Wavelet application part 2: JPEG2000

Dept. Electronics Engineering, Pusan National Univ.
Video Signal Processing Lab.
Il Kyu Eom
Necessity of JPEG2000 (1)

- **Low bit-rate compression**: At low bit-rates (e.g. below 0.25 bpp for highly detailed gray-level images) the distortion in JPEG becomes unacceptable.

- **Lossless and lossy compression**: Need for standard, which provide lossless and lossy compression in one code-stream.

- **Large images**: JPEG doesn't compress images greater than 64x64K without tiling.
Necessity of JPEG2000 (2)

- **Single decompression architecture**: JPEG has 44 modes, many of them are application specific and not used by the majority of the JPEG decoders.
- **Transmission in noisy environments**: in JPEG quality suffers dramatically, when bit errors are encountered.
- **Computer generated imaginary**: JPEG is optimized for natural images and performs badly on computer generated images.
- **Compound documents**: JPEG fails to compress bi-level (text) imagery.
JPEG2000 - Targets

- Coding standard for:
  - different types of still images (gray-level, color, ...)
  - different characteristics (natural, scientific, ...)
  - different imaging models (client/server, real-time,...) within a unified and integrated system.

- This coding system is intended for:
  - low bit-rate applications, exhibiting rate-distortion
  - subjective image quality performance superior to existing standards.
JPEG2000 - encoder-decoder scheme
The source image is decomposed into components (up to 256).

The image components are (optionally) decomposed into rectangular tiles. The tile-component is the basic unit of the original or reconstructed image.

A wavelet transform is applied on each tile. The tile is decomposed into different resolution levels.

The decomposition levels are made up of subbands of coefficients that describe the frequency characteristics of local areas of the tile components, rather than across the entire image component.

The sub-bands of coefficients are quantized and collected into rectangular arrays of code blocks.
The bit planes of the coefficients in a code block (i.e. the bits of equal significance across the coefficients in a code block) are entropy coded.

The encoding can be done in such a way that certain regions of interest (ROI) can be coded at a higher quality than the background.

Markers are added to the bit stream to allow for error resilience.
Pre-processing – Image tiling

- Image may be quite large in comparison to the amount of memory available to the codec.
- Partition of the original image into rectangular non-overlapping blocks (tiles), to be compressed independently.
- Smaller tiles create more tiling artifacts.

The Effect of Tiling on Image Quality.

<table>
<thead>
<tr>
<th>Tiling</th>
<th>Bit Rate (b/p)</th>
<th>No Tiling</th>
<th>Tiles of Size $128 \times 128$</th>
<th>Tiles of Size $64 \times 64$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.125</td>
<td>24.75</td>
<td>23.42</td>
<td>20.07</td>
<td></td>
</tr>
<tr>
<td>0.25</td>
<td>26.49</td>
<td>25.69</td>
<td>23.95</td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>28.27</td>
<td>27.79</td>
<td>26.80</td>
<td></td>
</tr>
</tbody>
</table>

PSNR (in dB) for the color image “ski” (of size $720 \times 576$ pixels per component)

Figure 5: Example tiling of the 8-bit baboon image.
Pre-processing – DC level shifting

- The codec expects its input sample data to have a nominal dynamic range that is approximately centered about zero (0 -- 255 -> -128 -- 128)
- If the sample values are unsigned, the nominal dynamic range of the samples is adjusted by subtracting a bias from each of the sample values
Pre-processing – Component Transformations (1)

- Maps data from RGB to YCrCb
  - Y, Cr, Cb - less statistically dependent; compress better
  - serves to reduce the correlation between components, leading to improved coding efficiency.
- There are reversible and irreversible transforms
- Component transformations improve compression and allow visually relevant quantization

Figure G-1 — Placement of the DC level shifting with component transform
Pre-processing – Component Transformations (2)

Forward reversible component transform

\[
Y_0(x, y) = \left[ \frac{I_0(x, y) + 2I_1(x, y) + I_2(x, y)}{4} \right]
\]

\[
Y_1(x, y) = I_2(x, y) - I_1(x, y)
\]

\[
Y_2(x, y) = I_0(x, y) - I_1(x, y)
\]

Inverse reversible component transform

\[
I_1(x, y) = Y_0(x, y) - \left[ \frac{Y_2(x, y) + Y_1(x, y)}{4} \right]
\]

\[
I_0(x, y) = Y_2(x, y) + I_1(x, y)
\]

\[
I_2(x, y) = Y_1(x, y) + I_1(x, y)
\]
Irreversible component transformation (ICT):

- Floating point
- For use with irreversible (floating point 9/7) wavelet

\[
\begin{bmatrix}
Y \\
C_b \\
C_r
\end{bmatrix} = \begin{bmatrix}
0.299 & 0.587 & 0.114 \\
-0.16875 & -0.33126 & 0.5 \\
0.5 & -0.41869 & -0.08131
\end{bmatrix} \begin{bmatrix}
R \\
G \\
B
\end{bmatrix}
\]
Pre-processing – Component Transformations (4)

The JPEG 2000 multiple component encoder. Color transformation is optional. If employed, it can be irreversible or reversible.
Pre-processing – Component Transformations (5)

Figure 7: The ICT of the baboon image.
Pre-processing – Component Transformations (6)

Reversible component transformation (RCT):

- Integer approximation
- For use with reversible (integer 5/3) wavelet

\[
\begin{pmatrix}
Y_r \\
U_r \\
V_r
\end{pmatrix} = \begin{pmatrix}
\frac{R + 2G + B}{4} \\
R - G \\
B - G
\end{pmatrix}
\]
Wavelet transform (1)

- **Floating point 9/7 wavelet filter** for lossy compression
  - Best performance at low bit rate
  - High implementation complexity, especially for hardware

- **Integer 5/3 wavelet filter** for lossless coding
  - Integer arithmetic, low implementation complexity
Wavelet transform (2)
Wavelet transform (3)

- Two filtering modes:
  - **Convolution based**: performing a series of dot products between the two filter masks and the extended 1-D signal
  - **Lifting based**: sequence of very simple filtering operations for which alternately odd sample values of the signal are updated with a weighted sum of even sample values, and vise versa.
Wavelet transform (4)

Lifting
Wavelet transform (5)

**Lossless 1D DWT**

\[
\begin{align*}
Y(2n+1) &= X_{ext}(2n+1) - \left\lfloor \frac{X_{ext}(2n) + X_{ext}(2n+2)}{2} \right\rfloor \\
Y(2n) &= X_{ext}(2n) + \left\lfloor \frac{Y(2n-1) + Y(2n+1) + 2}{4} \right\rfloor
\end{align*}
\]

**Lossy 1D DWT**

\[
\begin{align*}
Y(2n+1) &\leftarrow X_{ext}(2n+1) + (\alpha \times [X_{ext}(2n) + X_{ext}(2n+2)]) \\
Y(2n) &\leftarrow X_{ext}(2n) + (\beta \times [Y(2n-1) + Y(2n+1)]) \\
Y(2n+1) &\leftarrow Y(2n+1) + (\gamma \times [Y(2n) + Y(2n+2)]) \\
Y(2n) &\leftarrow Y(2n) + (\delta \times [Y(2n-1) + Y(2n+1)]) \\
Y(2n+1) &\leftarrow -K \times Y(2n+1) \\
Y(2n) &\leftarrow (1/K) \times Y(2n)
\end{align*}
\]
Wavelet transform (6)

- **Symmetric extension:**
  - To ensure that for the filtering operations that take place at both boundaries of the signal, one signal sample exists and spatially corresponds to each coefficient of the filter mask.

![Figure F-16 — Periodic symmetric extension of signal](image)

JPEG2000
In JPEG2000 multiple stages of the DWT are performed. JPEG2000 supports from 0 to 32 stages. For natural images, usually between 4 to 8 stages are used.
The wavelet coefficients are quantized using a uniform quantizer with deadzone. For each subband $b$, a basic quantizer step size $\Delta b$ is used to quantize all the coefficients in that subband according to

$$q = \text{sign}(y) \left\lfloor \frac{|y|}{\Delta b} \right\rfloor$$
Quantization - Example

Given a quantizer step of 10 and an encoder input value of 21.82, the quantizer index is determined as shown:

\[
\text{Quantizer Index} = - \left\lfloor \frac{21.82}{10} \right\rfloor = -2
\]
Wavelet coefficients are associated with different subbands arising from the 2D separable transform applied.

These coefficients are then arranged into rectangular blocks within each sub-band, called code-blocks.

Figure 15: Example division of subbands into code-blocks.
Coefficient Bit Modeling (2)

- Code-blocks: coded a **bit-plane** at a time starting from the **Most Significant Bit-Plane** to the **Least Significant Bit-Plane**
Coefficient Bit Modeling (3)

- a special code-block scan pattern
Each coefficient bit in the bit-plane is coded in only one of the **Three Coding Passes:**
- Significance Propagation
- Magnitude Refinement
- Clean-up
Significance Propagation Pass

- If a bit is insignificant (=0) but at least one of its eight neighbors is significant (=1), then it is encoded.
- If the bit at the same time is a 1, its significance flag is set to 1 and the sign of the symbol is encoded.
Magnitude Refinement Pass:
• Samples which are significant and were not coded in the significance propagation pass.

Clean-up Pass:
● It codes all bits which were passed over by the previous two coding passes (insignificant bits). It is the first pass for MSB plane.
Quality layers organization

- The resulting bit streams for each code-block are organized into quality layers. A quality layer is a collection of some consecutive bit-plane coding passes from each tile.

![Quality layers organization diagram](image)

*Figure 18: Example quality layer distribution for stage 2 DWT with subbands containing only one code-block.*
Rate control is the process by which the code-stream is altered so that a target bit rate can be reached. The ideal truncation strategy is one that minimizes distortion while still reaching the target bit-rate. The code-blocks are compressed independently, so any bit stream truncation policy can be used.
The compressed data from the bit-plane coding passes are separated into packets.

Then, the packets are multiplexed together in an ordered manner to form one code-stream.
**Bit stream organization (2)**

**Precinct**: each sub-band is divided into rectangular blocks called precincts.

**Packets**: three spatially consistent rectangles comprise a packet.

**Code-block**: each precinct is further divided into non-overlapping rectangles called code-blocks.

Each code-block forms the input to the entropy encoder and is encoded independently.

Within a packet, code-blocks are visited in raster order.
Bit stream organization (3)
Decoding

- The decoder basically performs the opposite of the encoder:

Reverse-ICT

\[
\begin{bmatrix}
R \\
G \\
B
\end{bmatrix} =
\begin{bmatrix}
1.0 & 0.0 & 1.4021 \\
1.0 & -0.3441 & -0.7142 \\
1.0 & 1.7718 & 0.0
\end{bmatrix}
\begin{bmatrix}
Y \\
C_r \\
C_b
\end{bmatrix}
\]
JPEG2000- Characteristics

1. Compress once - decompress many ways
2. Region-Of-Interest (ROI) encoding
3. Progression
4. Error resilience
JPEG2000 - Markets & Applications

- Internet
- Mobile
- Printing
- Scanning
- Digital Photography
- Remote Sensing
- Facsimile
- Medical
- Digital Libraries
- E-Commerce
JPEG2000

Compress once, decompress many ways
A ROI is a part of an image that is encoded with higher quality than the rest of the image (the background). The encoding is done in such a way that the information associated with the ROI precedes the information associated with the background.

2 methods: Scaling based and Maxshift
Region-of-interest (ROI)

Scaling of ROI coefficients
Region-of-interest (ROI) - Scaling based

1. The wavelet transform is calculated
2. ROI mask is derived, indicating the set of coefficients that are required for up to lossless ROI reconstruction
3. The wavelet coefficients are quantized
4. The coefficients that lay out of the ROI are downscaled by a specified scaling value
5. The resulting coefficients are progressively entropy encoded (with the most significant bit planes first)
6. ROI's scaling value and coordinates are added to the bit stream.
Region-of-interest (ROI) - Maxshift method

- **ROI mask** (a bit map) is created describing which quantized transform coefficients must be encoded with better quality.
- The quantized transform coefficients outside the ROI mask (background coefficients) are scaled down so that the bits associated with the ROI are placed in highest bit-planes and coded before the background.
- the LSB of all shifted ROI coefficients is above the MSB (non zero) of all background's coefficients.
- **Advantage**: arbitrary shaped ROIs without the need for shape information at the decoder.
ROI - example

Original Image with ROI Defined

Decoded Image with ROI Intact
Scalability and bit-stream parsing

- 2 important modes of scalability:
  - Resolution/Spatial
  - Quality (SNR)

- Bit-stream parsing
  - A combination of spatial and quality scalability.
  - It is possible to progress by spatial scalability to a given (resolution) level and then change the progression by SNR at a higher level.
Scalability and bit-stream parsing

Multiresolution decomposition

- Different modes are realized depending on the way information is written into the codestream

JPEG2000
Resolution scalability (1)
Resolution scalability (2)
Resolution scalability (3)
Resolution scalability (4)
Quality scalability (1)
Quality scalability (3)
Error resilience

- **Error effects:**
  1. In a packet body: corrupted arithmetically coded data for some code-block => severe distortion.
  2. In a packet head: wrong body length can be decoded, code block data can be assigned to wrong code-blocks => total synchronization loss.
  3. Bytes missing (i.e. network packet loss): combined effects of error in packet head and body
Error resilience example


No transmission errors

No error resilience

Full error resilience
Reconstructed images compressed at 0.25 bpp by means of (a) JPEG and (b) JPEG2000
Example

Reconstructed images compressed at 0.125 bpp by means of (a) JPEG and (b) JPEG2000
Example

JPEG 2000 (1.83 KB)

Original (979 KB)

JPEG (6.21 KB)
EZW (Embedded Zerotree Wavelet)
EZW – Overview (1)

- **Two-fold problem**
  - Obtaining best image quality for a given bit rate
  - Accomplishing this task in an embedded fashion

- **What is Embedded Zerotree Wavelet (EZW)?**
  - An embedded coding algorithm
  - 2 properties, 4 features and 2 advantages

- **What is Embedded Coding?**
  - Representing a sequence of binary decisions that distinguish an image from the “null” image
  - Similar in spirit to binary finite-precision representations of real number
EZW – Overview (2)

• 2 Properties
  – Producing a fully embedded bit stream
  – Providing competitive compression performance

• 4 Features
  – Discrete wavelet transform
  – Zerotree coding of wavelet coefficients
  – Successive-approximation quantization (SAQ)
  – Adaptive arithmetic coding

• 2 Advantages
  – Precise rate control
  – No training of any kind required
Fig. 6. Flow chart for encoding a coefficient of the significance map.
## EZW Coding - Example

<table>
<thead>
<tr>
<th></th>
<th>26</th>
<th>6</th>
<th>13</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>-7</td>
<td>7</td>
<td>6</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>-4</td>
<td>4</td>
<td>-3</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>-2</td>
<td>-2</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

- **Initial threshold**

\[
T_0 = 2^{\left\lfloor \log_2 26 \right\rfloor} = 16
\]

- **8 bits from bit budget**
EZW Coding - Example

- 26 > 16 $\Rightarrow$ sp
- 6 < 16 $\Rightarrow$
  - descendants < 16 $\Rightarrow$ zr
- -7 < 16 $\Rightarrow$
  - descendants < 16 $\Rightarrow$ zr
- 7 < 16 $\Rightarrow$
  - descendants < 16 $\Rightarrow$ zr
- labels to be transmitted
  sp zr zr zr

<table>
<thead>
<tr>
<th></th>
<th>26</th>
<th>6</th>
<th>13</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>-7</td>
<td>7</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>-4</td>
<td>4</td>
<td>-3</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>-2</td>
<td>-2</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ T_0 = 2^{|\log_2 26|} = 16 \]
EZW Coding - Example

- $Ls = \{26\}$
- *The significant coefficient reconstructed value*
  
  $1.5T = 24$

- reconstructed bands

<table>
<thead>
<tr>
<th>24</th>
<th>0</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Reconstruction value
EZW Coding - Example

- $L_s = \{26\}$
- The significant coefficient
- $1.5T = 24$
- reconstructed bands

<table>
<thead>
<tr>
<th>24</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

- Difference $26 - 24$
- with a 2-level quantizer with reconstruction levels $\pm T/4$, correction term of 4
- Reconstruction
  
  $24 + 4 = 28$
- Transmitting the correction term costs a single bit.

<table>
<thead>
<tr>
<th>28</th>
<th>0</th>
<th>0</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
EZW Coding - Example

- Threshold = 8 (16/2)
- $6 < 8 \rightarrow$ descendants $> 8 \rightarrow$ iz
- $-7 < 8 \rightarrow$ descendants $< 8 \rightarrow$ zr
- $7 < 8 \rightarrow$ descendants $< 8 \rightarrow$ zr
- 13 no descendants $> 8 \rightarrow$ sp
- 10 no descendants $> 8 \rightarrow$ sp
- 6 no descendants $< 8 \rightarrow$ iz
- 4 no descendants $< 8 \rightarrow$ iz

<table>
<thead>
<tr>
<th></th>
<th>6</th>
<th>13</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>-7</td>
<td>7</td>
<td>6</td>
<td>4</td>
</tr>
<tr>
<td>4</td>
<td>-4</td>
<td>4</td>
<td>-3</td>
</tr>
<tr>
<td>2</td>
<td>-2</td>
<td>-2</td>
<td>0</td>
</tr>
</tbody>
</table>

- labels to be transmitted: iz zr zr sp sp iz iz
- Requires 14 bits
- Total bits = $9 + 14 = 23$
EZW Coding - Example

- The significant coefficient
  \[ 1.5T_1 = 1.5 \times 8 = 12 \]
- \( L_s = \{26, 13, 10\} \)
- reconstructed bands

with a 2-level quantizer with reconstruction levels \( \pm T_1 / 4 = \pm 2 \)

- \( 26 - 28 = -2 \) Correction term = -2
- \( 13 - 12 = 1 \) Correction term = 2
- \( 10 - 12 = -2 \) Correction term = -2

Each correction requires a single bit, the total bits \( 23 + 3 = 26 \).

<table>
<thead>
<tr>
<th>28</th>
<th>0</th>
<th>12</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

Reconstruction

<table>
<thead>
<tr>
<th>26</th>
<th>0</th>
<th>14</th>
<th>10</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Comment</td>
<td>Subband</td>
<td>Coefficient Value</td>
<td>Symbol</td>
</tr>
<tr>
<td>---------</td>
<td>---------</td>
<td>-------------------</td>
<td>--------</td>
</tr>
<tr>
<td>1</td>
<td>LL3</td>
<td>63</td>
<td>SP</td>
</tr>
<tr>
<td></td>
<td>HL3</td>
<td>-34</td>
<td>SN</td>
</tr>
<tr>
<td>2</td>
<td>LH3</td>
<td>-31</td>
<td>IZ</td>
</tr>
<tr>
<td>3</td>
<td>HH3</td>
<td>23</td>
<td>ZR</td>
</tr>
<tr>
<td></td>
<td>HL2</td>
<td>49</td>
<td>SP</td>
</tr>
<tr>
<td>4</td>
<td>HL2</td>
<td>10</td>
<td>ZR</td>
</tr>
<tr>
<td></td>
<td>HL2</td>
<td>-13</td>
<td>ZR</td>
</tr>
<tr>
<td></td>
<td>LH2</td>
<td>14</td>
<td>ZR.</td>
</tr>
<tr>
<td>5</td>
<td>LH2</td>
<td>15</td>
<td>ZR.</td>
</tr>
<tr>
<td></td>
<td>LH2</td>
<td>-9</td>
<td>ZR.</td>
</tr>
<tr>
<td>6</td>
<td>HL1</td>
<td>7</td>
<td>Z</td>
</tr>
<tr>
<td></td>
<td>HL1</td>
<td>13</td>
<td>Z</td>
</tr>
<tr>
<td>7</td>
<td>LH1</td>
<td>4</td>
<td>Z</td>
</tr>
<tr>
<td></td>
<td>LH1</td>
<td>-1</td>
<td>Z</td>
</tr>
<tr>
<td></td>
<td>LH1</td>
<td>-3</td>
<td>Z</td>
</tr>
<tr>
<td></td>
<td>LH1</td>
<td>-2</td>
<td>Z</td>
</tr>
<tr>
<td></td>
<td>LH1</td>
<td>47</td>
<td>SP</td>
</tr>
</tbody>
</table>