JPEG2000 Notes

Based on the papers:

1. D. Taubman and M. Marcellin, “JPEG2000: Standard for Interactive Imaging,” *Proc. of the IEEE*, Vol. 90, No. 8, pp. 1336 – 1357, August 2002.

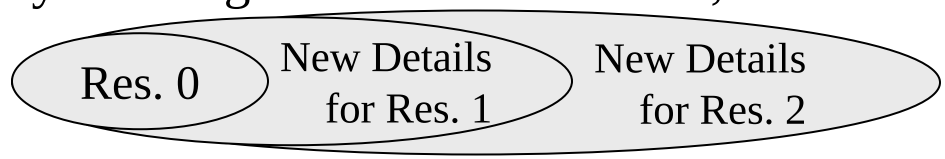
2. C. Christopoulos, A. Skodras, and T. Ebrahimi, “The JPEG2000 Still Image Coding System: An Overview,” *IEEE Trans. Cons. Elect.*, Vol. 46., No. 4, pp. 1103 – 1127, Nov. 2000.

JPEG2000 Overview [1]

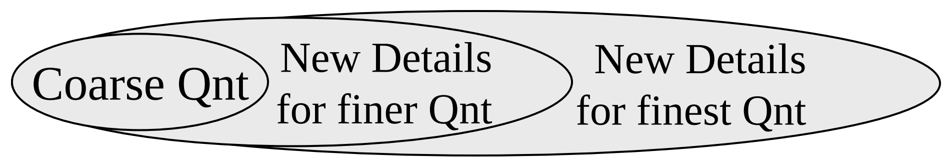
- A New Standard:
  - efficient, flexible, interactive
- Big emphasis on providing a scalable data stream:
  - an embedded set of smaller streams, where each sub-stream gives an efficient R-D compression of the image
- Other JPEG2000 Advantages include:
  - Better R-D performance than past standards
  - Progressive lossy-to-lossless stream
  - Can resequence compressed stream
  - Can crop in compressed domain
  - Can define regions of interest (ROI) to have better quality
  - Can work with "truly enormous" images

Scalability [1]

- JPEG2000 provides four types of scalability
  - **Resolution Scalability**
    - As the stream progresses you first get low resolution data, then medium resolution, etc.



  - **Distortion (or SNR) Scalability**
    - As the stream progresses you first get low SNR version, then medium SNR, etc.

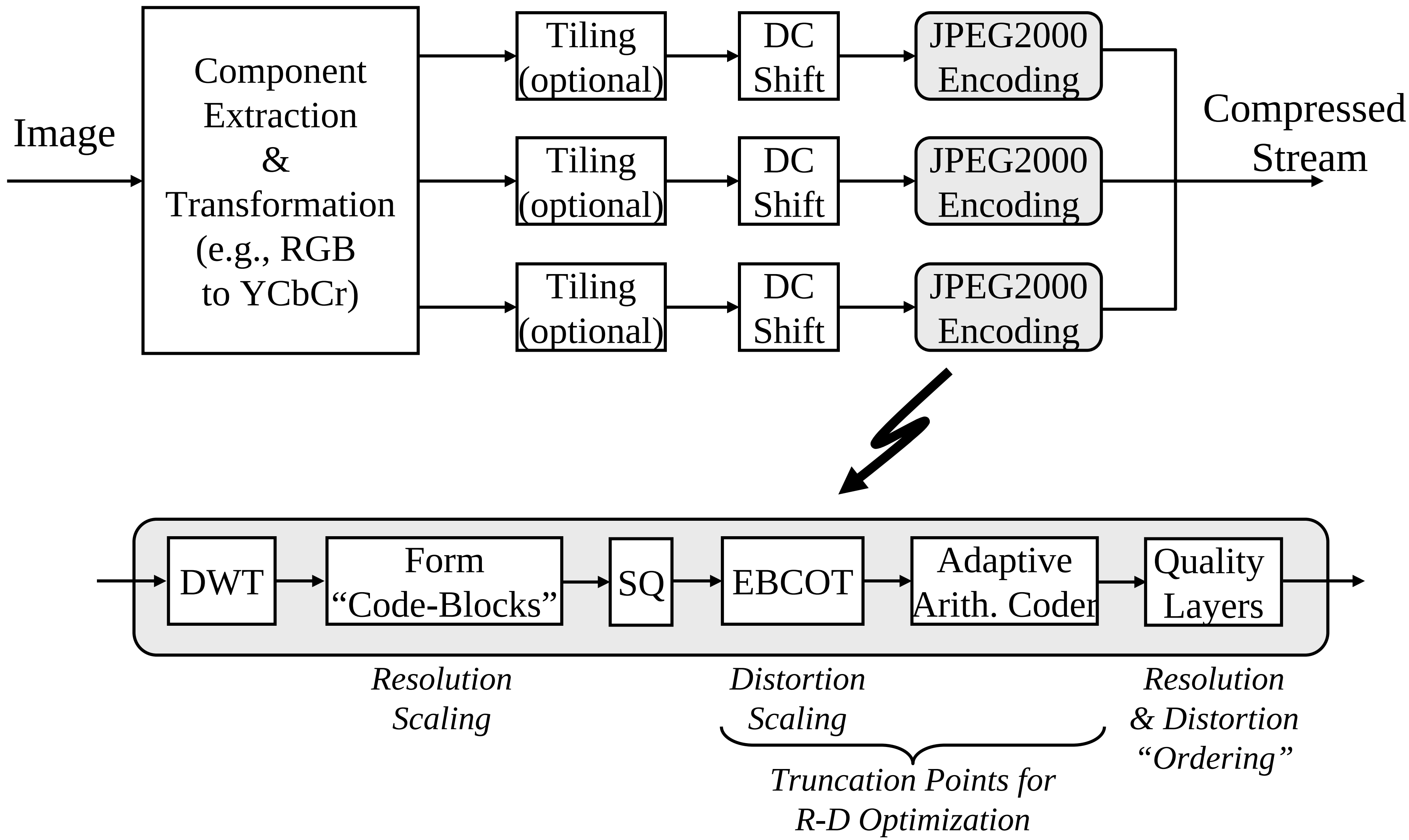


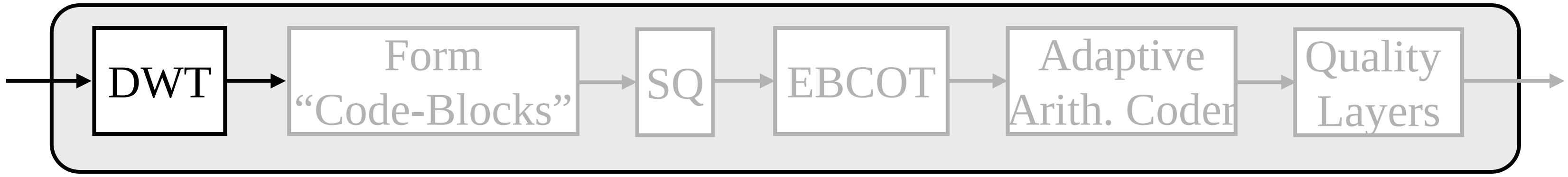
  - **Spatial Scalability**
    - As the stream progresses you first get a specific spatial region, then an expanded spatial region, etc.
  - **Component Scalability**
    - Example: As the stream progresses you first get gray-scale image, then color, etc.
- Can have combinations of these types

Advantages of Scalability [1]

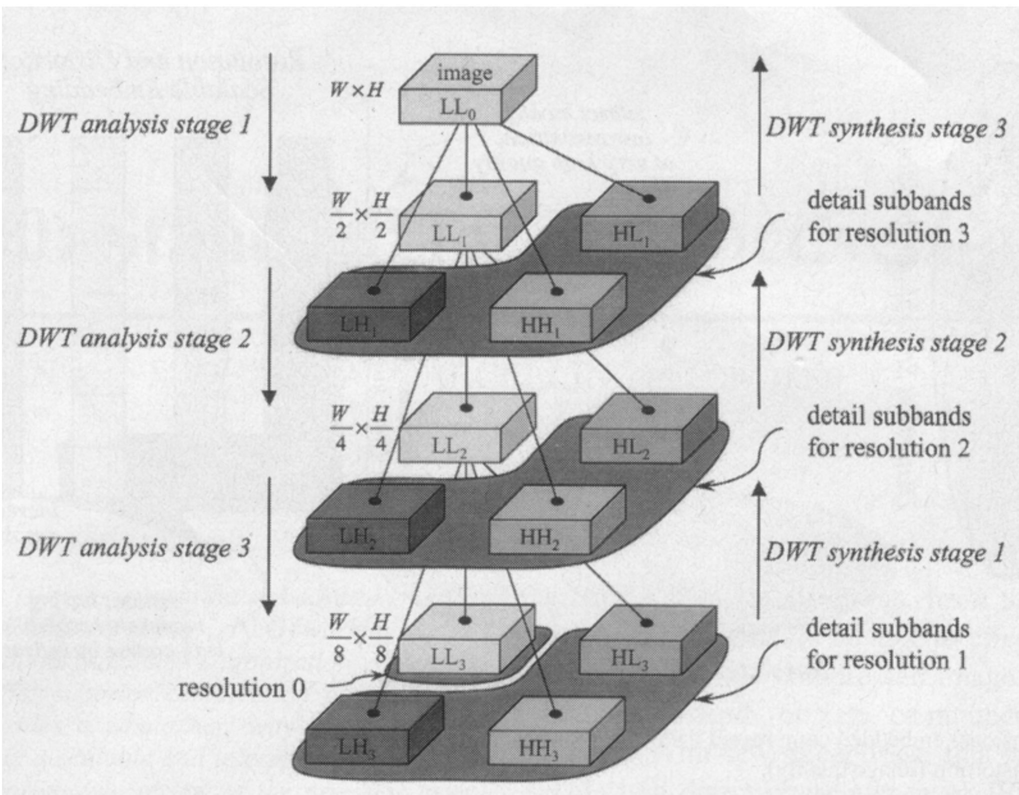
- "Compress Once… Decompress in Many Ways"
  - At compression… no need to know consumer's needs
- Can distribute to multiple clients with different:
  - display resolutions, regions of interest, communication rates
- Can send to a single client interactively:
  - Dynamically change ROI, resolution, SNR…. as client's interest or link speed changes
- Elegant solution to transmission-time limited cases:
  - Truncate stream when deadline occurs
  - You get the best quality image possible within that deadline

Basic JPEG2000 Architecture [1,2]

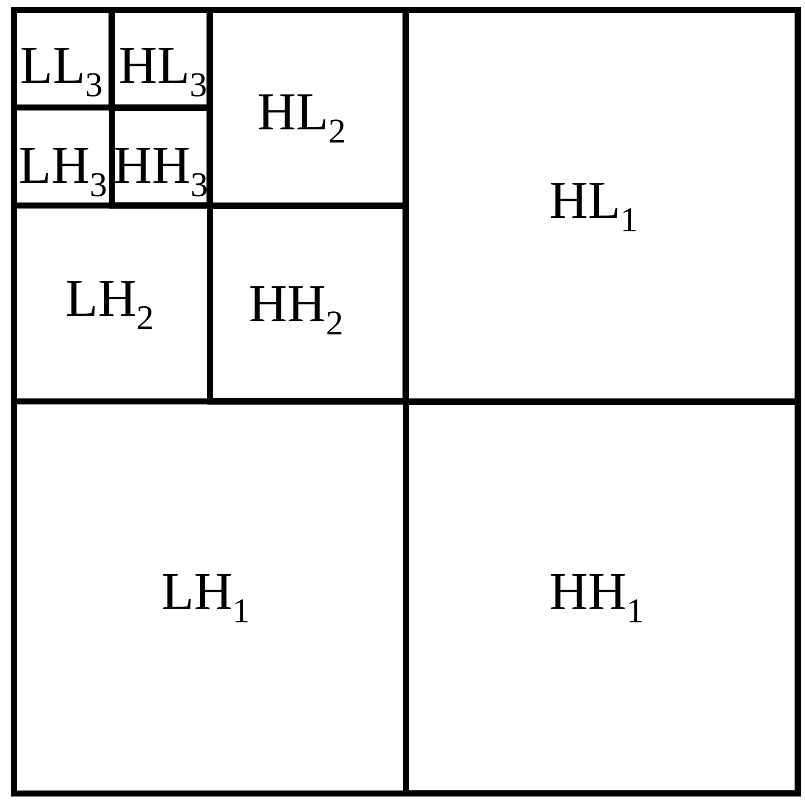




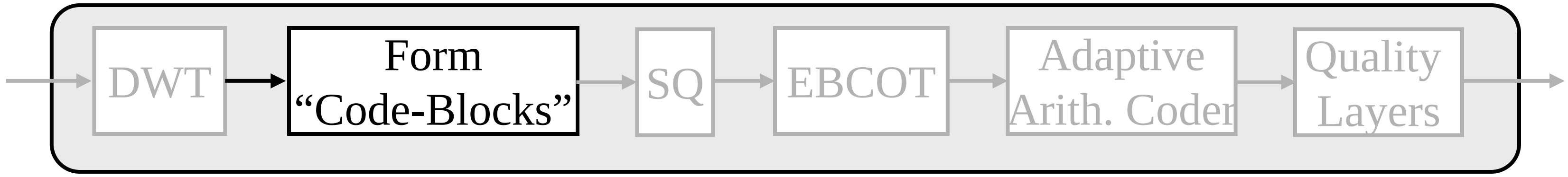
- Discrete Wavelet Transform (DWT)
  - Provides Multiresolution Decomposition
  - Enables Resolution Scaling
  - Each lower resolution level: dimensions of subband images are halved



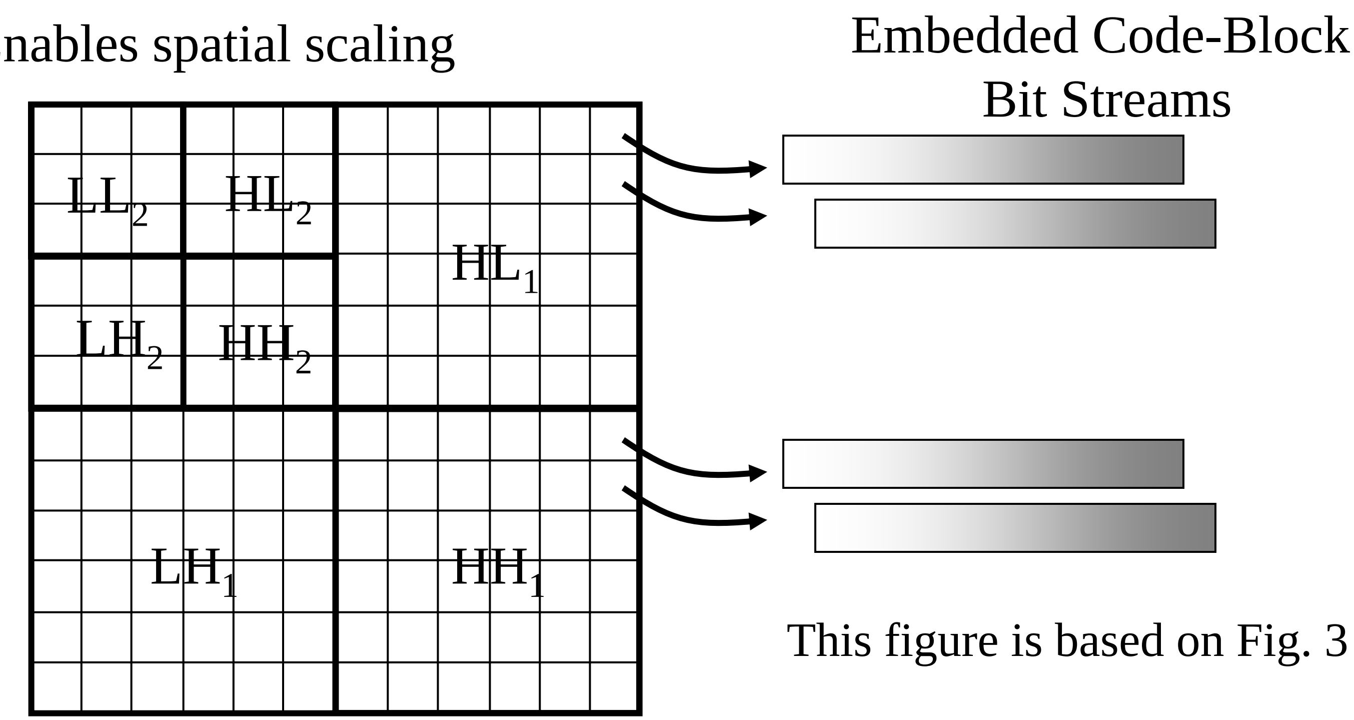
This figure is Fig. 2 in [1]



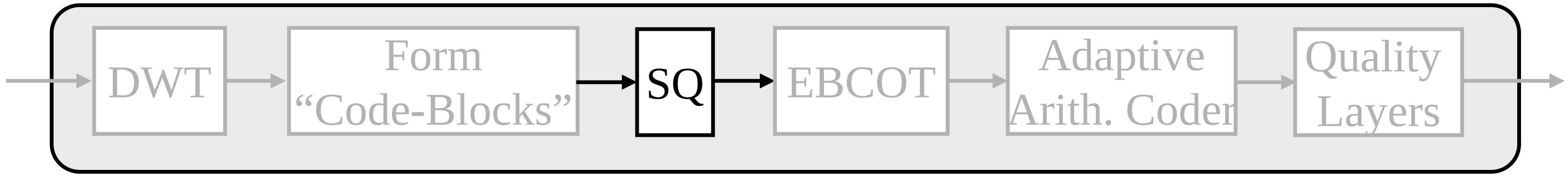
- DWT for JPEG2000
  - Two different filter pairs used: 5/3 and 9/7
    - Symmetric PR filters of odd length must differ in length by two
    - 5/3 is used to ensure integer sample values get transformed to integer wavelet values ➔ allows lossless wavelet-based coding
  - DWT is implemented using so-called “lifting” method
    - Lifting is essential to enabling lossless wavelet-based coding
    - It also enables computational and memory savings



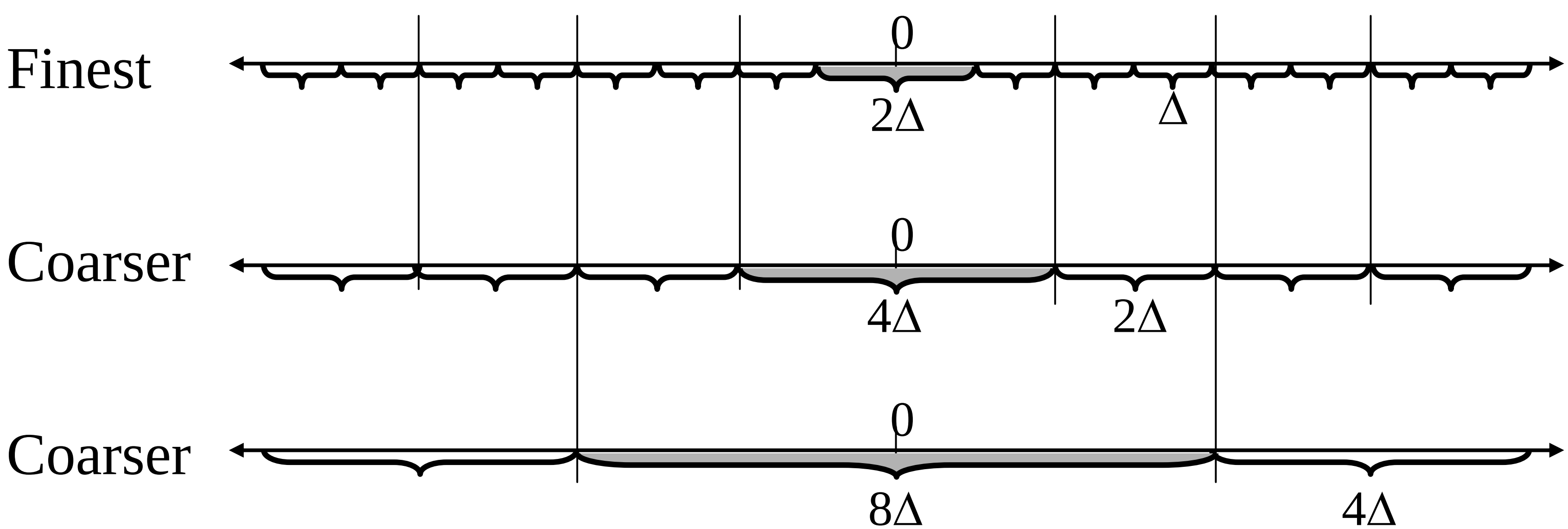
- Form “Code-Blocks”
  - Each subband is partitioned into code-blocks:
    - typical block sizes are 32×32 or 64×64
  - Each code-block is coded separately
  - Code-blocks constitute one aspect of Embedding
    - Enables resolution scaling
    - Enables spatial scaling



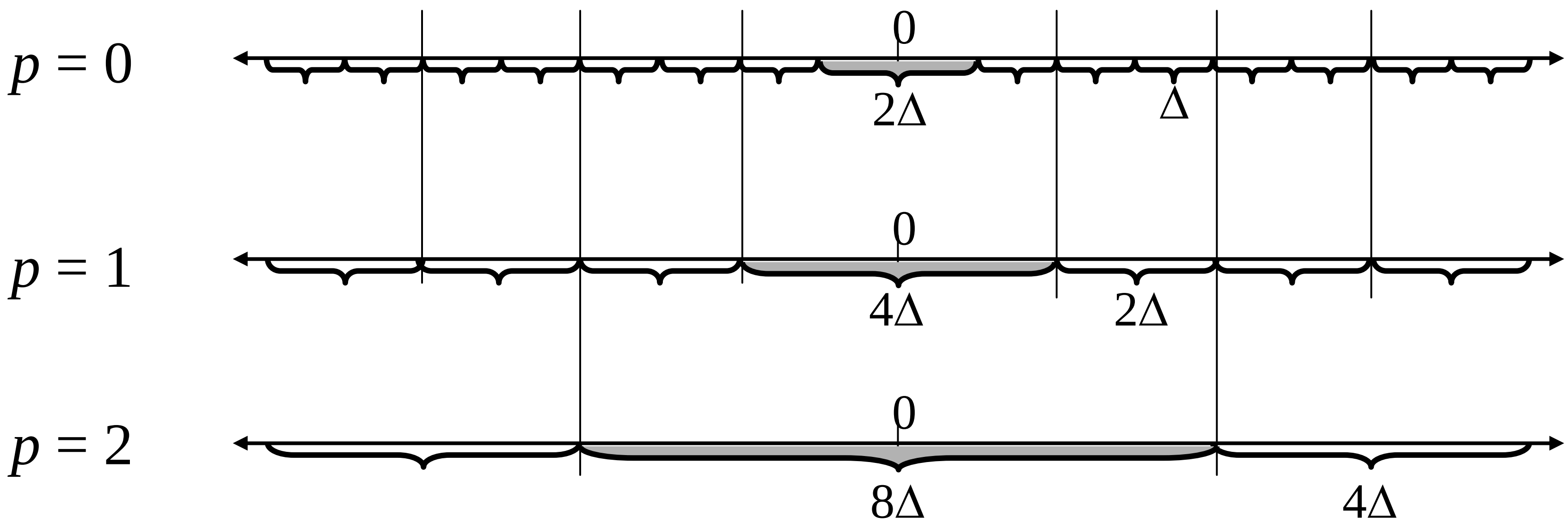
This figure is based on Fig. 3 in [1]



- Scalar Quantizer viewed as Embedded Quantizers w/ Dead-Zone
  - Embedded *Coders* necessarily call for Embedded *Quantizers* :
  - Embedded Quantizers lead to idea of "Bit Planes"
  - Dead-Zone (or "fat zero") useful for image coding
    - The commonly small high-frequency coefficients get quantized to zero and lead to lots of insignificant coefficients
    - This structure gets exploited in the Arithmetic Coder



This figure is based on Fig. 13 in [1]



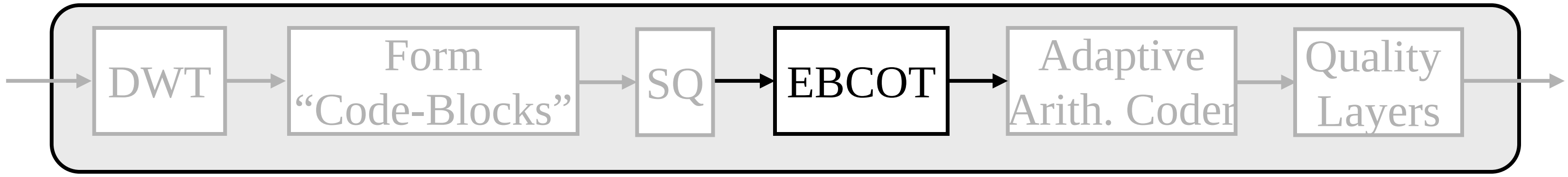
$$q_b^{(p)}[j] = \operatorname{sign}\{y_b[j]\} \cdot \left\lfloor \frac{|y_b[j]|}{2^p \Delta} \right\rfloor$$

$$= \operatorname{sign}\{q_b^{(0)}[j]\} \cdot \left\lfloor \frac{|q_b^{(0)}[j]|}{2^p} \right\rfloor$$

Shift Right & Truncate

Coarser quantization is obtained by keeping the most significant bits and sign of the sample

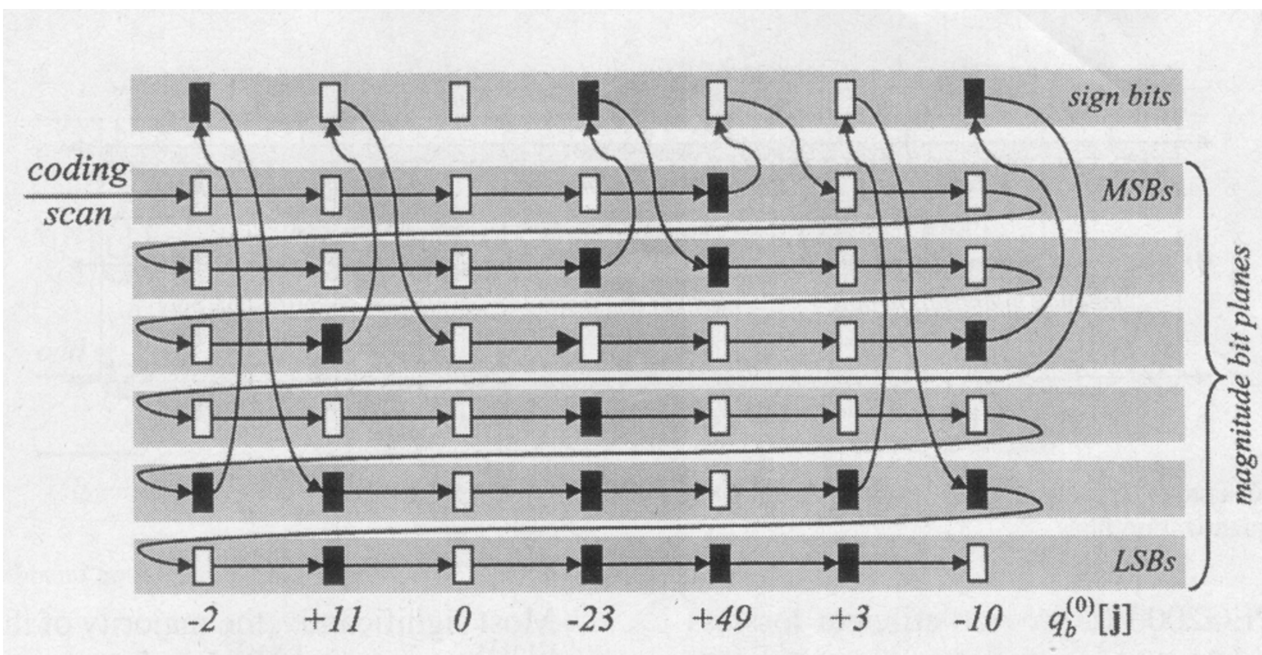
Bit-Plane Viewpoint

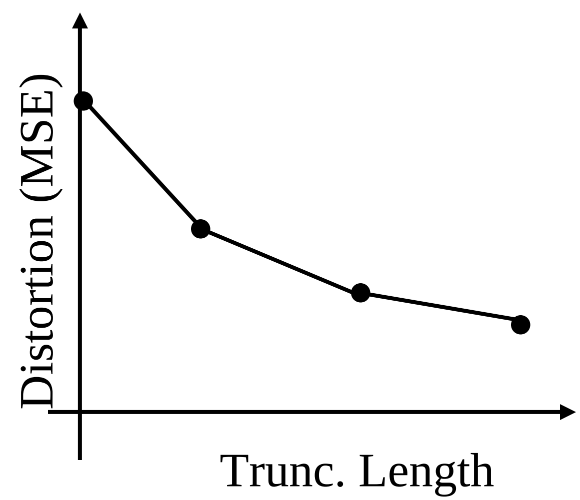


EBCOT Bit-Plane Coding Procedure

EBCOT = Embedded Block Coding with Optimized Truncation

Each code-block is optimally embedded coded

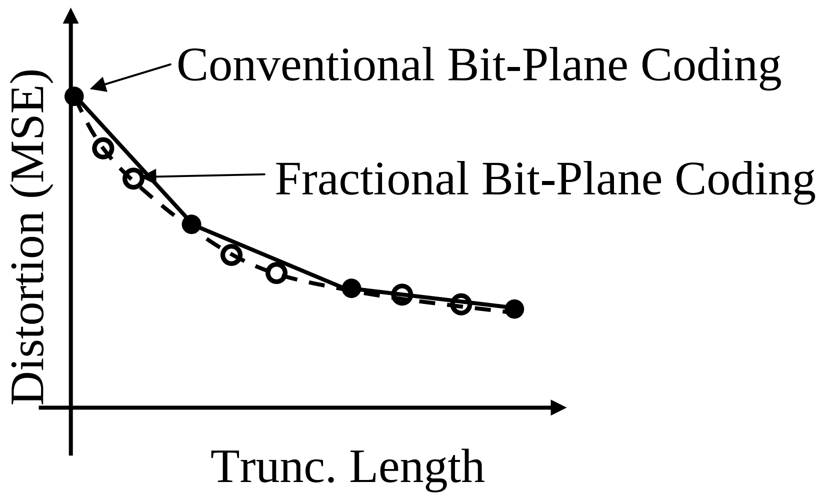




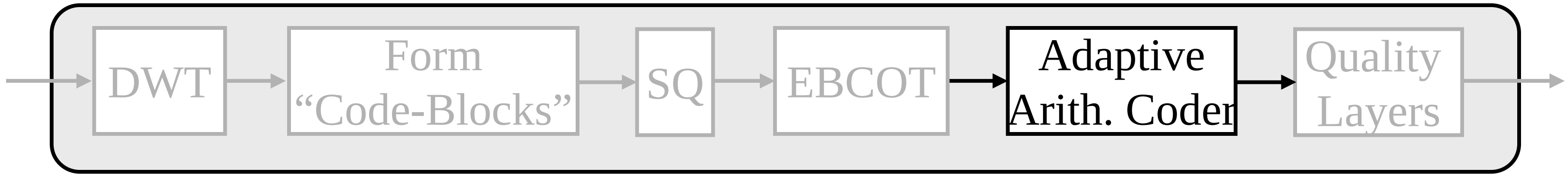
This figure is Fig. 14 in [1]

EBCOT Fractional Bit-Plane Coding

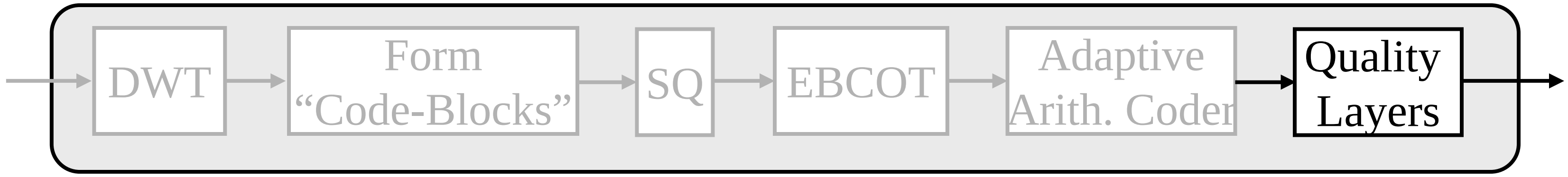
- The above coding allows truncation only at bit-plane end-points
  - Not enough points to give “fine grain” operational R-D function
- So… Use Sequence of Three Fractional Bit-Plane Passes
  - First Pass: Code Magnitude & Sign Bit only for samples likely to give largest distortion improvement
    - Samples for which
      - All coded magn. bits have been so far been zero
      - At least one neighbor is non-zero
  - Last Pass: Code those expected to be least effective in reducing distortion



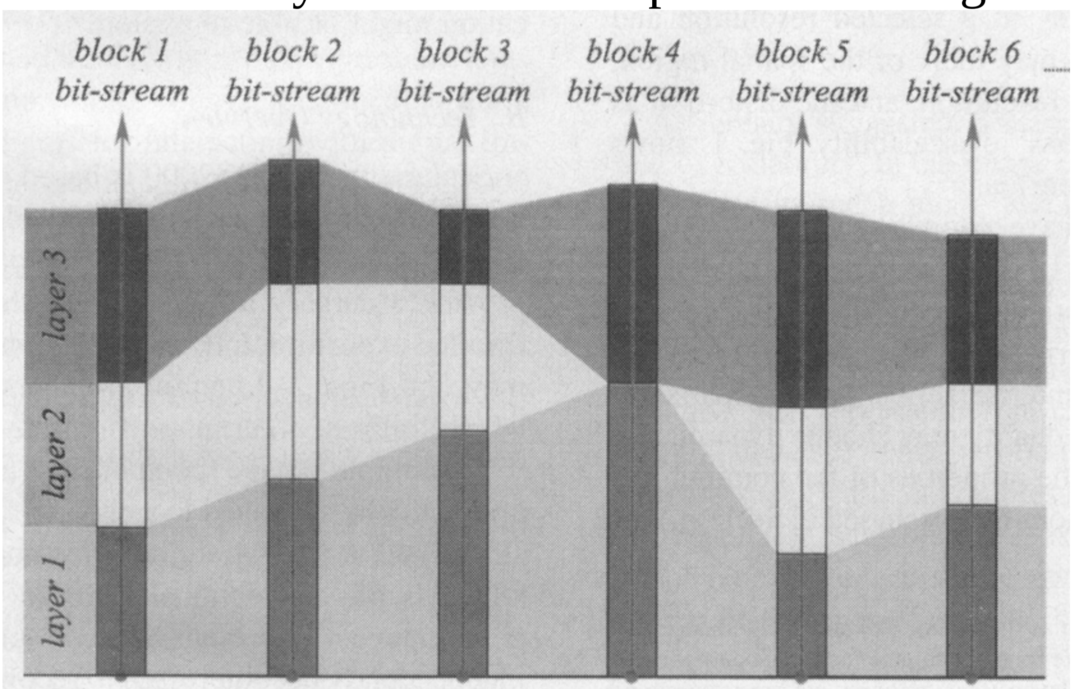
This figure is based on Fig. 15 in [1]



- Adaptive Arithmetic Coder
  - Used to remove substantial redundancy between successive bit planes
  - Uses the so-called MQ coder implementation
  - Coder switches between 18 different adaptive probability models
    - selects model based on previously coded bits from the current & previous bit planes
    - each model adaptively estimates its probability distribution
    - This is context coding using conditional probability



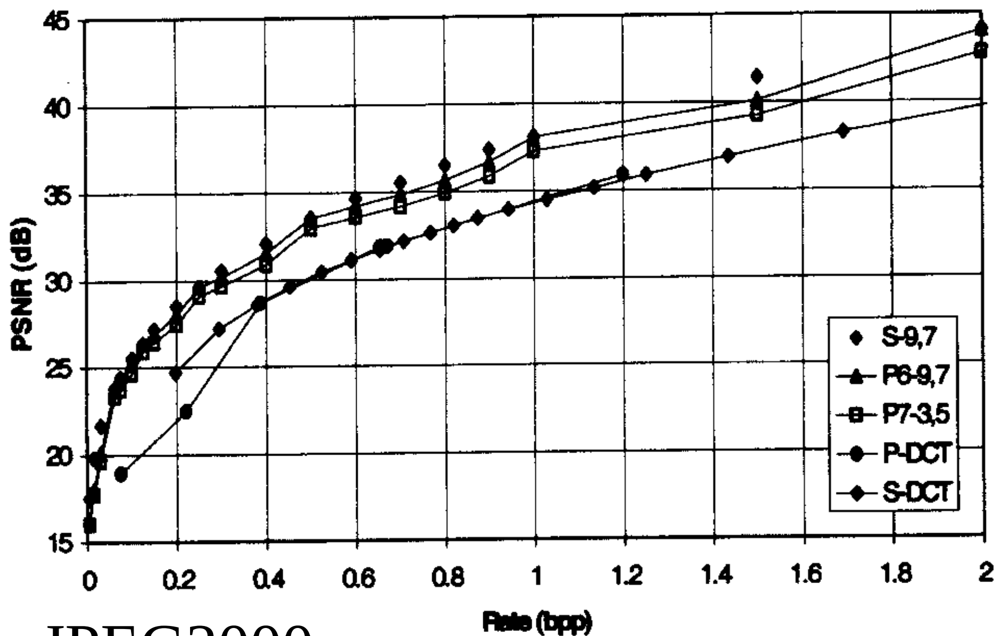
- Recall: Two Main Types of Scalability Structure
  - Resolution Scalability: drop components from high resolution sub-bands
  - Distortion Scalability: drop LSB's from embedded quantization
- Quality Layers: Allow management of this scalability structure
  - Specifies how each code-block should be truncated in relation to others
  - Each layer in the embedded stream provides a progressive improvement
  - Layer structure is optimized during compression for optimal embedding



This figure is Fig. 4 in [1]

Performance of JPEG2000 vs. JPEG [1]

This figure is
Fig. 21 in [1]



- Three different “modes” for JPEG2000
  - S-9,7: Single Layer using 9/7 Wavelet
    - Single layer is not distortion scalable; new stream is generated for each rate under test
  - P6-9,7: Six-Layer Quality Progressive using 9/7 Wavelet
  - P7-3,5: Seven-Layer Quality Progressive using 5/3 Wavelet
- Two different “modes” for JPEG (Optimized Huffman Tables)
  - P-DCT: Progressive Mode – S-DCT: Sequential Mode