Digital Image Processing
Chapter 7: Wavelets

Prepared by:

Dr. Ali J. Abboud
University of Diyala
2012-2013
Key Features of Chapter 7:

- Fourier Analysis – Shortcomings.
- Wavelet Transforms.
- CWT and DWT.
- One Dimension (1D) DWT.
- Multi-Resolution 2D Wavelet Transforms.
- Different Decomposition Schemes.
- Statistical Properties of Wavelet subbands.
- Applications of Wavelet Transforms.
- DWT Filters Types in Matlab.
- DWT Functions in Matlab.
7.1 Fourier Analysis – Shortcomings

- Fourier analysis of the two signals, below, give the same answer
- Thus FT does not provide spatial support, i.e. cannot provide frequency and time/space information simultaneously

\[ f(x) = \text{concatenation of } \cos x, \cos 2x, \text{ and } \cos 3x \]

\[ g(x) = \frac{\cos x + \cos 2x + \cos 3x}{3} \]
7.2 Short-Time Fourier Transform (STFT)

To overcome the above problem, STFT uses the Fourier transform on small window.

- It maps a signal into a two-dimensional function of time and frequency, i.e. provides some information about both when and at what frequencies a signal event occurs.

- The drawback is that once you choose a particular size for the time window, it remains the same for all frequencies. Many signals require a more flexible approach -- one where we can vary the window size.

- Wavelet Transforms represents the next logical step: a windowing technique with variable-sized regions.
Wavelet transforms

Wavelet analysis allows the use of long time intervals where we want more precise low-frequency information, and shorter regions where we want high-frequency information.

A wavelet (i.e. small wave) is a mathematical function used to analyze a continuous-time signal into different frequency components and study each component with a resolution that matches its scale.

A wavelet transform is the representation of a function by wavelets. The wavelets are scaled and translated copies of a finite-length or fast-decaying oscillating waveform \( \psi(t) \), known as the mother wavelet.

There are many wavelet filters to choose from.
7.3 Wavelet Transforms

- WT provides powerful insight into an image’s spatial and frequency characteristics. However, The FT exposes only an image’s frequency attributes.

- WT became the preferred image transform for various reasons:
  - Localization.
  - Lossless Transform.
  - Multi-resolution Characteristics.

- Some WT applications for a variety of image processing/analysis:
  - Feature preserving of image/video quantization for compression.
  - Content based video retrieval.
  - Feature extraction for face detection.
  - Image watermarking and steganography (i.e. information security).
  - Object authentication/recognition.
7.4 CWT and DWT

- The continuous wavelet transform (CWT) of \( f(t) \) is defined as the sum over all time multiplied by a scaled, shifted versions of the wavelet function \( \psi(t) \):

\[
C(\text{scale, position}) = \int f(x) \psi(s, p, t) \, dt.
\]

where \( \psi(s,p,t) \) is the scaled and mother wavelet. The best choice for the scale and position is to use multiple powers of 2.

- Computing CWT is rather inefficient. The Discrete wavelet transform (DWT) however provides a compact representation of a signal’s frequency components with strong spatial support.
7.5 One Dimension (1D) DWT

- The Wavelet transform is a short time analysis tool of finite energy quasi-stationary signals at multi-resolutions.
- The Discrete wavelet transform (DWT) provide a compact representation of a signal’s frequency components with strong spatial support.
- DWT decomposes a signal into frequency subbands at different scales from which it can be perfectly reconstructed.
- For example one dimension wavelet transform is shown below
7.5.1 Example: The Haar Wavelet Filter

- The *Haar wavelet* is a discontinuous, and resembles a step function.

- It is a crude version of the Truncated cosine.

- It can be implemented using a simple filter:

If \( X = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\} \) is a time-signal of length 8, then the Haar wavelet decomposes \( X \) into an *approximation* subband containing the \textbf{Low} frequencies and a \textbf{detail} subband containing the \textbf{high} frequencies:

\[
\text{Low} = \{x_2 + x_1, x_4 + x_3, x_6 + x_5, x_8 + x_7\}/\sqrt{2}
\]

\[
\text{High} = \{x_2 - x_1, x_4 - x_3, x_6 - x_5, x_8 - x_7\}/\sqrt{2}
\]
7.6 Multi-Resolution 2D Wavelet Transforms

A Haar wavelet decomposes images first on the rows and then on the columns resulting in 4 subbands, the LL-subband which is an approximation of the original image while the other subbands contain the missing details.

The LL-subband output from any stage can be decomposed further.
7.6 Multi-Resolution 2D Wavelet Transforms
7.6 Multi-Resolution 2D Wavelet Transforms

LL sub-band: Representing the low frequencies in both horizontal and vertical directions. This sub-band is also called the approximation or the scaled sub-band.

LH sub-band: Representing the high frequencies in the horizontal direction and the low frequencies in the vertical one. Hence, this sub-band holds information about horizontal features (e.g. edges) in the image.

HL sub-band: Representing the high frequencies in the vertical direction and the low frequencies in the horizontal one. It holds information about vertical features in the image.

HH sub-band: Representing the highest frequencies in the image, and holds information about diagonal features in the image.
7.6. Multi-Resolution Wavelet Transforms - Example

(a) Decomposition stage 1

(b) Decomposition stage 2

(a) Original  (b) Stage 1  (c) Stage 2  (d) Stage 3
7.6. Multi-Resolution Wavelet Transforms – Example2

Figure 3.7 Example of applying different wavelet filters (db1, db2, db3 and Meyer) for B, C, D and E respectively, where (G, H, I and J shows the histogram for the corresponding LL2 sub-bands.
7.7 Different Decomposition Schemes

- The previous 2 decomposition scheme is known as the Pyrimad scheme, whereby at successive stages only the LL subband is wavelet transformed.

- Other decomposition schemes include:
  - The *standard* scheme – At every stage all the image is wavelet transformed
  - The *wavelet packet* – After stage 1, a non-LL subband is transformed only if it satisfied certain condition.
7.8. Statistical Properties of Wavelet subbands

The distribution of the LL-subband approximate that of the original but all non-LL subbands have a Laplacian distribution. This remains valid at all depths.
The list of applications is growing fast. These include:

- Image and video Compression
- Feature detection and recognition
- Image denoising
- Face Recognition
- Signal interpolation

Most applications benefit from the statistical property of the non-LL subbands (The laplacian distribution of the wavelet coefficients in these subbands).
7.9.1 Wavelet-based Feature Detection

- Non-LL subbands of a wavelet decomposed image contains high frequencies (i.e. image features) which are highlighted. These significant coefficients are the furthest away from the mean.

- Thresholding reveals the main features.

Horizontal features

Vertical features
7.9.2 Image and Video Compression

Compression is done in 4 steps:

1. Transform into frequency domain (Fourier or Wavelet) the image to the required depth

2. Quantise each subband according to the required ratio

3. Create a code **Entropy-based** code book - the more frequent quantised coefficients have shorter representation than the others.

4. **Encode** the coefficients

- Fourier based compression suffer from blocking effect at high compression rate.
- JPEQ2000 uses wavelet transforms
7.9.3. Wavelet-based Video compression schemes

In the ROI scheme, the non-LL coefficients are not calculated outside the region. For FP, significant LL-coefficients are quantised finely at the expense of others.
7.9.4. Wavelet-based Face Identification

The match score for class $i$ is:

$$S_i = \min (S_{i,1}, S_{i,2}, S_{i,3}, \ldots, S_{i,m}), \text{ for } i = 1 \ldots n$$
clear all
c = imread('ImageA.bmp');
imshow(c);
f = c;
[m n] = size(c);
k = n/2;
for i = 1:1:m
    for j = 1:1:k
        f(i, j) = c(i, 2*j) + c(i, 2*j-1);
        f(i, j+k) = c(i, 2*j) - c(i, 2*j-1);
    end
end
c = f;
k = m/2;
for j = 1:1:n
    for i = 1:1:k
        c(i, j) = uint8((f(2*i, j) + f(2*i-1, j))/2);
        c(i+k, j) = uint8((f(2*i, j) - f(2*i-1, j))/2);
    end
end
figure; imshow(c)
7.11 DWT Types in Matlab

<table>
<thead>
<tr>
<th>Wavelet</th>
<th>wfamily</th>
<th>wname</th>
</tr>
</thead>
<tbody>
<tr>
<td>Haar</td>
<td>'haar'</td>
<td>'haar'</td>
</tr>
<tr>
<td>Daubechies</td>
<td>'db'</td>
<td>'db2', 'db3', ..., 'db45'</td>
</tr>
<tr>
<td>Coiflets</td>
<td>'coif'</td>
<td>'coif1', 'coif2', ..., 'coif5'</td>
</tr>
<tr>
<td>Symlets</td>
<td>'sym'</td>
<td>'sym2', 'sym3', ..., 'sym45'</td>
</tr>
<tr>
<td>Discrete Meyer</td>
<td>'dmev'</td>
<td>'dmev'</td>
</tr>
<tr>
<td>Biorthogonal</td>
<td>'bior'</td>
<td>'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'</td>
</tr>
<tr>
<td>Reverse Biorthogonal</td>
<td>'rbio'</td>
<td>'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'</td>
</tr>
</tbody>
</table>
### 7.11 DWT Functions in Matlab

#### Analysis-Decomposition Functions

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dwt2</code></td>
<td>Single-level decomposition</td>
</tr>
<tr>
<td><code>wavedec2</code></td>
<td>Decomposition</td>
</tr>
<tr>
<td><code>wmaxlev</code></td>
<td>Maximum wavelet decomposition</td>
</tr>
</tbody>
</table>

#### Synthesis-Reconstruction Functions

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>idwt2</code></td>
<td>Single-level reconstruction</td>
</tr>
<tr>
<td><code>wavedec2</code></td>
<td>Full reconstruction</td>
</tr>
<tr>
<td><code>wcoefs2</code></td>
<td>Selective reconstruction</td>
</tr>
<tr>
<td><code>upcoef2</code></td>
<td>Single reconstruction</td>
</tr>
</tbody>
</table>

#### Decomposition Structure Utilities

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>detcoef2</code></td>
<td>Extraction of detail coefficients</td>
</tr>
<tr>
<td><code>appcoef2</code></td>
<td>Extraction of approximation coefficients</td>
</tr>
<tr>
<td><code>upwlev2</code></td>
<td>Recomposition of decomposition structure</td>
</tr>
</tbody>
</table>

#### De-Noising and Compression

<table>
<thead>
<tr>
<th>Function Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ddencmp</code></td>
<td>Provide default values for de-noising and compression</td>
</tr>
<tr>
<td><code>wbmpen</code></td>
<td>Penalized threshold for wavelet 1-D or 2-D de-noising</td>
</tr>
<tr>
<td><code>wdcbm2</code></td>
<td>Thresholds for wavelet 2-D using Birgé-Massart strategy</td>
</tr>
<tr>
<td><code>wdenlcmp</code></td>
<td>Wavelet de-noising and compression</td>
</tr>
<tr>
<td><code>wthresh</code></td>
<td>Threshold settings manager</td>
</tr>
</tbody>
</table>
END of Chapter 7