have
  $$f(x) = \sum_k \alpha_k \psi_{j,k}(x)$$

Relation between Scaling and Wavelet Subspaces

- The scaling and wavelet function subspaces are related by

$$V_{j+1} = V_j \oplus W_j$$

, where $\oplus$ denotes the union of spaces(like the union of sets)

- The orthogonal complement of V_j in V_{j+1} is W_j, and all members of V_j are orthogonal to the members of W_j, i.e.,

$$< \varphi_{j,k}(x), \psi_{j,k}(x) >= 0$$

for all appropriate $j, k, l \in \boldsymbol{Z}$

- the space of all measurable , square-integrable function as

$L^2(R) = V_0 \oplus W_0 \oplus W_1 \oplus \ldots$

or $L^2(R) = V_1 \oplus W_1 \oplus W_2 \oplus \ldots$

or $L^2(R) = \ldots \oplus W_{-2} \oplus W_{-1} \oplus W_0 \oplus W_1 \oplus W_2 \oplus \ldots$

Wavelet Function

- We have the generalized result

$$L^2(R) = V_{j_0} \oplus W_{j_0} \oplus V_{j_0+1} \oplus \ldots$$

where j_0 is an arbitrary starting scale

- Since wavelet spaces reside within the spaces spanned by the next higher resolution scaling funcions, any wavelet function can also be expressed as a weighted sum of shifted , double-resolution scaling functions, i.e.,

$$\psi(x) = \sum_n h_\psi(n)\sqrt{2}\varphi(2x - n)$$

where the $h_\psi(n)$ are called the wavelet function coefficients and h_ψ is the wavelet vector

- Using the condition that wavelets span the orthogonal complement spaces, and that integer wavelet translates are orthogonal, it can be shown that $h_\psi(n)$ is related to $h_\varphi(n)$ by (Burrus,Gpopinath, and Guo[1998])

$$h_\psi(n) = (-1)^n h_\varphi(2k - 1 - n)$$

an Example of Wavelet Function

- For Haar wavelet, the corresponding wavelet vector is $h_\psi(0) = (-1)^0 h_\varphi(1 - 0) = 1/\sqrt{2}$ and $h_\psi(1) = (-1)^1 h_\varphi(1 - 1) = -1/\sqrt{2}$

so the Haar wavelet function is $\psi(x) = \varphi(2x) - \varphi(2x - 1)$

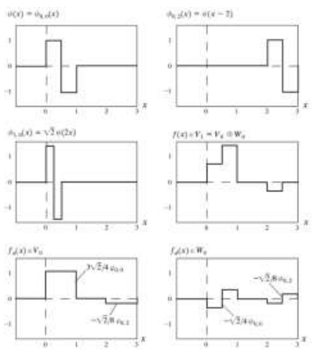


FIGURE 7.12 Haar wavelet functions in W_0 and W_1.

Outline

Wavelet Transforms in One Dimension

- We define the wavelet series expansion of function $f(x) \in L^2(R)$ relative to wavelet function $\psi(x)$ and scaling function $\phi(x)$ as

$$f(x) = \sum_k c_{j_0}(k)\,\varphi_{j_0,k}(x) + \sum_{j=j_0}^{\infty}\sum_k d_j(k)\,\psi_{j,k}(x)$$

where j_0 is an arbitrary starting scale, $c_{j_0}(k)$ are normally called the *approximation or scaling coefficients* and the $d_j(k)$ are called as the *detail or wavelet coefficients*.

- If the expansion functions form an orthonormal basis or tight frame, which is often the case, the expansion coefficents are calculated by

$$c_{j_0}(k) =< f(x), \varphi_{j_0,k}(x) >= \int f(x)\varphi_{j_0,k}(x)dx$$

and

$$d_{j_0}(k) =< f(x), \psi_{j,k}(x) >= \int f(x)\psi_{j,k}(x)dx$$

An Example of Wavelet Transforms in One Dimension

- Consider the simple function,

$$y = \begin{cases} x^2 & 0 \le x < 1 \\ 0 & orherwise \end{cases}$$

compute the expansion coefficients using Haar wavelet to represent it

Solution: Let $j_0 = 0$, we have

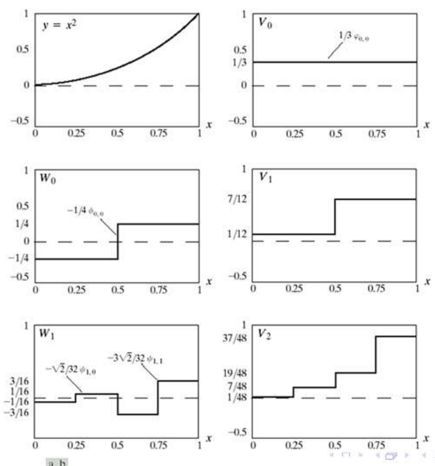
$$c_0(0) = \int_0^1 x^2\varphi_{0,0}(x)dx = \int_0^1 x^2dx = \frac{x^3}{3}|_0^1 = \frac{1}{3}$$

$$d_0(0) = \int_0^1 x^2\psi_{0,0}(x)dx = \int_0^{0.5} x^2dx - \int_{0.5}^1 x^2dx = -\frac{1}{4}$$

$$d_1(0) = \int_0^1 x^2\psi_{1,0}(x)dx = \int_0^{0.25} x^2\sqrt{2}dx - \int_{0.25}^{0.5} x^2\sqrt{2}dx = -\frac{\sqrt{2}}{32}$$

$$d_1(1) = \int_0^1 x^2\psi_{1,1}(x)dx = \int_{0.5}^{0.75} x^2\sqrt{2}dx - \int_{0.75}^{1} x^2\sqrt{2}dx = -\frac{3\sqrt{2}}{32}$$

An Example of Wavelet Transforms in One Dimension



Discrete Wavelet Transforms in One Dimension

- The wavelet series expansion maps a function of a continuous variable into a sequence of coefficients. If the function being expanded is a sequence of numbers , like samples of a continuous function $f(x)$, the resulting coefficients are called the *discrete wavelet transform* (DWT).

- the DWT transform pair is defined as

$$W_\psi(j_0,k) = \frac{1}{\sqrt{M}}\sum_x f(x)\varphi_{j_0,k}(x)$$

$$W_\psi(j,k) = \frac{1}{\sqrt{M}}\sum_x f(x)\varphi_{j,k}(x)$$

for $j \ge j_0$ and

$$f(x) = \frac{1}{\sqrt{M}}\sum_x W_\varphi(j_0,k)\varphi_{j_0,k}(x) + \frac{1}{\sqrt{M}}\sum_{j=j_0}^{\infty}\sum_x W_\psi(j_0,k)\psi_{j,k}(x)$$

Here $f(x)$, $\varphi_{j_0,k}(x)$, and $\psi_{j,k}(x)$ are functions of the discrete variable $x = 0, 1, \ldots, M-1$

Discrete Wavelet Transforms in One Dimension

- Consider the discrete function of four points: $f(0) = 1, f(1) = 4, f(2) = -3, and f(3) = 0$. We will use the Haar scaling and wavelet functions and assume that the four samples of $f(x)$ are distributed over the support of the basis functions, which is 1. With $j_0 = 0$, we can compute the DWT coefficients as

$$W_\varphi(0,0) = \frac{1}{2}\sum_{x=0}^{3} f(x)\varphi_{0,0}(x) = \frac{1}{2}[1\cdot 1 + 4\cdot 1 - 3\cdot 1 + 0\cdot 1] = 1$$

$$W_\psi(0,0) = \frac{1}{2}[1\cdot 1 + 4\cdot 1 - 3\cdot(-1) + 0\cdot(-1)] = 4$$

$$W_\psi(1,0) = \frac{1}{2}[1\cdot\sqrt{2} + 4\cdot(-\sqrt{2}) - 3\cdot 0 + 0\cdot 0] = -1.5\sqrt{2}$$

$$W_\psi(1,1) = \frac{1}{2}[1\cdot 0 + 4\cdot 0 - 3\cdot\sqrt{2} + 0\cdot(-\sqrt{2})] = -1.5\sqrt{2}$$

- Now we construct the original function from its transform.

$$f(x) = \frac{1}{2}[W_\varphi(0,0)\varphi_{0,0}(x) + W_\psi(0,0)\psi_{0,0}(x) + W_\psi(1,0)\psi_{1,0}(x) + W_\psi(1,1)\psi_{1,1}(x)]$$

- For $x = 0, f(0) = \frac{1}{2}[1\cdot 1 + 4\cdot 1 - 1.5\sqrt{2}\cdot\sqrt{2} - 1.5\sqrt{2}\cdot 0] = 1$

Continuous Wavelet Transforms in One Dimension

- The continuous wavelet transform of a continuous, square-integrable function, $f(x)$, relative to a real-value wavelet, $\psi(x)$, is

$$W_\psi(s,\tau) = \int_{-\infty}^{\infty} f(x)\psi_{s,\tau}(x)dx$$

where

$$\psi_{s,\tau}(x) = \frac{1}{\sqrt{s}}\psi\left(\frac{x-\tau}{s}\right)$$

and s and τ are called *scale* and *translation* parameters, respectively.

- Given $W_\psi(s,\tau), f(x)$ can be obtained using the *inverse continuous wavelet transform*:

$$f(x) = \frac{1}{C_\psi}\int_0^{\infty}\int_{-\infty}^{\infty} W_\psi(s,\tau)\frac{\psi_{s,\tau}(x)}{s^2}d\tau ds$$

Where $C_\psi = \int_{-\infty}^{\infty}\frac{|\Psi(u)|^2}{|u|}du$ and $\Psi(u)$ is the Fourier transform of $\psi(x)$

The Similarity between CWT and DWT

- The continuous translation parameter τ takes the place of the integer translation parameter k
- The continuous scale parameter, s is inversely related to the binary scale parameter 2^j
- The continuous transform is similar to a series expansion or discrete transform in which the starting scale $j_0 = -\infty$, so that the function is represented in terms of wavelets alone
- Like the discrete transform, the continuous transform can be viewed as a set of transform coefficients $W_\psi(s,\tau)$, that measure the similarity of $f(x)$ with a set of basis functions, $\psi_{s,\tau}(x)$. In the continuous case, however , both sets are infinite.

An Example of CWT: Mexican hat wavelet

$$\psi(x) = \frac{2}{\sqrt{3}}(\pi^{-1/4})(1-x^2)e^{x^2/2}$$

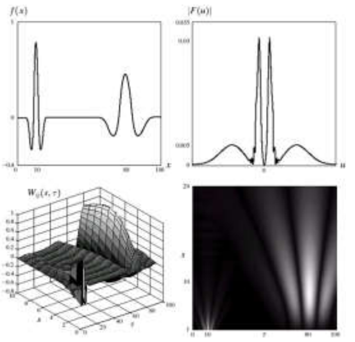
- It gets its name from its distinctive shape
- It is proportional to the second derivative of the Gaussian probability function , has an average value of zero and it compactly supported.
- Although it satisfies the admissibility requirement for the existence of continuous, reversible transforms, there is not an associated scaling function, and the computed transform does not result in an orthogonal analysis.
- Its most distinguishing features are its symmetry and the existence of the explicit expression.

An Example of CWT: Mexican hat wavelet(cont.)

The continuous, one-dimensional funtion is the sum of two Mexican hat wavelets:

$$f(x) = \psi_{1,10}(x) + \psi_{6,80}(x)$$

FIGURE 7.14 The continuous wavelet transform (c and d) and Fourier spectrum (b) of a continuous one-dimensional function (a).



The Fast Wavelet Transform

- The *fast wavelet transform* (FWT) is a computationally efficient implementation of the discrete wavelet transform (DWT) that exploit a surprising but fortunate relationship between the coefficients of the DWT at adjacent scales, also called *Mallat's herringbone algorithm* (Mallat[1989a,b]).
- Consider again the multiresolution refinement equation

$$\varphi(x) = \sum_n h_\varphi(n)\varphi(2x-n)$$

Scaling x by 2^j, tranlating it by k, and letting $m = 2k + n$ gives

$$\varphi(2^j x - k) = \sum_n h_\varphi(n)\sqrt{2}\varphi(2(2^j x - k) - n)$$
$$= \sum_m h_\varphi(m-2k)\sqrt{2}\varphi(2^{j+1}x - m)$$

The Fast Wavelet Transform(cont.)

- For wavelet function, we have the similar result
$\psi(2^j x - k) = \sum_m h_\psi(m-2k)\sqrt{2}\varphi(2^{j+1}x - m)$
- For discrete wavelet function, we have
$W_\psi(j,k) = \frac{1}{\sqrt{M}}\sum_x f(x)2^{j/2}\psi(2^j x - k)$
- Further, we have
$W_\psi(j,k) = \frac{1}{\sqrt{M}}\sum_x f(x)2^{j/2}[\sum_m h_\psi(m-2k)\sqrt{2}\varphi(2^{j+1}x - m)]$
- Interchanging the sum and integral and rearranging term then we get
$W_\psi(j,k) = \sum_m h_\psi(m-2k)\,[\frac{1}{\sqrt{M}}\sum_x f(x)2^{(j+1)/2}\varphi(2^{j+1}x - m)]$
- We can finally obtain

$$W_\psi(j,k) = \sum_m h_\psi(m-2k)\, W_\varphi(j+1,\ m)$$

- In the similar way, We find that

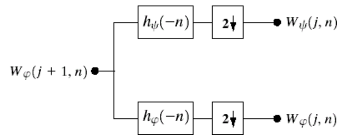
$$W_\varphi(j,k) = \sum_m h_\varphi(m-2k)\, W_\varphi(j+1,\ m)$$

The Fast Wavelet Transform(cont.)

$$W_\psi(j,k) = \sum_m h_\psi(m-2k)\, W_\varphi(j+1,\ m)$$
$$W_\varphi(j,k) = \sum_m h_\varphi(m-2k)\, W_\varphi(j+1,\ m)$$

- The above equations reveal a remarkable relationship between the DWT coefficients of adjacent scales.
- We can see that both $W_\varphi(j,k)$ and $W_\psi(j,k)$, the scale j approximation and the detail coefficients, can be computed by convolving $W_\psi(j+1,k)$, the scale $j+1$ approximation coefficient, with the time-revversed scaling and wavelet vectors, $h_\varphi(-n)$ and $h_\psi(-n)$, and subsampling the results.

FIGURE 7.15 An FWT analysis bank.



The Fast Wavelet Transform(cont.)

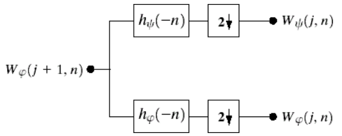
Just as the analysis portion of the two-band subband coding and decoding system, with and , we can write

$$W_\psi(j,k) = h_\psi(-n) * W_\varphi(j+1,\ n)|_{n=2k,k\geq 0}$$

$$W_\varphi(j,k) = h_\varphi(-n) * W_\varphi\ (j+1,\ n)|_{n=2k,k\geq 0}$$

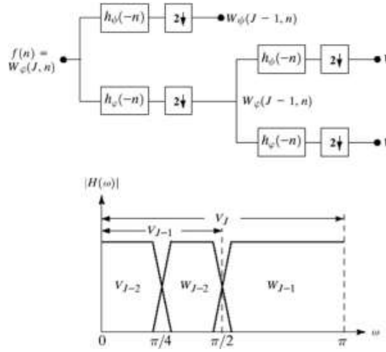
where the convolutions are evaluated at instants $n = 2k$ for $k \geq 0$

FIGURE 7.15 An FWT analysis bank.



The Fast Wavelet Transform (cont.)

- It should be noted that the filter bank can be "iterated" to create multistage structures for computing DWT coefficients at two or more sucessive scales, as shown in the figure



a
b
FIGURE 7.16 (a) A two-stage or two-scale FWT analysis bank and (b) its frequency splitting characteristics.

An Example of Fast Wavelet Transform

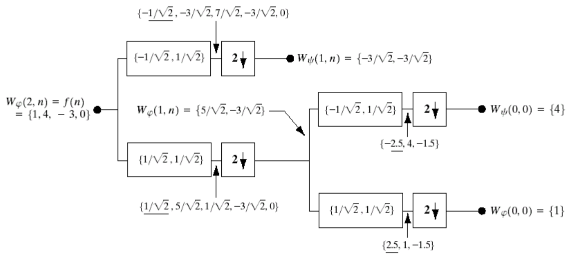


FIGURE 7.17 Computing a two-scale fast wavelet transform of sequence $\{1, 4, -3, 0\}$ using Haar scaling and wavelet vectors.

The Inverse Fast Wavelet Transform

- Since the perfect reconstruction (for two-band orthonormal filters) requires $g_i(n) = h_i(-n)$ for $i = 0, 1$, and the FWT analysis filters are $h_0(n) = h_\varphi(-n)$ and $h_1(n) = h_\psi(-n)$, the required inverse FWT synthesis filters are $g_0(n) = h_0(-n) = h_\varphi(n)$ and $g_1(n) = h_1(-n) = h_\psi(n)$.

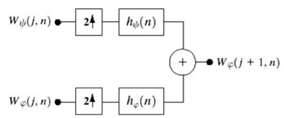


FIGURE 7.18 The FWT^{-1} synthesis filter bank.

- The inverse FWT filter bank implements the computation

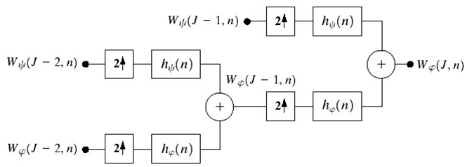
$$W_\varphi(j+1,k) = h_\varphi(k) * W_\varphi^{up}(j,k) + h_\psi(k) * W_\psi^{up}(j,k)$$

where W^{up} signifies upsampling by 2.

The Inverse Fast Wavelet Transform (cont.)

- As with the forward FWT, the inverse filter bank can also be iterated

FIGURE 7.19 A two-stage or two-scale FWT^{-1} synthesis bank.



An Example of Inverse Fast Wavelet Transform

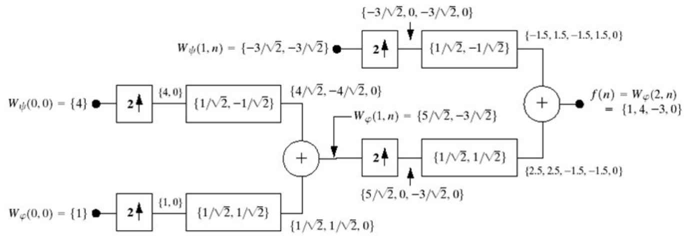
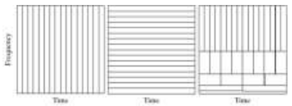


FIGURE 7.20 Computing a two-scale inverse fast wavelet transform of sequence $\{1, 4, -1.5\sqrt{2}, -1.5\sqrt{2}\}$ with Haar scaling and wavelet vectors.

The Differences between FFT and FWT

- The number of mathematical operation involved in the computation of the FWT of a length $M = 2^J$ sequence is on the order of $O(M)$. This compares favorably with the FFT algorithm, which requires $O(M \log M)$
- While the Fourier basis functions(i.e., sinusoids) guarantee the existence of the FFT , the existence of FWT depends on the availability of a scaling function for the wavelets being used, as well as the orthogonality (or biorthogonality) of the scaling function and corresponding wavelets.
- The FWT cannot analyze a function simultaneously in time and frequency, but FWT can do it.



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

The Fast Wavelet Transform

- Wavelet toolbox:
  [Lo_D,Hi_D,Lo_R,Hi_R]=wfilters(wname)

Wavelet	wfamily	wname
Haar	'haar'	'haar'
Daubechies	'db'	'db2', 'db3', ..., 'db45'
Coiflets	'coif'	'coif1', 'coif2', ..., 'coif5'
Symlets	'sym'	'sym2', 'sym3', ..., 'sym45'
Discrete Meyer	'dmey'	'dmey'
Biorthogonal	'bior'	'bior1.1', 'bior1.3', 'bior1.5', 'bior2.2', 'bior2.4', 'bior2.6', 'bior2.8', 'bior3.1', 'bior3.3', 'bior3.5', 'bior3.7', 'bior3.9', 'bior4.4', 'bior5.5', 'bior6.8'
Reverse Biorthogonal	'rbio'	'rbio1.1', 'rbio1.3', 'rbio1.5', 'rbio2.2', 'rbio2.4', 'rbio2.6', 'rbio2.8', 'rbio3.1', 'rbio3.3', 'rbio3.5', 'rbio3.7', 'rbio3.9', 'rbio4.4', 'rbio5.5', 'rbio6.8'

TABLE 7.1 Wavelet Toolbox FWT filters and filter family names.

The Fast Wavelet Transform

- To obtain a digital approximation of an orthonormal transform's scaling:
  [phi,psi,xval]=wavefun(wname,iter)
- Biorthogonal transforms:
  [phi1,psi1,phi2,psi2,xval]=wavefun(wname,iter)
- An Example:
  - [phi,psi,xval]=wavefun('haar',10);
  - xaxis=zeros(size(xval));
  - subplot(121);plot(xval,phi,'k',xval,xaxis,'–k');
  - axis([0 1 -1.5 1.5]);axis square;
  - title('haar scaling function');
  - subplot(122);plot(xval,psi,'k',xval,xaxis,'–k');
  - axis([0 1 -1.5 1.5]);axis square;
  - title('haar wavelet function');

The Fast Wavelet Transform

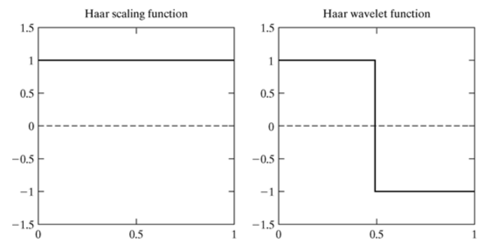


FIGURE 7.3 The Haar scaling and wavelet functions.

The Fast Wavelet Transform

- Computing the associated wavelet transform:
  [C,S]=wavedec2(X,N,Lo_D,Hi_D)
  or
  [C,S]=wavedec2(X,N,wname)
  where X: image or matrix
  N: the number of scales to be computed
  Lo_D,Hi_D: decomposition filters wname: value from table 7.1
- An Example:
  - f=magic(4);
  - [c1,s1]=wavedec2(f,1, 'haar');
  - [c2,s2]=wavedec2(f,2, 'haar');

The Fast Wavelet Transform

STATUS	Description
'sym'	The image is extended by mirror reflecting it across its borders. This is the normal default mode.
'zpd'	The image is extended by padding with a value of 0.
'spd', 'sp1'	The image is extended by first-order derivative extrapolation—or padding with a linear extension of the outmost two border values.
'sp0'	The image is extended by extrapolating the border values—that is, by boundary value replication.
'ppd'	The image is extended by periodic padding.
'per'	The image is extended by periodic padding after it has been padded (if necessary) to an even size using 'sp0' extension.

TABLE 7.2 Wavelet Toolbox image extension or padding modes.

The Fast Wavelet Transform

- FWTs without the wavelet toolbox
  - See the files wavefilter,wavefast.
  - An Example:
    - f=imread('Fig0704(Vase).tif');
    - [ratio,maxdifference]=fwtcompare(f,5,'db4');

Wavelet Transforms in Two Dimensions

- The on-dimensional transforms can be easily extended to two-dimensional functions like images. Concretely, the scaling function is

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

and the separable, "directionaly sentitive " wavelets

$$\psi^H = \psi(x)\varphi(y)$$
$$\psi^V = \varphi(x)\psi(y)$$
$$\psi^D = \psi(x)\psi(y)$$

- We define the two-dimensional scaled and translated basis function

$$\varphi_{j,m,n}(x,y) = 2^{j/2}\varphi(2^j x - m, 2^j y - n)$$
$$\psi^i_{j,m,n}(x,y) = 2^{j/2}\psi^i(2^j x - m, 2^j y - n), i = \{H,V,D\}$$

Wavelet Transforms in Two Dimensions(cont.)

- The discrete wavelet transforms of size $M \times N$ is

$$W_\varphi(j_0,m,n) = \frac{1}{\sqrt{MN}}\Sigma_{x=0}^{N-1}\Sigma_{x=0}^{M-1} f(x,y)\varphi_{j_0,m,n}(x,y)$$
$$W_\varphi^i(j,m,n) = \frac{1}{\sqrt{MN}}\Sigma_{x=0}^{M-1}\Sigma_{x=0}^{N-1} f(x,y)\psi^i_{j,m,n}(x,y)$$
$$i = \{H,V,D\}$$

- The inverse discrete wavelet transform is defined as :

$$f(x,y) = \frac{1}{\sqrt{MN}}\sum_{x=0}^{N-1}\sum_{x=0}^{M-1} W_\varphi(j_0,m,n)\varphi_{j_0,m,n}(x,y)$$
$$+\frac{1}{\sqrt{MN}}\sum_{i=H,V,D}\sum_{j=j_0}^{\infty}\sum_{m}\sum_{n} W_\psi^i(j,m,n)\varphi^i_{j,m,n}(x,y)$$

Computation of 2-D FWT

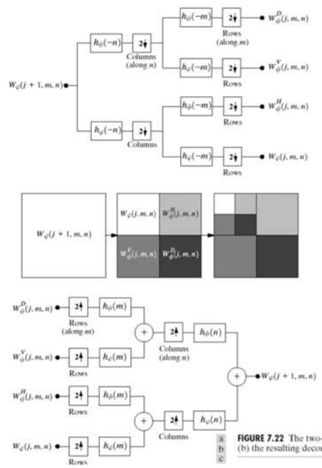
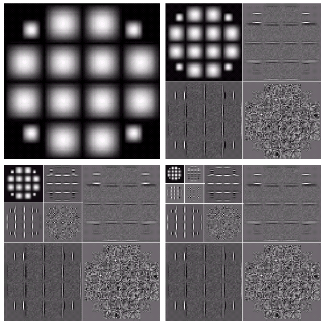


FIGURE 7.22 The two-dimensional fast wavelet transform: (a) the analysis filter bank; (b) the resulting decomposition; and (c) the synthesis filter bank.

An Example of 2-D FWT using Symlet Wavelet



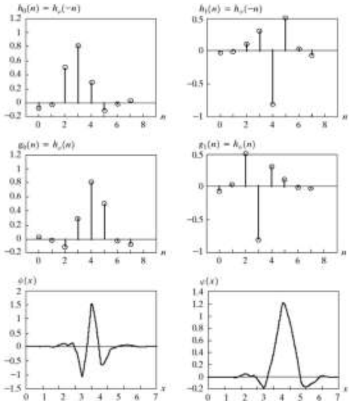
a b
c d

FIGURE 7.23 A three-scale FWT.

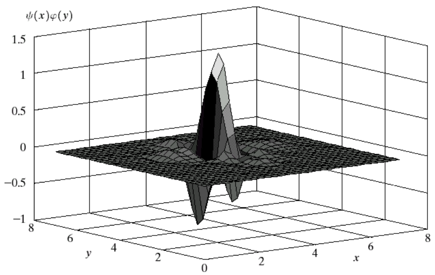
Scaling and Wavelet Function of Symlet Wavelet

a b
c d
e f
g

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, $\psi^H(x, y)$.



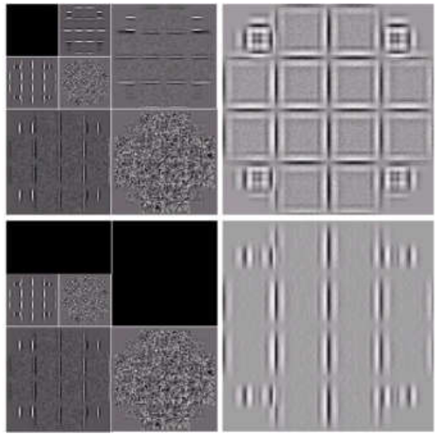
Scaling and Wavelet Function of Symlet Wavelet(cont.)



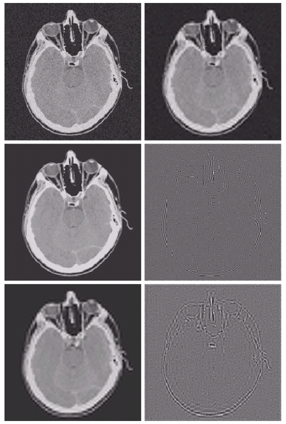
Application I: Modifying a DWT for Edge Detection

a b
c d

FIGURE 7.25 Modifying a DWT for edge detection: (a) and (c) two-scale decompositions with selected coefficients deleted; (b) and (d) the corresponding reconstructions.



Application II: Modifying a DWT for Noise Removel



a b
c d
e f

FIGURE 7.26 Modifying a DWT for noise removal: (a) a noisy MRI of a human head; (b), (c) and (e) various reconstructions after thresholding the detail coefficients; (d) and (f) the information removed during the reconstruction of (c) and (e). (Original image courtesy Vanderbilt University Medical Center.)

Working with wavelet decomposition structures

- c=[AN(:)′ HN(:)′ ⋯Hi(:)′ Vi(:)′ Di(:)′ ⋯V1(:)′ D1(:)′]
- S=[saN; sdN; sdN-1; ⋯sdi; ⋯sd1; sf]

Working with wavelet decomposition structures

- f=magic(8);
- [c1 s1]=wavedec2(f,3,’haar’);
- approx=appcoef(c1,s1,’haar’);
- horizdet2=detcoef2(’h’,c1,s1,2);
- newc1=wthcoef2(’h’,c1,s1,2);
- newhorizdet2=detcoef2(’h’,newc1,s1,2);

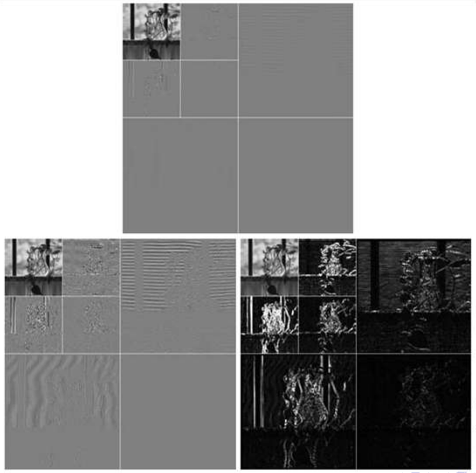
Working with wavelet decomposition structures

- a=appcoef2(c,s,wname);
- d=detcoef2(o,c,s,n);
- nc=wthcoef2(type,c,s,n,t,sorh);

Working with wavelet decomposition structures

- f=imread('Fig0704(Vase).tif');
- [c s]=wavefast(f,2,'db4');
- wave2gray(c,s);
- figure,wave2gray(c,s,8);
- figure,wave2gray(c,s,-8);

Working with wavelet decomposition structures



a
b c
FIGURE 7.5 Displaying a two-scale wavelet transform of the image in Fig. 7.4: (a) Automatic scaling; (b) additional scaling by 8; and (c) absolute values scaled by 8.

Outline

The inverse fast wavelet transform

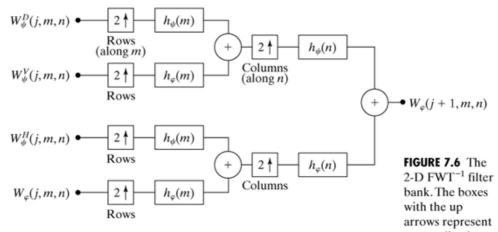


FIGURE 7.6 The 2-D FWT^{-1} filter bank. The boxes with the up arrows represent upsampling by inserting zeroes between every element.

Working with wavelet decomposition structures

- Wavelet Toolbox:
  g=waverec2(C,S,wname)
  or
  g=waverec2(C,S,Lo_R,Hi_R)

Outline

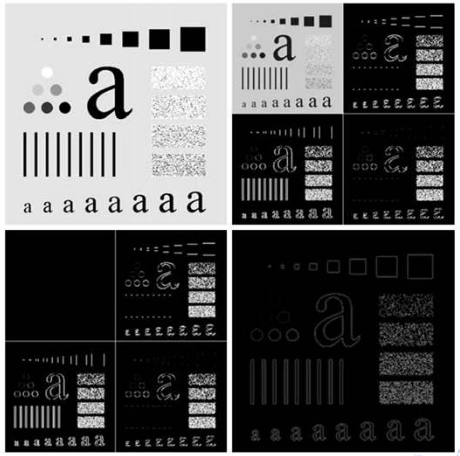
Wavelet in image processing

- Basic approach:
  - Compute the two-dim wavelet transform of an image.
  - Alter the transform coefficients.
  - Compute the inverse transform.

Wavelet in image processing

- f=imread('Fig0707(a)(Original).tif');
- subplot(221),imshow(f);
- [c s]=wavefast(f,1,'sym4');
- subplot(222),wave2gray(c,s,-6);
- [nc,y]=wavecut('a',c,s);
- subplot(223),wave2gray(nc,s,-6);
- edges=abs(waveback(nc,s,'sym4'));
- subplot(224),imshow(mat2gray(edges));

Wavelet in image processing



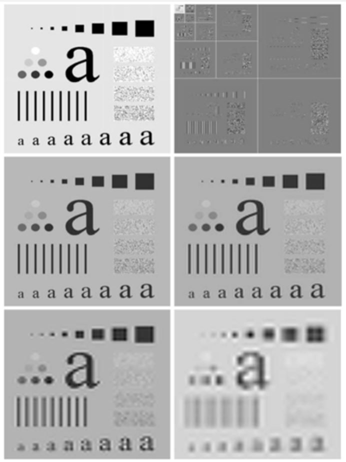
a b
c d

FIGURE 7.7 Wavelets in edge detection: (a) A simple test image; (b) its wavelet transform; (c) the transform modified by zeroing all approximation coefficients; and (d) the edge image resulting from computing the absolute value of the inverse transform.

Wavelet in image processing

- Smoothing and blurring
  - f=imread('Fig0707(a)(Original).tif');
  - [c s]=wavefast(f,4,'sym4');
  - wave2gray(c,s,20);
  - [c,g8]=wavezero(c,s,1,'sym4');
  - [c,g8]=wavezero(c,s,2,'sym4');
  - [c,g8]=wavezero(c,s,3,'sym4');
  - [c,g8]=wavezero(c,s,4,'sym4');

Wavelet in image processing



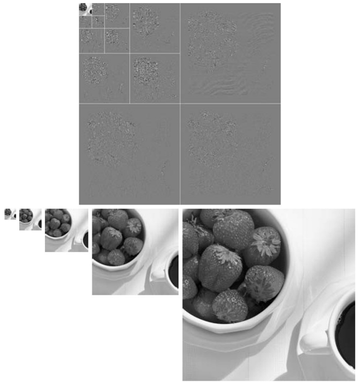
a b
c d
e f

FIGURE 7.8 Wavelet-based image smoothing: (a) A test image; (b) its wavelet transform; (c) the inverse transform after zeroing the first-level detail coefficients; and (d) through (f) similar results after zeroing the second-, third-, and fourth-level details.

Wavelet in image processing

- Progressive reconstruction
  - f=imread('Fig0709(original_strawberries).tif');
  - [c s]=wavefast(f,4,'jpeg9.7');
  - wave2gray(c,s,8);
  - f=wavecopy('a',c,s);
  - figure,imshow(mat2gray(f));
  - [c s]=waveback(c,s,'jpeg9.7',1);
  - f=wavecopy('a',c,s);
  - figure,imshow(mat2gray(f));

Wavelet in image processing



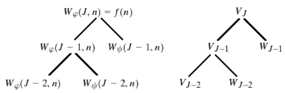
a
b c d e f

FIGURE 7.9 Progressive reconstruction: (a) A four-scale wavelet transform; (b) the fourth-level approximation image from the upper-left corner; (c) a refined approximation incorporating the fourth-level details; (d) through (f) further resolution improvements incorporating higher-level details.

Motivation of Wavelet Packets

- The fast wavelet transform decomposes a function into a series of logarithmically related frequency bands. That is, the low frequencies are grouped into narrow bands, while the high frequencies are grouped into wider bands.
- If we want greater control over the partitioning of the time-frequency plane, the FWT must be generalized to yield a more flexible decomposition-called a *wavelet packet*.

Subspace Analysis Tree



a b

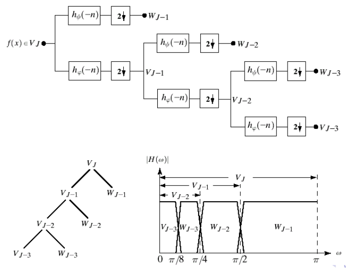
FIGURE 7.27 A coefficient (a) and analysis (b) tree for the two-scale FWT analysis bank of Fig. 7.16.

- The *root node* is assigned the highest-scale approximation coefficients, which are samples of the function itself, while the *leaves* inherit the transform's approximation and detail coefficient output.
- Replace the generating coefficients if Fig. 7.27(a) by the corresponding subspace, then we get the *subspace analysis tree*.
- Analysis trees provide a compact and informative way of representing multiscale wavelet transforms. They are simple to draw, take less space than their corresponding filter and subsampler-based block diagrams, and make it relatetively easy to spot valid decompositions.

An Three-scale Analysis Tree

- The three-scale analysis tree of Fig. 7.28(b), for example, offers the following three expansion options:

$$V_J = V_{J-1} \oplus W_{J-1}$$
$$V_J = V_{J-2} \oplus W_{J-2} \oplus W_{J-1}$$
$$V_J = V_{J-3} \oplus W_{J-3} \oplus W_{J-2} \oplus W_{J-1}$$



a
b c

FIGURE 7.28 A three-scale FWT filter bank: (a) block diagram; (b) decomposition space tree; and (c) spectrum splitting characteristics.

Wavelet Packet Three

- Analysis trees are also an efficient mechanism for representing *wavelet packets*, which are nothing more than *conventional wavelet transforms in which the details are iteratively filtered*.
- Thus, the three-scale FWT analysis tree of Fig. 7.28(b) becomes the three-scale *wavelet packet tree* of Fig. 7.29.

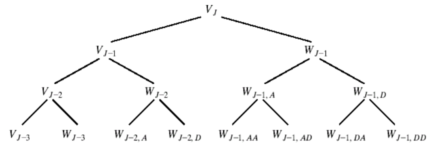


FIGURE 7.29 A three-scale wavelet packet analysis tree.

Wavelet Packet Three (cont.)

- The wavelet packet tree of Fig. 7.29 supports 26 different decompositions. For instance, V_J can be expanded as

$$V_J = V_{J-3} \oplus W_{J-3} \oplus W_{J-2,A} \oplus W_{J-2,D} \oplus W_{J-1,AA} \oplus W_{J-1,AD} \oplus W_{J-1,DA} \oplus W_{J-1,DD}$$

or

$$V_J = V_{J-1} \oplus W_{J-1,D} \oplus W_{J-1,AA} \oplus W_{J-1,AD}$$

- In general, P-scale, one-dimensional wavelet packet transforms(and associated P+1-level analysis trees) support

$$D(P+1) = [D(P)]^2 + 1$$

unique decompositions, where $D(1) = 1$.

Wavelet Packet Three (cont.)

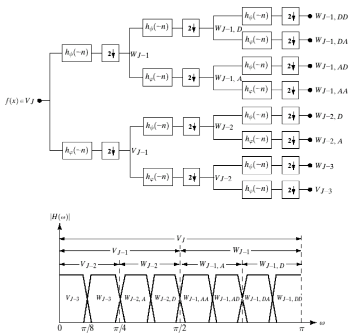


FIGURE 7.30 The (a) filter bank and (b) spectrum splitting characteristics of a three-scale full wavelet packet analysis tree.

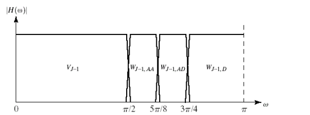
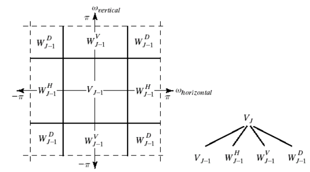


FIGURE 7.31 The spectrum of the decomposition in Eq. (7.6-5).

Two-dimensional Wavelet Packet Tree

- The two-dimensional, four-band filter bank splits approximation $W_\varphi(j+1,m,n)$ into outputs, $W_\varphi(j,m,n)$, $W_\psi^H(j,m,n)$, $W_\psi^V(j,m,n)$, $W_\psi^D(j,m,n)$. It can be "iterated" to generate P scale transforms for scales $j = J-1, \ldots, J-P$, with $W_\varphi(J,m,n) = f(m,n)$.

FIGURE 7.32 The first decomposition of a two-dimensional FWT: (a) the spectrum and (b) the subspace analysis tree.



- A P-scale, two-dimensional wavelet packet tree supports

$$D(P+1) = [D(P)]^4 + 1$$

unique expansions, where $D(1) = 1$.

Two-dimensional Wavelet Packet Tree

- Fig. 7.33 shows a portion of a three-scale, two-dimensional wavelet packet analysis tree.

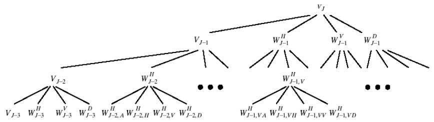
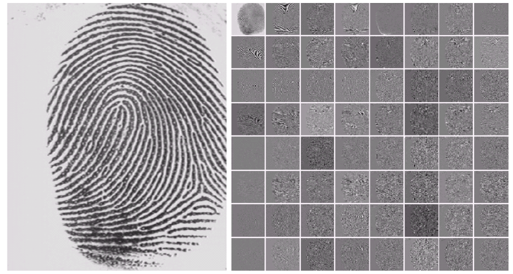


FIGURE 7.33 A three-scale, full wavelet packet decomposition tree. Only a portion of the tree is provided.

- A single wavelet packet tree presents numerous decomposition options. In fact the number of possible decomposition is often so large that it is impractical, if not impossible, to enumerate or examine them individually.
- An efficient algorithm for finding optimal decomposition with respect to application specific criteria is highly desirable. Classical entropy-based criteria are applicable in many situations and well suited to binary and quartenary tree searching algorithms.

An example of Compressing the Fingerprint Image

- Using three-scale wavelet packet trees, there are 83,522 potential decompositions that could serve as the starting point for the compression process. Fig. 7.34(b) shows one of them–a full wavelet packet, 64-leaf decomposition.



a b

FIGURE 7.34 (a) A scanned fingerprint and (b) its three-scale, full wavelet packet decomposition. (Original image courtesy of the National Institute of Standards and Technology.)

Wavelet Packets

- One reasonable criterion for selecting a decomposition for the compression of the image of Fig. 7.34(a) is the additive cost function

$$E(f) = \sum_{m,n} |f(m,n)|$$

- This function measures the entropy of information content of two-dimensional function f.

  The cost function is both computationally simple and easily adapted to tree optimization routines. The optimization algorithm must use the function to minimize the "cost" of the decomposition tree's leaf nodes.

Wavelet Packets

- For each node of the analysis tree, beginning with the root and proceeding level by level to the leaves:
  - Compute both the entropy of the node, denoted E_p(for parent entropy), and the entropy of its four offspring-denoted E_A, E_H, E_V and E_D. For two-dimensional wavelet packet decompositions, the parent is a two-dimensional wavelet packet decompositions, the parent is a two-dimensional array of approximation or detail coefficients; the offspring are the filtered approximation, horizontal, vertical,and diagonal details.
  - If the combined entropy of the offspring is less than the entropy of the parent–that is, $E_A + E_H + E_V + E_D < E_p$–include the offspring in the analysis tree. If the combined entropy of the offspring is greater than or equal to that of the parent, prune the offspring, keeping only the parent. It is a leaf of the optimized analysis tree.

Wavelet Packets

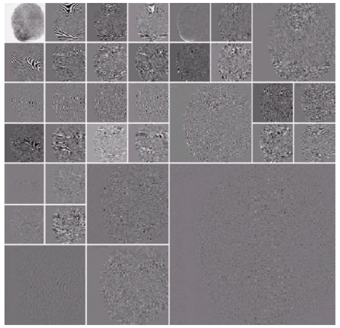


FIGURE 7.35 An optimal wavelet packet decomposition for the fingerprint of Fig. 7.34(a).

Wavelet Packets

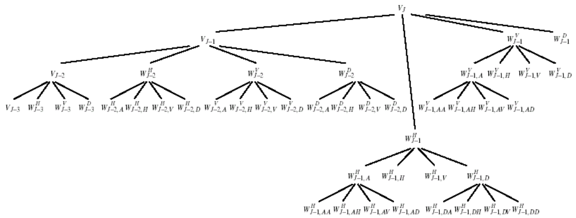


FIGURE 7.36 The optimal wavelet packet analysis tree for the decomposition in Fig. 7.35.