JPEG 2000

• Baseline JPEG Pros and Cons (as compared to JPEG2000)

  ➢ Advantages
    ✓ Memory efficient
    ✓ Low complexity
    ✓ Compression efficiency
    ✓ Visual model utilization

  ➢ Disadvantages
    ✓ Single resolution
    ✓ Single quality
    ✓ No target bit rate
    ✓ No lossless capability
    ✓ No tiling
    ✓ No region of interest
    ✓ Blocking artifacts
    ✓ Poor error resilience
**JPEG 2000**

- **Major Advantages of JPEG 2000 over JPEG**
  - Lossy to lossless in one system: encoder decides maximum image quality (up to and including lossless).
  - Encoder decides maximum resolution (or size).
  - New functionalities including
    - Region-of-Interest coding (ROI)
    - Error resilience
    - Progression orders supported in four dimensions
      - SNR (quality, layer)
      - Resolution
      - Spatial location
      - Component
  - Any image quality or size can be decompressed from the resulting codestream.
  - Compression noise “build-up” that occurs in most compression schemes when repetitive compress/decompress cycles are utilized can be avoided.
  - Better compression at low bit rates than JPEG.
  - Better at compressing compound images and graphics.
JPEG 2000

- JPEG 2000 Features
  - Compress once; decompress many ways.
  - High compression efficiency.
  - Lossless color transformation.
  - Lossy and lossless coding in one algorithm.
  - Embedded lossy to lossless coding.
  - Progressive by resolution and quality.
  - Static and dynamic region-of-interest.
  - Error resilience (independent block coding; resync markers; possible error detection and concealment mechanisms).
  - Visual (fixed and progressive) coding.
  - Multiple component images.
  - Continuous-tone and bi-level image compression.
  - Paletized (mixed-mode) images (Part 2).
JPEG 2000

• Brief Historical Perspective
  
  ➢ Published as a joint ISO/IEC standard, as well as am ITU-T recommendation
    ✓ ISO stands for International Organization for Standardization.
    ✓ IEC stands for International Electrotechnical Commission.
    ✓ ITU-T stands for International Telecommunications Union-Terminal Sector; known formerly as the Consultative Committee for the International Telephone and Telegraph (CCITT).

  ➢ Motivated by Ricoh’s submission of the CREW algorithm to the earlier JPEG-LS standardization effort for lossless and near-lossless compression => LOCO-I of HP selected as the basis for JPEG-LS.

  ➢ CREW provided a rich set of features that prompted a new JPEG 2000 standardization effort; a proposal was prepared mainly by Martin Boliek of Ricoh and a call for technical contributions was issued in March 1997 by ISO/IEC JTC 1/SC 29/WG1 (Working Group 1 of Study Committee 29 of Joint Technical Committee 1).
JPEG 2000

- Published in several parts:
  - Part 1: Core Coding System; Baseline JPEG2000
    ✓ Describes minimal decoder and code-stream syntax required.
    ✓ Suitable for maximal interchange with a limited number of options.
  - Part 2: Extended JPEG2000
    ✓ Provides optional extensions to enhance compression performance and/or enable compression of unusual data types.
    ✓ Suitable for image compression applications where interchange is less important than other requirements.
  - Part 3: Motion JPEG2000
    ✓ Provides extensions for image sequences (video).
  - Part 4: includes information on compliance/conformance.
  - Part 5: Reference Software
    ✓ JasPer: C implementation of Part 1 by UBC and Image Power.
    ✓ JJ2000: Java implementation of Part 1 by EPFL and Canon Research Centre France (CRF).
    ✓ Other implementations (not included in Part 5):
      - Kakadu: C implementation of Part 1 by David Taubman (Univ. New South Wales).
      - Verification Model (VM): C implementations provided by the ISO/IEC.
  - Part 6: includes an additional file format tailored for compound documents.
  - Part 7: abandoned!
  - Parts 8 to 12 dealing with security, interactive protocols, volumetric imaging, wireless applications, ISO Base Media File Format (common with MPEG 4, Part 12), respectively.
JPEG 2000

• JPEG 2000 Algorithm

1. Tiling: Dividing the image into non-overlapping rectangular tiles
2. DC Level shifting: offset so that B-bit image samples are in \([-2^{B-1}, 2^{B-1})\).
3. Wavelet Transform
4. Quantization: Uniform scalar quantization with dead-zone
5. Divide into coding blocks: Smallest geometric structures for coding
6. Tier-1 Coding: Bit-Plane coding in three passes:
   1. Significance propagation
   2. Magnitude refinement
   3. Clean-up pass
7. Tier-2 Coding: Generate final bit-stream
8. Rate Control: Minimize MSE for desired bit-rate
JPEG 2000

- **Tiling**
  - Image is usually divided into non-overlapping rectangular tiles.
  - Tiles provide a simple vehicle for limiting implementation memory and for spatial random access.

- **Wavelet Transform**
  - For high-performance lossy compression, (9,7) wavelet transform is used.
  - For lossless compression, (5,3) wavelet transform is implemented.

- **Quantization:**
  - Uniform scalar quantizer with dead-zone twice the step size q.
JPEG 2000

- **Note:** Embedded bit-plane coding corresponds to embedded uniform scalar quantizers with dead-zone twice the step size.
JPEG 2000

- Coding blocks
  - Smallest geometric structure.
  - Subbands are partitioned into coding blocks.
  - The end user can choose the coding block size (default 64×64).
  - Bit-plane coding is performed on coding blocks.
  - Coding blocks help enable low-memory implementation.
  - They provide fine-grain random access to spatial regions.
JPEG 2000

• Tier-1 Coding
  - Consists of context-dependent, binary, arithmetic coding of bit-planes of coding blocks.
  - Bit-planes are coded from MSB to LSB
  - Scan pattern for coding is column-wise, in columns with length 4 (called stripe).
JPEG 2000

- Coding is done in three passes (except for MSB which is coded using only one clean-up pass):
  - Significance propagation
  - Magnitude refinement
  - Clean-up

- Definition: Significant Coefficients
  A wavelet coefficient is considered significant with respect to bit-plane $p$ if its magnitude is $\geq 2^p$.

*Note:* Once a coefficient becomes significant in a bit-plane $p$, it stays significant in all the remaining bit-planes $p-1, p-2, \ldots, 1$. 
For a given bit-plane $p$ (other than the MSB):

- The **significance propagation (SP)** pass is first used to code the significance (bits in the current bit-plane) of the insignificant neighbors of an already significant wavelet coefficient and determine which of these insignificant neighboring wavelet coefficients have become significant and, if they became significant, code their signs.
- The **magnitude refinement (MR)** coding pass is then used to code the bits corresponding to the already significant wavelet coefficients.
- Finally, the **clean-up (CU)** pass is used to determine which of the remaining, previously insignificant wavelet coefficients have just become significant in the current considered bit plane, and to code the bits of all wavelet coefficients that have not been coded as part of the SP and MR passes for the current bit-plane $p$.

For the MSB bit-plane, only the clean-up coding pass is applied.

Context-based arithmetic coding (same as MQ arithmetic coder used in JBIG2) is used for coding in all the passes.
JPG E 2000

For MSB, coding is done using the clean-up pass as follows:

- Scan each column in considered stripe and check if all 4 coefficients are insignificant in current bit-plane and none has a significant neighbor:
  - If yes => enter significance coding/run mode
  - If no => enter significance coding/normal mode

- Significance coding/Normal Mode: code significance (current bit) of any coefficient that is insignificant and has insignificant neighbors using significance coding primitive. If coefficient became significant in current bit-plane, the sign of that sample is coded (using sign coding primitive) immediately after its significance coding.

- Significance coding/Run Mode: For considered column, check if any coefficient became significant in current bit-plane:
  - If no => send 0
  - If yes => send 1 followed by arithmetically coded position of first significant coefficient in column followed by its arithmetically coded sign (using sign coding primitive); in addition, all remaining samples are coded using significance coding/normal mode until run mode encountered again.
For all other bit-planes:

- For the SP pass, for each stripe (4 rows), scan bits in columns:
  - If coefficient is insignificant in previous (more significant) bit-plane but has any of its 8 neighbors as significant, arithmetically code using significance coding primitive (significance coding/normal mode); otherwise, skip and do not code.

- For the MR pass, all the already significant coefficient (those which became significant in previous bit-planes) are arithmetically coded using the magnitude refinement primitive; scanning is also done columnwise within each stripe as before.

- The clean-up pass is finally used to code all bits that were not coded in the SP and MR passes; coding is done same as for the MSB.
The arithmetic coding employs different coding context models depending on the coding pass and subband type.

For significance coding, nine different contexts are used, depending on the significance states of the eight-connected neighbors of the current bit.

- The eight neighbors are classified into three groups: horizontal neighbors (H), vertical neighbors (V) and diagonal neighbors (D).
- X denotes the current sample.
- In table on left, “x” denotes a don’t care entry.

<table>
<thead>
<tr>
<th>D₀</th>
<th>V₀</th>
<th>D₁</th>
</tr>
</thead>
<tbody>
<tr>
<td>H₁</td>
<td>X</td>
<td>H₀</td>
</tr>
<tr>
<td>D₂</td>
<td>V₁</td>
<td>D₃</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>LL and LH subbands (vertical high pass)</th>
<th>HL subband (horizontal high pass)</th>
<th>HH subband (diagonal high pass)</th>
<th>Context label</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\sum H)</td>
<td>(\sum V)</td>
<td>(\sum D)</td>
<td>(\sum H)</td>
</tr>
<tr>
<td>2</td>
<td>x</td>
<td>x</td>
<td>x</td>
</tr>
<tr>
<td>1</td>
<td>(\geq 1)</td>
<td>x</td>
<td>(\geq 1)</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>(\geq 1)</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>x</td>
<td>2</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>x</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>(\geq 2)</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
JPEG 2000

- For sign coding, five different contexts are used, based on the four-connected neighbors significance status and sign. Each neighbor can take three values: significant positive (+1), significant negative (-1) and insignificant (0).

<table>
<thead>
<tr>
<th>$H_0 + H_1$</th>
<th>$V_0 + V_1$</th>
<th>Context label</th>
<th>Sign flipping</th>
</tr>
</thead>
<tbody>
<tr>
<td>&gt; 0</td>
<td>&gt; 0</td>
<td>4</td>
<td>No</td>
</tr>
<tr>
<td>&gt; 0</td>
<td>= 0</td>
<td>3</td>
<td>No</td>
</tr>
<tr>
<td>&gt; 0</td>
<td>&lt; 0</td>
<td>2</td>
<td>No</td>
</tr>
<tr>
<td>= 0</td>
<td>&gt; 0</td>
<td>1</td>
<td>No</td>
</tr>
<tr>
<td>= 0</td>
<td>= 0</td>
<td>0</td>
<td>No</td>
</tr>
<tr>
<td>= 0</td>
<td>&lt; 0</td>
<td>1</td>
<td>Yes</td>
</tr>
<tr>
<td>&lt; 0</td>
<td>&gt; 0</td>
<td>2</td>
<td>Yes</td>
</tr>
<tr>
<td>&lt; 0</td>
<td>= 0</td>
<td>3</td>
<td>Yes</td>
</tr>
<tr>
<td>&lt; 0</td>
<td>&lt; 0</td>
<td>4</td>
<td>Yes</td>
</tr>
</tbody>
</table>
JPEG 2000

- The contexts that are used to code the magnitude refinement (MR) information, are determined by the summation of the significance status of the 8 neighbors and whether the bits being coded are the first refinement bits. There are three coding contexts:

<table>
<thead>
<tr>
<th>$\sum H + \sum V + \sum D$</th>
<th>First refinement for this coefficient</th>
<th>Context label</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>false</td>
<td>2</td>
</tr>
<tr>
<td>$\geq 0$</td>
<td>true</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>true</td>
<td>0</td>
</tr>
</tbody>
</table>

“$x$” denotes a don’t care entry
JPEG 2000

- Rate Allocation:
  The rate increase and the distortion reduction associated with each coding pass is recorded. This information is then used by the Tier-2 coding stage to determine each coding block’s contribution to the final bit-stream in order to optimize the R-D performance.
Tier-2 Coding

- Given the compressed bit-stream for each coding block, tier-2 coding is performed to determine each coding block’s contribution to the final bit-stream in order to optimize the R-D performance.
- In the existing rate-distortion optimization process, the distortion metric is chosen to be the conventional MSE. In this case, the rate allocation attempts to minimize the peak signal to noise ratio (PSNR).
- One can also choose other perceptual distortion metrics that take various HVS properties into consideration.
- Given the rate distortion information collected in the tier-1 coding stage for each coding pass of each coding block, the rate control attempts to figure out the optimal coding pass inclusion strategy.
- First, a convex hull search is performed to find out the candidate \((R,D)\) truncation points for each coding block.
JPEG 2000

- An example of a Rate-Distortion (R-D) curve for one coding block:

- Each coding pass determines a point (shown as a dot) in the R-D plane. After the convex hull search, only those points shown as black dots, are kept as candidate truncation points. The R-D slope values associated with each black dot are stored.
Given the R-D curve for each coding block, a bisection search is performed among all the coding blocks to find the R-D slope values that meet the desired bit rate.

Those coding passes with steeper R-D slopes are included in the final bit-stream.

Those coding passes that are not included in the final bit-stream are ignored.

The basic unit in the final bit-stream is a packet.

The formed packets can be of variable sizes.

A packet consists of a packet header and a packet body.

The packet header contains the following information: which coding blocks are included in this packet, number of bit planes for each newly included coding block (for which no parts have yet been transmitted in previous packets), number of coding passes included in this packet from each included coding block, number of bytes contributed by each included coding block.
In order to allow progressive bit-rate control, different bit-stream layers can be formed, where each layer corresponds to a bit-rate in an incremental fashion.

For a given layer, all the blocks in the same subband are usually included in one single packet, except when the “precinct” mode is used.

In the precinct mode, blocks belonging to the same subband can be classified into different disjoint groups in order to allow resolution scalability for each group separately.

- Only coding blocks (including several adjacent coding blocks) corresponding to the same location and same DWT level can be put in the same precinct.
JPEG 2000

Illustration: Precinct

Precinct

Coding block
JPEG 2000

• Bit-Stream Structure

- To encode the information in the packet header efficiently, a tag tree representation is used.
- The packet body contains the actual coded bits from each included coding block.
JPEG 2000

• Color Image or Multi-Component Image Coding
  ➢ Image can consist of a collection of two-dimensional rectangular arrays of samples, called image components. Example: an RGB color image consists of 3 components.
  ➢ Image components need not have the same number of samples.
  ➢ There are two types of transforms that can be applied to first three components of the image (assumption: RGB image):
    ✓ Irreversible color transform (ICT): used in conjunction with the irreversible wavelet transform; it employs floating-point arithmetics.
    ✓ Reversible color transform (RCT): used in conjunction with the reversible wavelet transform; it maps integer color components to integer transformed image components.
  ➢ The ICT and RCT are color transforms that convert the RGB data into an opponent color representation with a luminance channel and two color difference channels \((Y C_b C_r \text{ and } Y D_b D_r, \text{ respectively})\).
JPEG and JPEG2000 Coding - Comparison

JPEG - 10,696 Bytes  
JPEG2000 - 10,436 Bytes

http://www.elecard.com/products/j2kwavelet.php
JPEG2000 Codec

Downloadable JPEG2000 codec and viewer at
http://www.kakadusoftware.com/
https://www.ece.uvic.ca/~Frodo/jasper/
JPEG 2000

• References