Wavelets

• Unlike the Fourier transform, whose basis functions are sinusoids, wavelet transforms are based on small waves of limited duration

• Applications
  – Image compression
  – Image denoising

• Background we need
  – Multiresolution image pyramids
  – Subband coding
  – Multiresolution analysis and scaling functions
Multiresolution

- Statistics of images such as histograms can vary significantly from one part of the image to another
- Small objects
  - Analyze at high-resolution
- Large objects
  - Analyze at low-resolution
- Need to analyze images at multiple resolutions

Image pyramids

\[ N = 2^j \]
\[ j = \log_2 N \]

- Level 0 is of little value
- Normally truncated at level \( P \)
  \[ j = J - P, \ldots, J \] (\( P+1 \) levels)

Approximation filter
- Gaussian
- Mean
- No filtering

Interpolation filter
- Nearest neighbor
- Bilinear interpolation

\[ f_{21}(n) = \begin{cases} f(n/2) & \text{if } n \text{ is even} \\ 0 & \text{otherwise} \end{cases} \]

\[ f_{21}(n) = f(2n) \]
Approximation and residual pyramids

- **Approximation pyramid**
  - 512 x 512 (j=9) to 64 x 64 (j=6)

- **Residual pyramid**
  - 64 x 64 approximation image at top of pyramid, rest are residuals
  - Higher compressibility
    - Fewer bits
Reconstruction from residual pyramid

Upsample 64 x 64 approximation image to 128 x 128
Apply interpolation filter to predict 128 x 128 image
Add 128 x 128 residual image (Now have 128 x 128 approximation image)

Upsample 128 x 128 approximation image from previous step to 256 x 256
Apply interpolation filter to predict 256 x 256 image
Add 256 x 256 residual image (Now have 256 x 256 approximation image)

Upsample 256 x 256 approximation image from previous step to 512 x 512
Apply interpolation filter to predict 512 x 512 image
Add 512 x 512 residual image

Exact reconstruction if the approximation/residual images were not quantized
Functionally related impulse responses

\[ h_1(n) \]

\[ h_2(n) = -h_1(n) \]

\[ h_3(n) = h_1(-n) \]

\[ h_4(n) = h_1(K - 1 - n) \]

\[ h_5(n) = (-1)^n h_1(n) \]

\[ h_6(n) = (-1)^n h_1(K - 1 - n) \]

**Figure 7.5** Six functionally related filter impulse responses: (a) reference response; (b) sign reversal; (c) and (d) order reversal (differing by the delay introduced); (e) modulation; and (f) order reversal and modulation.
Subband Coding

• Perfect reconstruction filters: $\hat{f}(n) = f(n)$

  – Biorthogonal filters
    • Two prototyped required
    • Cross modulation constraint:
      
      $g_0(n) = (-1)^n h_1(n)$
      
      $g_1(n) = (-1)^{n+1} h_0(n)$

  • Biorthogonality condition

    $$\langle h_i(2n-k), g_j(k) \rangle = \delta(i-j)\delta(n)$$

Subband Coding

- **Perfect reconstruction filters:**
  \[ \hat{f}(n) = f(n) \]

- **Orthonormal filters**
  - Given a single prototype filter \( g_0 \), remaining 3 can be computed to satisfy the orthonormality constraints

\[
\begin{align*}
g_1(n) &= (-1)^{n+1} g_0 \left( K_{\text{even}} - 1 - n \right) \\
h_i(n) &= g_i \left( K_{\text{even}} - 1 - n \right) \\
\langle g_i(n), g_j(n + 2m) \rangle &= \delta(i - j) \delta(m)
\end{align*}
\]
Daubechies filter

\[
g_1(n) = (-1)^{n+1} g_0(K_{even} - 1 - n)
\]

\[
h_i(n) = g_i(K_{even} - 1 - n)
\]
Two-dimensional subband coding

- Daubechies orthonormal filters
  - Prototype:

<table>
<thead>
<tr>
<th>n</th>
<th>$g_0(n)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0.23037781</td>
</tr>
<tr>
<td>1</td>
<td>0.71484657</td>
</tr>
<tr>
<td>2</td>
<td>0.63088076</td>
</tr>
<tr>
<td>3</td>
<td>-0.02798376</td>
</tr>
<tr>
<td>4</td>
<td>-0.18703481</td>
</tr>
<tr>
<td>5</td>
<td>0.03084138</td>
</tr>
<tr>
<td>6</td>
<td>0.03288301</td>
</tr>
<tr>
<td>7</td>
<td>-0.01059740</td>
</tr>
</tbody>
</table>
Reconstruction

1. Upsample columns of all subimages
2. Filter $a$ and $d^H$ along columns with $g_0$
3. Filter $d^V$ and $d^D$ along columns with $g_1$
4. Result of step 2: upsample rows, filter with $g_0$
5. Result of step 3: upsample rows, filter with $g_1$
6. Add result of steps 4 and 5
The Haar basis

- Recursively keep replacing approximation image with its decomposition
  - Stop at some level and keep approximation

- Properties:
  - Histograms of all detail images very similar
  - Can reconstruct image at various resolutions

\[ h_0 = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \end{bmatrix} \]

\[ h_1 = \begin{bmatrix} 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix} \]

Orthonormal Haar basis
Series expansions

Function $f(x) = \sum_{k} \alpha_k \varphi_k(x)$

Expansion set: Sines and cosines
Span: Periodic functions

$V = \text{Span}\left\{\varphi_k(x)\right\}$

Example Fourier series
- Expansion set: Sines and cosines
- Span: Periodic functions
Expansion coefficients

\[ f(x) = \sum_k \alpha_k \varphi_k(x) \quad \alpha_k = \langle \tilde{\varphi}(x), f(x) \rangle = \int \tilde{\varphi}_k(x)f(x)dx \]

\textit{Dual of basis function}

- Orthonormal basis (basis and dual equivalent):
  \[ \langle \varphi_j(x), \varphi_k(x) \rangle = \int \varphi_j^*(x)\varphi_k(x)dx = \delta_{jk} = \begin{cases} 0 & j \neq k \\ 1 & j = k \end{cases} \]

\textit{Inner product} \quad \textit{Defn. of inner product}

- Computation of expansion coefficients:
  \[ \alpha_k = \langle \varphi_k(x), f(x) \rangle \]
Scaling functions

Scaling function: Real, square-integrable prototype function

\[ \varphi_{j,k}(x) = 2^{j/2} \varphi \left( 2^j x - k \right) \]

Example (Haar):

- Width scaling changes shape
- Amplitude scaling makes sure that

\[ \left\langle \varphi_{j,k}, \varphi_{j,k} \right\rangle = 1 \]
Example: Haar scaling function

\[ \varphi(x) = \begin{cases} 
1 & 0 \leq x < 1 \\
0 & \text{otherwise} 
\end{cases} \]

\[ V_j = \text{Span} \left\{ \varphi_{j,k}(x) \right\} \]

- \( V_j \) is the span achieved by fixing \( j \) and varying \( k \)

- As \( j \) increases, the size of \( V_j \) increases to include functions with finer detail

\[ f(x) \in V_1 = 0.25 \varphi_{1,0}(x) + \varphi_{1,1}(x) - 0.25 \varphi_{1,4}(x) \]
Nested function spaces

- All $V_0$ expansion functions are contained in $V_1$
- All $V_j$ expansion functions are contained in $V_{j+1}$
- Any function in $V_j$ is also in $V_{j+1}$

$$
\varphi_{0,k}(x) = \frac{1}{\sqrt{2}} \varphi_{1,2k}(x) + \frac{1}{\sqrt{2}} \varphi_{1,2k+1}(x)
$$
Multiresolution analysis requirements

1. The scaling function is orthogonal to its integer translates
2. The subspaces spanned by the scaling function at low scales are nested within those spanned at higher scales
3. The only function common to all \( V_j \) is \( f(x) = 0 \) \( \quad V_{-\infty} = \{0\} \)
4. Any square integrable function can be represented with arbitrary precision:
   \[
   V_{\infty} = \left\{ L^2(\mathbb{R}) \right\}
   \]
   - Haar scaling function obeys all these requirements
   - Under these conditions: \( \varphi_{j,k}(x) = \sum_{n} \alpha_n \varphi_{j+1,n}(x) \)
Refinement equation

\[ \varphi_{j,k}(x) = \sum_n \alpha_n \varphi_{j+1,n}(x) \]

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

Substitute \( \varphi_{j+1,n}(x) = 2^{j+1/2} \varphi(2^{j+1} x - n) \)

and change \( \alpha_n \) to \( h_\varphi(n) \), then

\[ \varphi_{j,k}(x) = \sum_n h_\varphi(n)2^{j+1/2} \varphi(2^{j+1} x - n) \]

set \( j = k = 0 \), also note that \( \varphi_{0,0}(x) = \varphi(x) \)

\[ \varphi(x) = \sum_n h_\varphi(n)\sqrt{2}\varphi(2x - n) \]

Refinement equation:
Expansion functions can be built from double resolution copies of themselves
Refinement equation example

- Haar function
  \[ \varphi(x) = \begin{cases} 
  1 & 0 \leq x < 1 \\
  0 & \text{otherwise}
  \end{cases} \]

- Scaling coefficients
  \[ h_\varphi(0) = h_\varphi(1) = \frac{1}{\sqrt{2}} \]

- Refinement equation
  \[ \varphi(x) = \sum_n h_\varphi(n)\sqrt{2}\varphi(2x - n) \]
  \[ \varphi(x) = \varphi(2x) + \varphi(2x - 1) \]
Wavelet functions

• Wavelet functions span the difference between adjacent scaling subspaces $V_j$ and $V_{j+1}$

  – Given a scaling function that meets the requirements discussed, we can design a wavelet function

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

Wavelet function

$$W_j = \text{Span}_k \left\{ \psi_{j,k}(x) \right\}$$

Amplitude scaling  Scaling of width  Position along x-axis

Function representation

\[ L^2(\mathbb{R}) = V_0 \oplus W_0 \oplus W_1 \oplus \ldots \]
\[ L^2(\mathbb{R}) = V_1 \oplus W_1 \oplus W_2 \oplus \ldots \]
\[ L^2(\mathbb{R}) = V_{j_0} \oplus W_{j_0} \oplus W_{j_0+1} \oplus \ldots \]

\[ V_2 = V_1 \oplus W_1 = V_0 \oplus W_0 \oplus W_1 \]

\[ V_1 = V_0 \oplus W_0 \]
Generating wavelet functions

\[ \psi(x) = \sum_n h_\psi(n) \sqrt{2} \varphi(2x - n) \] because \( V_{j+1} = V_j \oplus W_j \)

\[ \langle \varphi_{j,k}(x), \psi_{j,l}(x) \rangle = 0 \quad \text{For any } k, l \]

\[ h_\psi(n) = (-1)^n h_\varphi(1 - n) \] \( \text{Wavelet function coefficients} \)

Modulation, time reversal

• For the Haar scaling function:

\[ h_\varphi(0) = h_\varphi(1) = \frac{1}{\sqrt{2}} \]

• Then:

\[ h_\psi(0) = (-1)^0 h_\varphi(1 - 0) = \frac{1}{\sqrt{2}} \]

\[ h_\psi(1) = (-1)^1 h_\varphi(1 - 1) = - \frac{1}{\sqrt{2}} \]
Haar wavelet function

\[
\psi(x) = \sum_{n} h_{\psi}(n) \sqrt{2}\varphi(2x - n)
\]

Substitute \( h_{\psi}(0) = \frac{1}{\sqrt{2}} \) and \( h_{\psi}(1) = -\frac{1}{\sqrt{2}} \)

\[
\psi(x) = \varphi(2x) - \varphi(2x - 1)
\]

\[
\psi(x) = \begin{cases} 
1 & 0 \leq x < 0.5 \\
-1 & 0.5 \leq x < 1 \\
0 & \text{otherwise}
\end{cases}
\]
Haar wavelet function

\[
\psi(x) = \begin{cases} 
1 & 0 \leq x < 0.5 \\
-1 & 0.5 \leq x < 1 \\
0 & \text{otherwise}
\end{cases}
\]

\[
f(x) = f_a(x) + f_d(x)
\]

- \(f(x)\)
  - In space \(V_1\)
- \(f_a(x)\) approximation
  - In space \(V_0\)
- \(f_d(x)\) difference
  - In space \(W_0\)

\[
V_1 = V_0 \oplus W_0
\]

<table>
<thead>
<tr>
<th>Wavelet Series</th>
<th>Fourier Series</th>
</tr>
</thead>
<tbody>
<tr>
<td>Continuous Wavelet</td>
<td>(Continuous) Fourier</td>
</tr>
<tr>
<td>Transform</td>
<td>Transform</td>
</tr>
<tr>
<td>Discrete Wavelet</td>
<td>Discrete Fourier</td>
</tr>
<tr>
<td>Transform</td>
<td>Transform</td>
</tr>
</tbody>
</table>
Wavelet Series

\[ f(x) = \sum_{k} c_{j_0}(k) \varphi_{j_0,k}(x) + \sum_{j=j_0}^{\infty} \sum_{k} d_{j}(k) \psi_{j,k}(x) \]

- The first sum uses scaling functions to provide an approximation of \( f(x) \) at a chosen starting scale \( j_0 \). This sum is over translations (k) only.
- The second sum is over scales (greater than \( j_0 \)) and over translations (k). It provides ever increasing detail (finer resolution).
Wavelet Series Coefficients

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

\[ c_{j_0}(k) = \int f(x)\varphi_{j_0,k}(x)dx \]

\[ d_j(k) = \int f(x)\psi_{j,k}(x)dx \]

- This assumes an orthonormal basis (or a tight frame) which is often the case.
Wavelet Series Example

\[ y = x^2 \]

\[ V_0 \]

\[ W_0 \]

\[ V_1 \]

\[ W_1 \]

\[ V_2 \]
Discrete Wavelet Transform (DWT)

- Like the DFT, the DWT operates on discrete functions (length \(M=2^J\)).

\[
f(n) = \frac{1}{\sqrt{M}} \sum_{k=0}^{2^j-1} W_\varphi (j_0, k) \varphi_{j_0,k}(n) + \frac{1}{\sqrt{M}} \sum_{j=j_0}^{J-1} \sum_{k=0}^{2^{j+1}-1} W_\psi (j, k) \psi_{j,k}(n)
\]

**Sampled scaling function**

\[
W_\varphi (j_0, k) = \frac{1}{\sqrt{M}} \sum_{n=0}^{M-1} f(n) \varphi_{j_0,k}(n)
\]

**Sampled wavelet function**

\[
W_\psi (j, k) = \frac{1}{\sqrt{M}} \sum_{n=0}^{M-1} f(n) \psi_{j,k}(n) \quad \text{for} \ j \geq j_0
\]
Haar transform

\[ \varphi(x) = \begin{cases} 1 & 0 \leq x < 1 \\ 0 & \text{otherwise} \end{cases} \]

\[ \psi(x) = \begin{cases} 1 & 0 \leq x < 0.5 \\ -1 & 0.5 \leq x < 1 \\ 0 & \text{otherwise} \end{cases} \]

\[ W_\varphi(0,k) = \frac{1}{\sqrt{M}} \sum_n f(n) \varphi_{0,k}(n) \]

\[ W_\psi(j,k) = \frac{1}{\sqrt{M}} \sum_n f(n) \psi_{j,k}(n) \]

\[ \varphi_{jk}(x) = 2^{j/2} \varphi(2^j x - k) \]

sample \( \varphi_{jk}(x) \) to get \( \varphi_{jk}(n) \)

\[ \psi_{jk}(x) = 2^{j/2} \psi(2^j x - k) \]

sample \( \psi_{jk}(x) \) to get \( \psi_{jk}(n) \)

\[ T = HFH^T \]

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

\[ H_4 = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix} \]
Fast Wavelet Transform (FWT)

• It can be shown that

\[ W_\psi(j, k) = \sum_{m} h_\psi(m - 2k)W_\varphi(j + 1, m) \]

\[ W_\varphi(j, k) = \sum_{m} h_\varphi(m - 2k)W_\varphi(j + 1, m) \]

• This is a very useful relationship between the DWT coefficients at adjacent levels.

Fast Wavelet Transform

Coefficients at highest scale are samples of the function itself.

Another iteration would split $V_{J-2}$ into 1/8th frequency bands.
Fast Wavelet Transform

Do this:

Coefficients at highest scale are samples of the function itself

Instead of this:

\[
W_\varphi(j_0, k) = \frac{1}{\sqrt{M}} \sum_{n=0}^{M-1} f(n)\varphi_{j_0,k}(n)
\]

\[
W_\psi(j, k) = \frac{1}{\sqrt{M}} \sum_{n=0}^{M-1} f(n)\psi_{j,k}(n) \text{ for } j \geq j_0
\]
Haar FWT example

$W_\varphi(2, n) = f(n) = \{1, 4, -3, 0\}$

$W_\varphi(I, n) = \{5/\sqrt{2}, -3/\sqrt{2}\}$

$W_\varphi(1, n) = \{-3/\sqrt{2}, -3/\sqrt{2}\}$

$W_\varphi(0, 0) = \{4\}$

$W_\psi(0, 0) = \{1\}$

$h_\psi(0) = \frac{1}{\sqrt{2}}$

$h_\psi(1) = -\frac{1}{\sqrt{2}}$

$h_\varphi(0) = h_\varphi(1) = \frac{1}{\sqrt{2}}$

$h_\psi(0) = h_\psi(1) = \frac{1}{\sqrt{2}}$

$h_\varphi(0) = h_\varphi(1) = \frac{1}{\sqrt{2}}$

$\phi(0, n) = \{1/\sqrt{2}, 5/\sqrt{2}, 1/\sqrt{2}, -3/\sqrt{2}, 0\}$

$\phi(I, n) = \{2.5, 1, -1.5\}$

$\phi(0, 0) = \{2.5, 4, -1.5\}$
Inverse FWT

Do this:

\[ f(n) = \frac{1}{\sqrt{M}} \sum_{k=0}^{2^{j_0} - 1} W_\varphi (j_0, k) \varphi_{j_0,k}(n) + \frac{1}{\sqrt{M}} \sum_{j=j_0}^{J-1} \sum_{k=0}^{2^j-1} W_\psi (j, k) \psi_{j,k}(n) \]

Instead of this:
Example: Haar inverse FWT

Do this:

\[ f(n) = \frac{1}{\sqrt{M}} \sum_{k=0}^{2^{j_0}-1} W_\varphi (j_0, k) \varphi_{j_0,k}(n) + \frac{1}{\sqrt{M}} \sum_{j=j_0}^{J-1} \sum_{k=0}^{2^j-1} W_\psi (j, k) \psi_{j,k}(n) \]

\[ J = 2, j_0 = 0 \]
**Figure 7.23** Time-frequency tilings for the basis functions associated with (a) sampled data, (b) the FFT, and (c) the FWT. Note that the horizontal strips of equal height rectangles in (c) represent FWT scales.
2D Wavelet transform

• Separable 2D scaling function
  \( \varphi(x,y) = \varphi(x)\varphi(y) \)

• Directionally sensitive separable 2D wavelet functions
  \( \psi^H(x,y) = \psi(x)\varphi(y) \) : horizontal detail
  \( \psi^V(x,y) = \varphi(x)\psi(y) \) : vertical detail
  \( \psi^D(x,y) = \psi(x)\psi(y) \) : diagonal detail
Scaled and translated basis functions

\[ \varphi_{j,m,n}(x,y) = 2^{j/2} \varphi(2^j x - m, 2^j y - n) \]

\[ \psi_{j,m,n}^H(x,y) = 2^{j/2} \psi^H(2^j x - m, 2^j y - n) \]

\[ \psi_{j,m,n}^V(x,y) = 2^{j/2} \psi^V(2^j x - m, 2^j y - n) \]

\[ \psi_{j,m,n}^D(x,y) = 2^{j/2} \psi^D(2^j x - m, 2^j y - n) \]

- We can enumerate the last 3 equations as

\[ \psi_{j,m,n}^i(x,y) = 2^{j/2} \psi^i(2^j x - m, 2^j y - n), \quad i = \{H, V, D\} \]
2D Wavelet Transform

- **Forward transform**

\[
W_{\varphi}(j_O,m,n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \varphi_{j_O,m,n}(x,y)
\]

\[
W_{\psi}^i(j,m,n) = \frac{1}{\sqrt{MN}} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x,y) \psi_{j,m,n}^i(x,y), \quad i = \{H,V,D\}
\]

- **Inverse transform**

\[
f(x,y) = \frac{1}{\sqrt{MN}} \sum_{m=0}^{2^{j_0}-1} \sum_{n=0}^{2^{j_0}-1} W_{\varphi}(j_O,m,n) \varphi_{j_O,m,n}(x,y) + \\
\frac{1}{\sqrt{MN}} \sum_{i=\{H,V,D\}} \sum_{j=j_0}^{J-1} \sum_{m=0}^{2^{j_0}-1} \sum_{n=0}^{2^{j_0}-1} W_{\psi}(j,m,n) \psi_{j,m,n}^i(x,y)
\]

\[
N = M = 2^j \\
j = j_0, \ldots, J - 1 \\
m = 0, 1, 2, \ldots, 2^j - 1 \\
n = 0, 1, 2, \ldots, 2^j - 1
\]
2D Wavelet Transform Implementation

\[ W_\psi(j + 1, m, n) \]

\[ \star h_\psi(-n) \]

\[ 2\downarrow \]

Columns (along \( n \))

\[ \star h_\psi(-m) \]

\[ 2\downarrow \]

Rows (along \( m \))

\[ W_\psi^D(j, m, n) \]

\[ W_\psi(j, m, n) \]

\[ \star h_\psi(-n) \]

\[ 2\downarrow \]

Columns

\[ \star h_\psi(-m) \]

\[ 2\downarrow \]

Rows

\[ W_\psi^V(j, m, n) \]

\[ W_\psi(j, m, n) \]

\[ \star h_\psi(-m) \]

\[ 2\downarrow \]

Rows

\[ W_\psi^H(j, m, n) \]

\[ W_\psi(j, m, n) \]
Example: DWT with various $j_0$

- Top left:
  - 128 x 128 image, $J=7$
- Top right:
  - 2D DWT $j_0=6$
- Bottom left:
  - 2D DWT $j_0=5$
- Bottom right:
  - 2D DWT $j_0=4$
Wavelet image processing

- Similar to processing in frequency domain
- Approach
  1. Compute 2D DWT of input image
  2. Alter the transform
  3. Compute inverse 2D DWT to get output image
Example: thresholding wavelet coefficients

- First row: edge detection
- Second row: vertical edge detection

Wavelet image denoising

• Reduce influence of higher scale detail coefficients
  – Set to 0 all the detail wavelet coefficients that fall below a threshold level
  – How many detail levels to threshold?
  – Should the threshold be the same for all levels?
  – Soft vs. hard thresholding

Example:
Two highest levels thresholded with an interactively chosen global threshold
Throwing away detail coefficients

• Throwing away the highest scale detail coefficients
  – Noise reduced
  – Edges preserved

• Throwing away 2 highest scale detail coefficients
  – Some detail and edges lost
Wavelet Compression

• Wavelets pack the most important visual information into a small number of coefficients. The remaining coefficients can be quantized coarsely or truncated with little loss of visual information

• JPEG: Quantizes Discrete Cosine Transform on 8x8 blocks.
  – Need to subdivide into blocks. Creates a blocking effect

• JPEG-2000: Quantized discrete wavelet transforms. Provides increased compression ratios.
  – Wavelets are inherently local basis functions. No need to subdivide image. No blocking effect
Wavelet selection

FIGURE 8.46
Three-scale wavelet transforms of Fig. 8.9(a) with respect to
(a) Haar wavelets,
(b) Daubechies wavelets,
(c) symlets, and
(d) Cohen-Daubechies biorthogonal wavelets.
Decomposition level selection

- Fixed global threshold of 25 applied to detail coefficients (not to approximation coefficients)
- Different levels of wavelet decomposition levels can be used
  - Initial levels provide the most gain

<table>
<thead>
<tr>
<th>Decomposition Level (Scales or Filter Bank Iterations)</th>
<th>Approximation Coefficient Image</th>
<th>Truncated Coefficients (%)</th>
<th>Reconstruction Error (rms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>256 × 256</td>
<td>74.7%</td>
<td>3.27</td>
</tr>
<tr>
<td>2</td>
<td>128 × 128</td>
<td>91.7%</td>
<td>4.23</td>
</tr>
<tr>
<td>3</td>
<td>64 × 64</td>
<td>95.1%</td>
<td>4.54</td>
</tr>
<tr>
<td>4</td>
<td>32 × 32</td>
<td>95.6%</td>
<td>4.61</td>
</tr>
<tr>
<td>5</td>
<td>16 × 16</td>
<td>95.5%</td>
<td>4.63</td>
</tr>
</tbody>
</table>
• Detail images have a very peaked probability distribution around zero which allows for a small number of quantization levels
Quantizing wavelet coefficients

- Coefficients need to be quantized
- Can use an uniform quantizer
- Better results can be achieved by
  - Introducing a larger quantization interval around zero
  - Adapting the size of the quantization intervals from scale to scale (JPEG-style scaling)
JPEG-2000