Basic Image Compression
or
Compression 101

Bernie Brower

EE- NGA - Kodak

beranrd.brower@kodak.com

585 253-5293

Derived from tutorial from Majid Rabbani

BWC 101 Outline

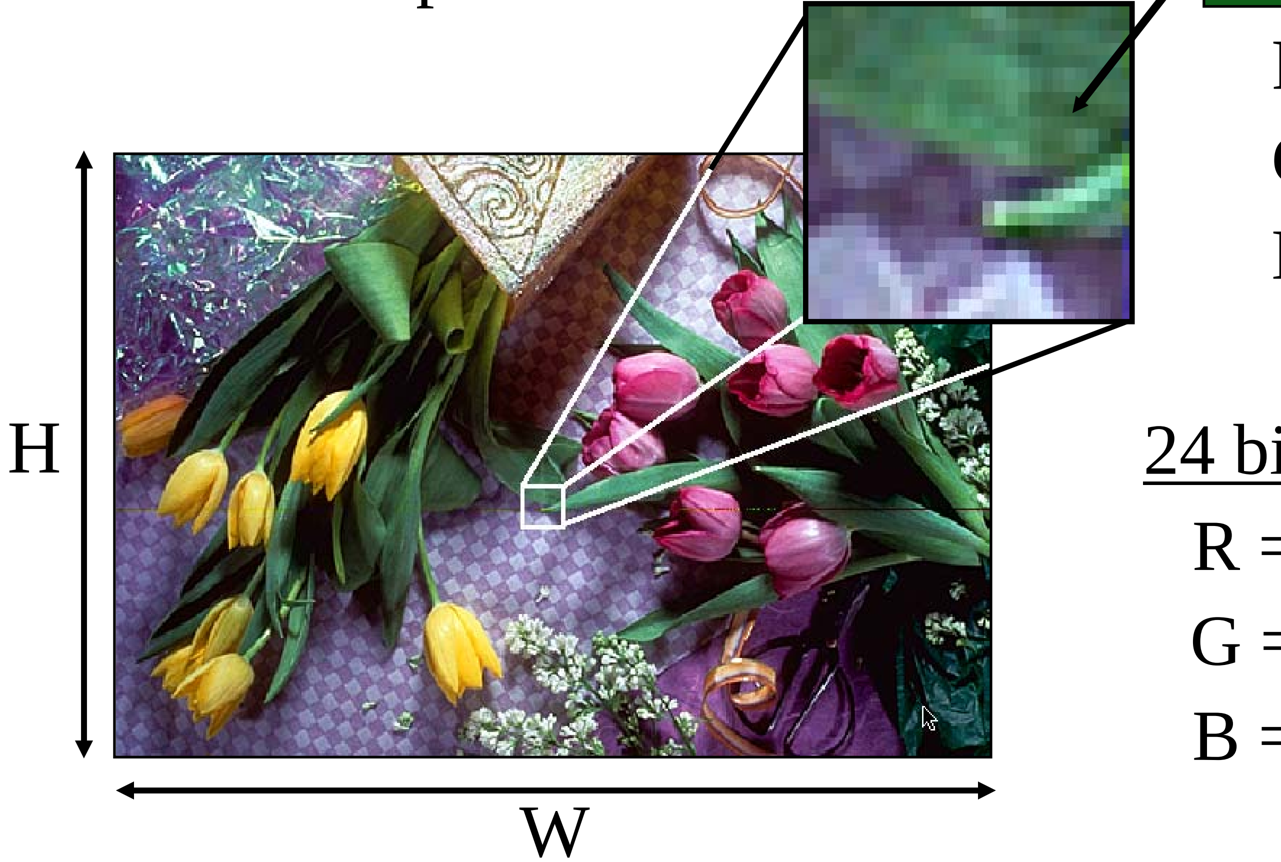
- Motivation for image compression
- Why images can be compressed
- Image compression basics
  - Transform (DPCM, DCT, Wavelet)
  - Quantization
  - Encoding (Huffman, arithmetic)
  - Rate Control
  - Color and Multidimensional transforms
- Selecting a compression algorithm
- Image compression standards
- Image compression systems
  - JPEG DCT
  - JPEG 2000
- Current Dissemination of Imagery

Digital Images

- Digital images are produced by:
  - Remote sensing satellites and aircraft
  - Digitized film and paper
  - Handheld cameras
  - Synthetic imaging (computer generation)
- Regardless, digital images are sampled in;
  - Space ⇨ Resolution (GSD, DPI)
  - Amplitude ⇨ Bit depth (bits per pixel)
  - Spectral ⇨ Resolution (# of bands/color)
- The area coverage or the size of the image "footprint" is dependent on the field of view of the camera system
- [High resolution] x [large bit depth] x [large size] x [large # of bands] = Storage and Transmission challenges

Digital Image Representation

W : H = Aspect Ratio



<u>Pixel</u>

Red = 15

Green = 95

Blue = 27

<u>24 bits per Pixel</u>

$R = 0 \rightarrow 255$

$G = 0 \rightarrow 255$

$B = 0 \rightarrow 255$

Bits and Bytes

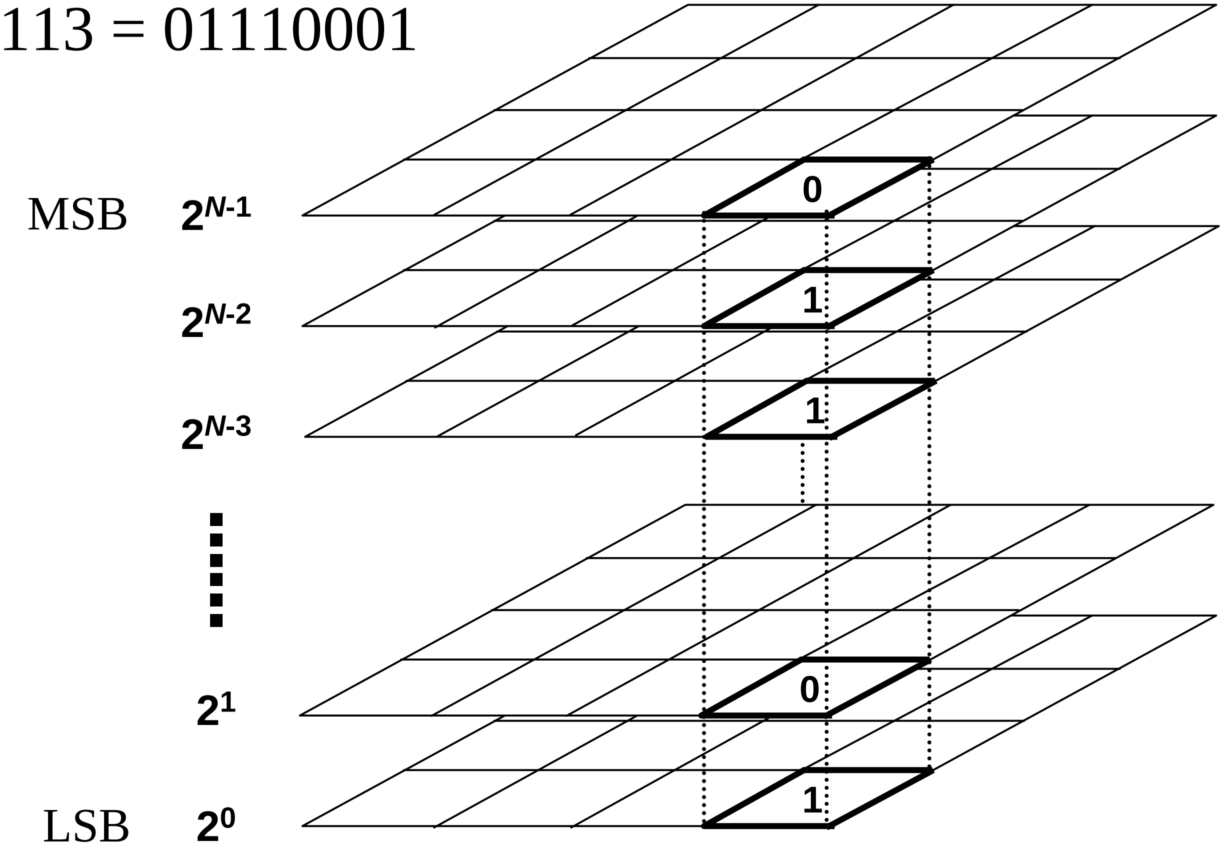
The following terminology is often used when referring to digital image sizes (**bit** is abbreviation for binary digit):

bit (b)	1 bit	1 bit
byte (B)	8 bits	8 bits
kilobyte (KB)	2^{10} bytes	10^{3} + 2.4%
megabyte (MB)	2^{20} bytes	10^{6} + 4.9%
gigabyte (GB)	2^{30} bytes	10^{9} + 7.4%
terabyte (TB)	2^{40} bytes	10^{12} + 10%

Bit Plane Representation

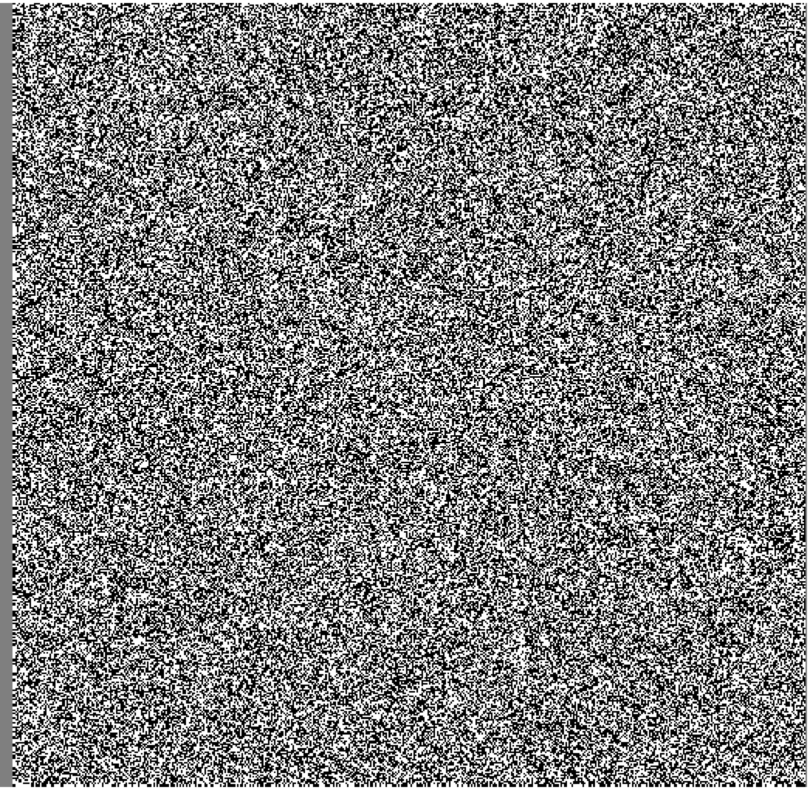
An image with *n* x *m* pixels each with a bit-depth of *N* bits can be represented by *N* binary bit planes of size *n* x *m* ranging from the most significant bit (MSB) to the least significant bit (LSB).

$x(k,l) = 113 = 01110001$



Example: Lena MSB and LSB Planes

Most significant bit plane



Least significant bit plane

Examples of Digital Still Images

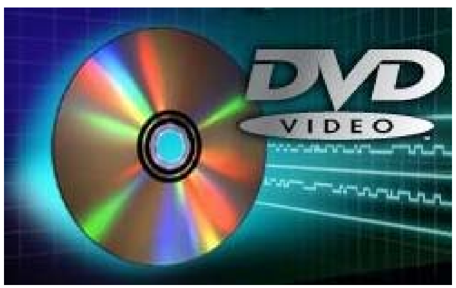
Application	Pixels	Lines	bpp	Filesize
Facsimile	1728	1100	1	240 KB
Icons	192	128	24	72 KB
Still-Video	768	512	24	1.1 MB
Kodak Picture CD	1536	1024	24	4.5 MB
LS 633 (Kodak)	2041	1533	24	8.9 MB
LS-443 (Kodak)	2448	1632	24	11.4 MB
Kodak Photo CD	3072	2048	24	18 MB
DCS Pro 14n	4500	3000	36	61.7 MB

Examples of Digital Video Sequences

Application	Pixels	Lines	bpp	f/s	Data Rate
Video Phone 4:2:0	176	144	12	15	540 KB/s
SIF – 4:2:0	352	240	12	30	3.6 MB/s
CCIR 601 - 4:2:0	720	480	12	30	15 MB/s
CCIR 601 - 4:4:4	720	480	24	30	30 MB/s
HDTV (Interlace)	1920	1080	24	30	180 MB/s
HDTV (progressive)	1280	720	24	60	160 MB/s
Cineon	3656	2664	30	24	1 GB/s

Examples of Storage Media

Optical



Magnetic

Solid State

Examples of Storage Media

Media	Type	Capacity
Floppy Disk	Magnetic	1.4 MB
Zip Disk	Magnetic	100 & 250 MB
Jaz Disk	Magnetic	1 & 2 GB
8mm Cartridge	Magnetic	Up to 70 GB
D2C	Magnetic	25 GB
CD-ROM	Optical	650 MB
DVD-120mm	Optical	4.7 GB
DVD-80mm	Optical	1.9 GB
Compact Flash	Solid State	1-200 MB
Memory Stick	Solid State	32 & 64 MB

Examples of Transmission Media

Media	Bandwidth
Dial-Up Telephone	33.6-56 Kb/s
ISDN (Integrated Services Digital Network)	64 Kb/s
T1 (24 x ISDN)	1.544 Mb/s
Ethernet	10 Mb/s
Fast Ethernet	100 Mb/s
Ultra-Fast Ethernet	1 Gb/s
Cable Modem	1-2 Mb/s
FDDI (Fiber Distributed Data Interchange)	100 Mb/s

Examples of Transmission Media

Media	Bandwidth
Cell Phone (US)	9.6 – 14.4 Kb/s
IR (Infra-Red) Line-of-sight	115 kb/s – 16 Mb/s
Bluetooth RF	400 Kb/s
Bluetooth RF (Asymmetric)	741 Kb/s Downstream 56 Kb/s Upstream
Kodak RF	16 Mb/s
DSL (Asymmetric)	384 Kb/s – 1.5 Mb/s Downstream 128 Kb/s – 750 Kb/s Upstream

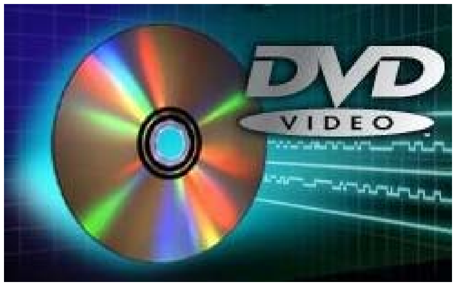
Image Transmission Example

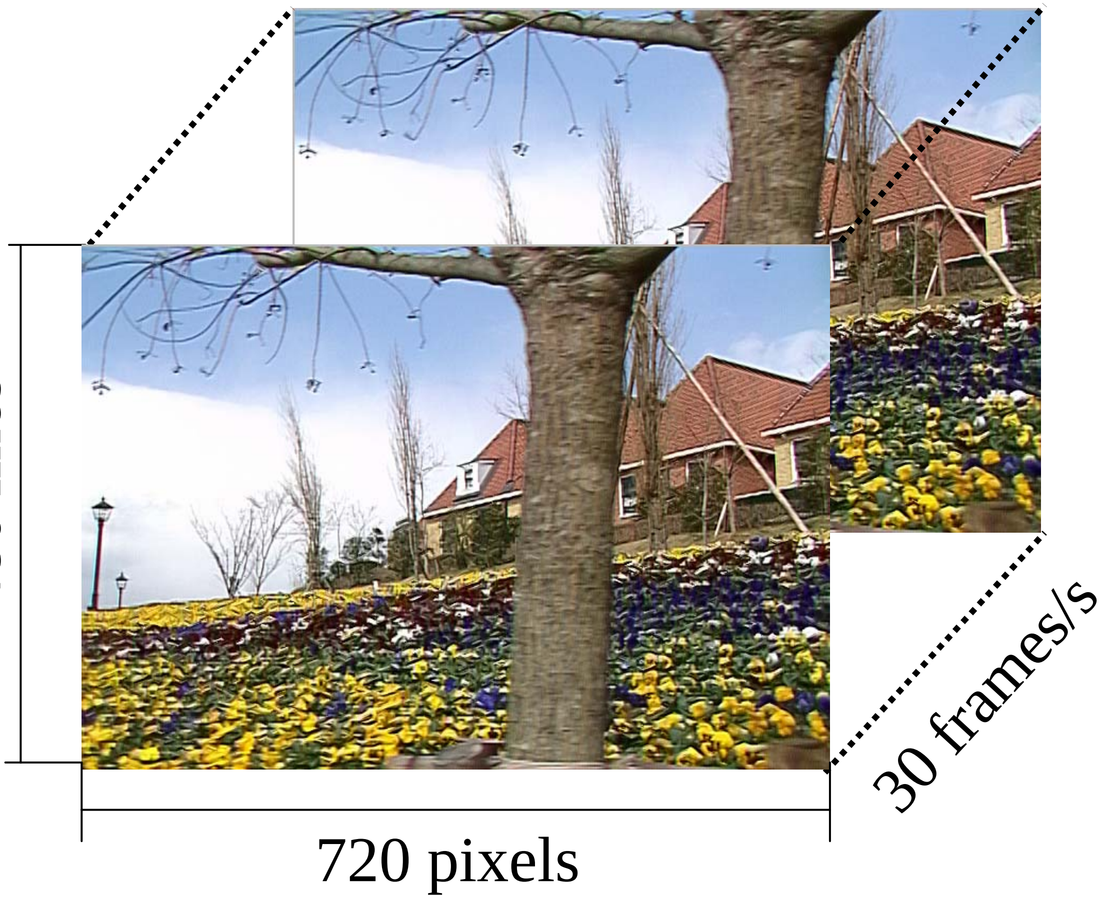
Media	Time seconds
Cell Phone	417
28.8 Modem	139
ISDN	62
Asymmetric DSL (Downstream)	10
Bluetooth	5.5
Cable Modem (1 Mb/s)	4
IR (4 Mb/s)	1
RF (16 Mb/s)	1/4

For an uncompressed VGA 640 x 480 (4:2:0) image ~500KB

Movies on DVD

- The full resolution RGB uncompressed DVD movie has a rate of ~60 MB/sec, i.e., a DVD disc holds ~1 minute of uncompressed video!





Aerial Photograph

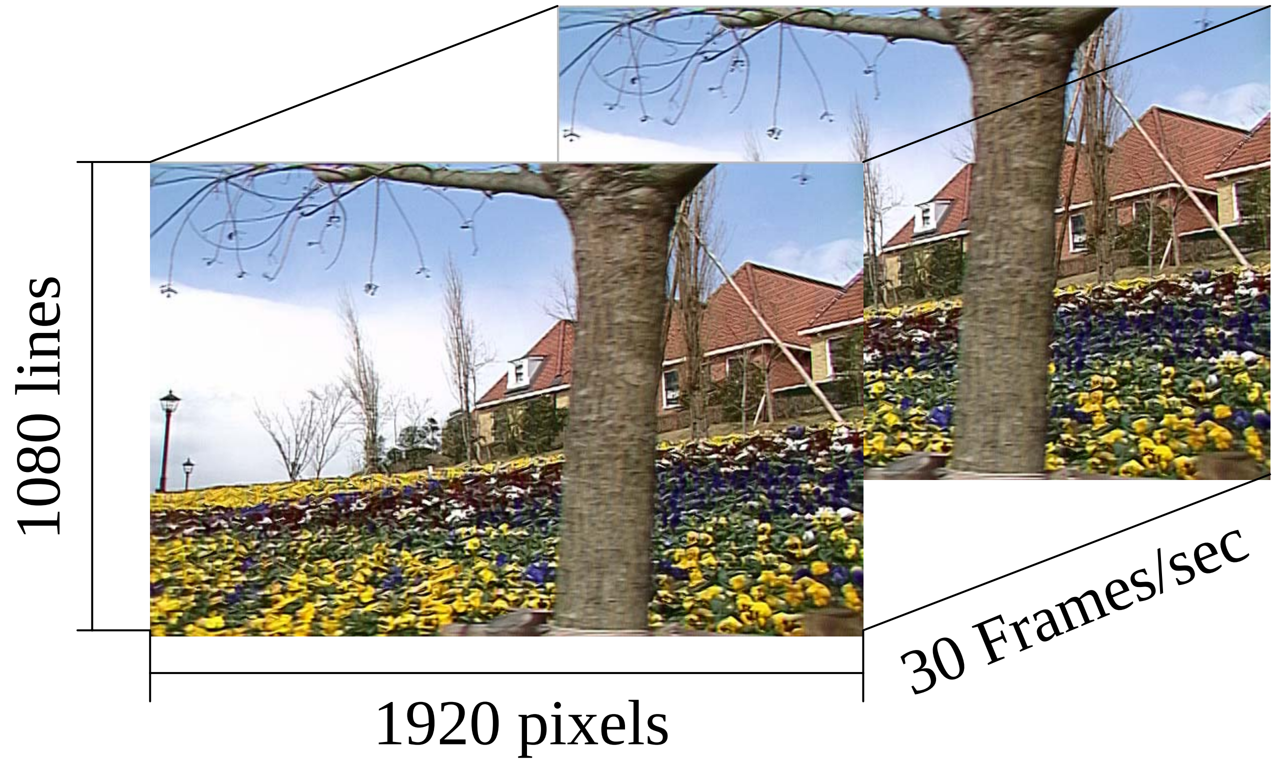
- 10" Aerial Mapping Camera at about 6" resolution
- 2.35 Square Miles of Coverage (about 1.5 Miles on a side)
- 16,000-by-16,000 pixels, 3 bands, 8 bits per band
- 730 MB uncompressed image does not fit on CD

Satellite Image

- IKONOS or QuickBird Images
- ~ 10K-by-10K, 5 bands, 10 bits per pixel

HDTV Transmission

- 1920 pixels × 1080 lines × 3 colors × 30 f/s ~ 180 MB/s
- At 18 Mb/s, a compression ratio of ~80:1 is required.



Need For Image Compression

Application	Size / Rate	CR	Method
Digital Camera	1-10 MB	>10	JPEG
Facsimile	240 KB	20	G3 & G4
CD-I (SIF 4:2:0)	3.6 MB/s	25	MPEG 1
DVD (CCIR 601 4:2:0)	15 MB/s	25	MPEG 2
HDTV	89 MB/s	70	MPEG 2
Teleconferencing	9 MB/s	50	H.261
Video Phone (QCIF)	0.54 MB/s	80	H.263

Image Compression

- Aims at finding methods for reducing the number of bits needed to represent a digital image without compromising the required image quality for a given application.
- Image compression is used to:
  - Reduce memory for image storage.
  - Reduce the bandwidth/time required for image transmission.
  - Increase effective data transfer rate.

Why images can be compressed

What is digital image compression

- Image compression is the art/science of finding efficient representations for digital images in order to:
  - Reduce the memory required for their storage,
  - Reduce the effective data access time when reading from storage devices,
  - Reduce the bandwidth and/or the time required for their transfer across communication channels.

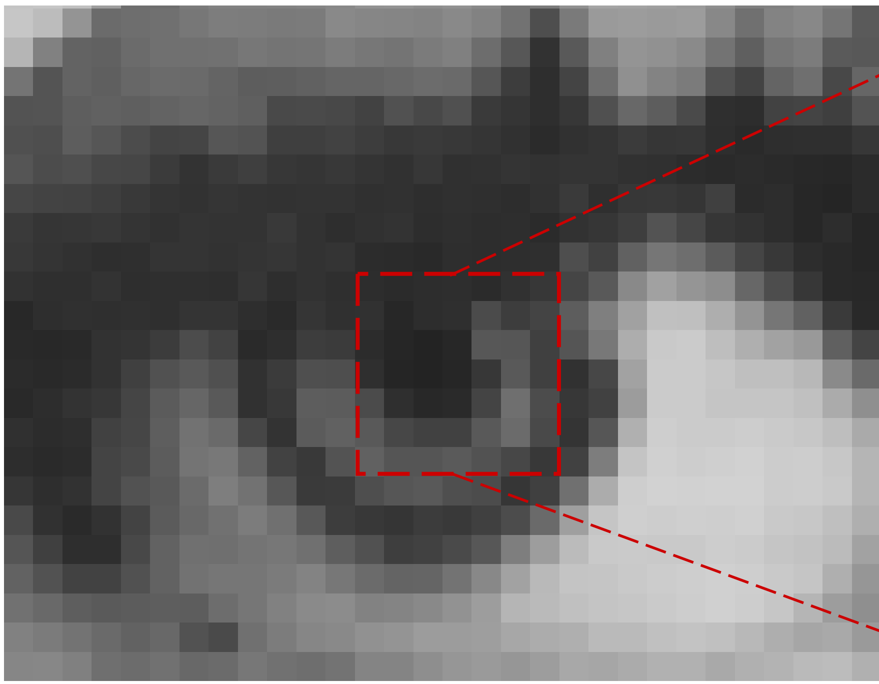
- The goal is to achieve the desired bit rate without compromising the image quality required for a given application.

Why can images be compressed?

- Image compression can be achieved because image data are often highly redundant and/or irrelevant. The redundancy or irrelevancy exist in **spatial**, **spectral**, and **temporal** forms.
  - **Redundancy** relates to the statistical properties of an image (e.g., pixel-to-pixel correlation, spectral (RGB) correlation, frame-to-frame similarity, etc.) and is a function of resolution, bit-depth, image noise, and image detail.
  - **Irrelevancy** relates to an observer viewing an image (HVS spatial and temporal CSF, visual masking, etc.) and is a function of image resolution, noise, detail, and viewing conditions.

Spatial Redundancy And Irrelevancy

- What is the value of the missing pixel? **(39)**
- How critical is its exact reproduction?



46	44	46	47	44	50	56
50	40	46	48	76	62	69
45	39	36	40	88	87	65
48	38	36	?	56	90	65
76	48	41	43	69	112	77
90	72	66	66	90	108	74
98	86	86	91	83	72	57

Example: Original and Difference Images

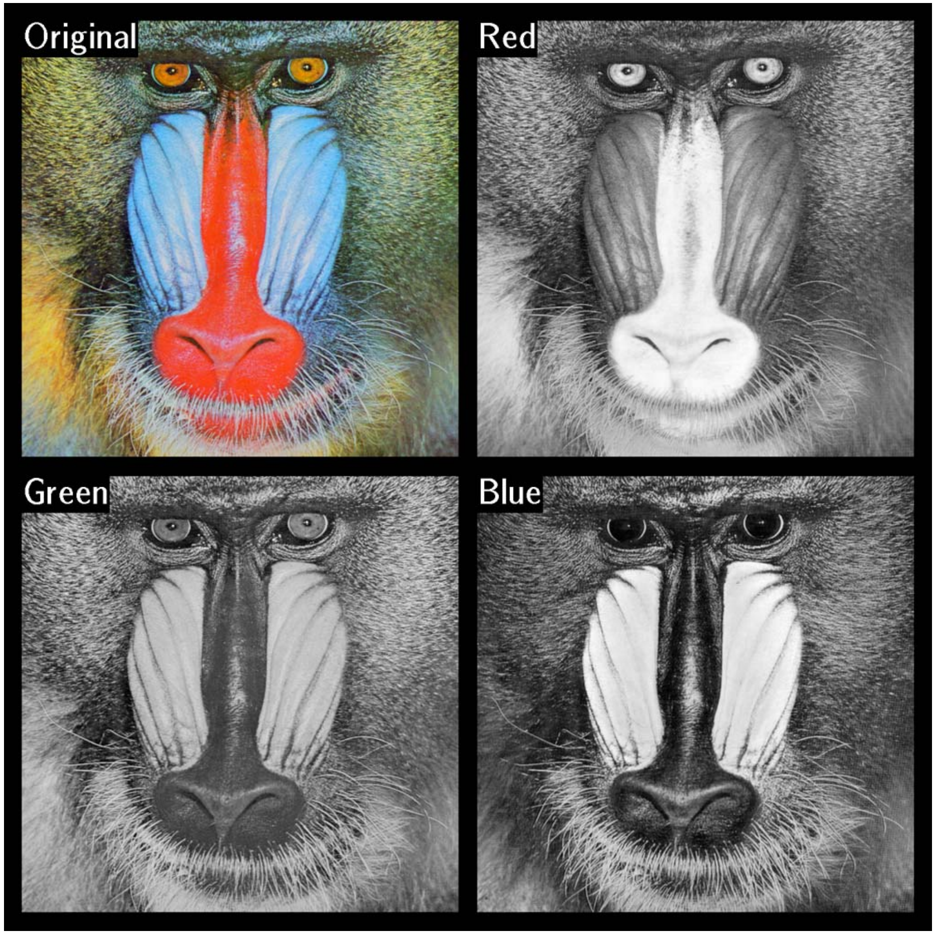
139 139	144 +5	149 +5	153 +4	155 +2	155 0	155 0	155 0
144 +5	151 +7	153 +2	156 +3	159 +3	156 -3	156 0	156 0
150 +6	155 +5	160 +5	163 +3	158 -5	156 -2	155 0	155 0
159 +9	161 +2	162 +1	160 -2	160 0	159 -1	159 0	159 0
159 0	160 +1	161 +1	162 +1	162 0	155 -7	155 0	155 0
161 +2	161 0	161 0	161 0	160 -1	157 -3	157 0	157 0
162 +1	162 0	161 -1	163 +2	162 -1	157 -5	157 0	157 0
162 0	162 0	161 -1	161 0	163 +2	158 -5	158 0	158 0

Spatial Redundancy

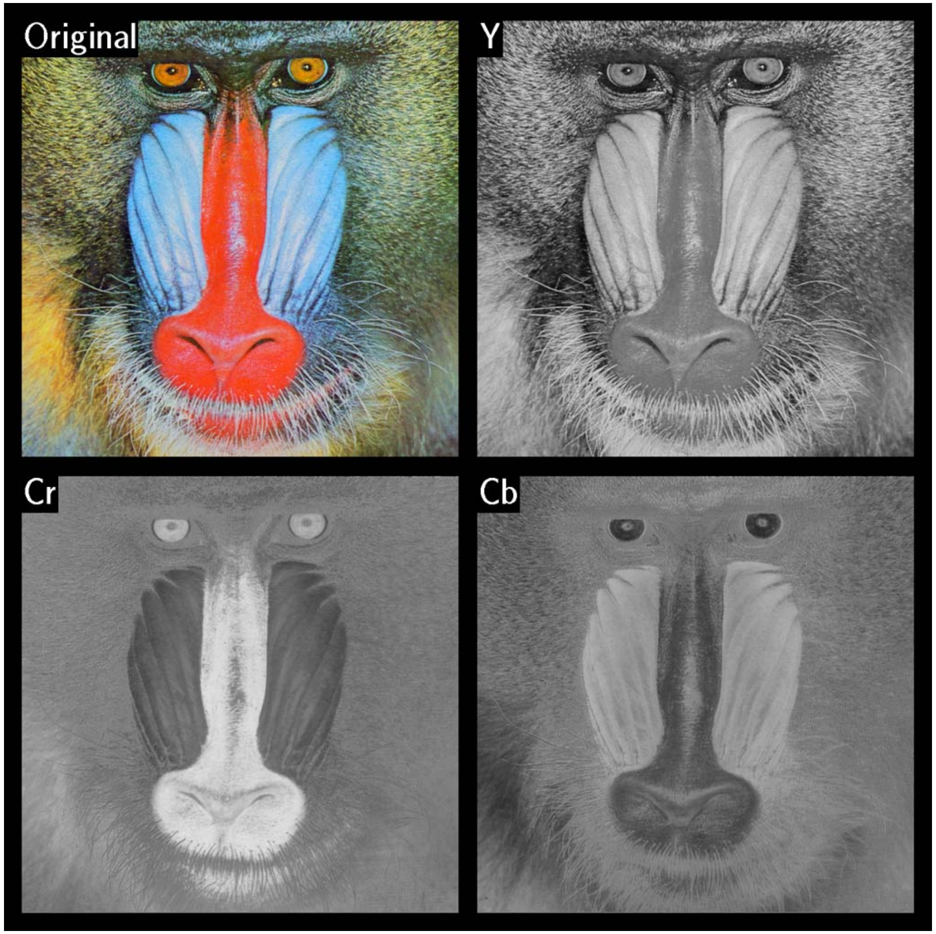
- The difference between two adjacent pixels has a very skewed distribution centered around zero.

Spectral Redundancy

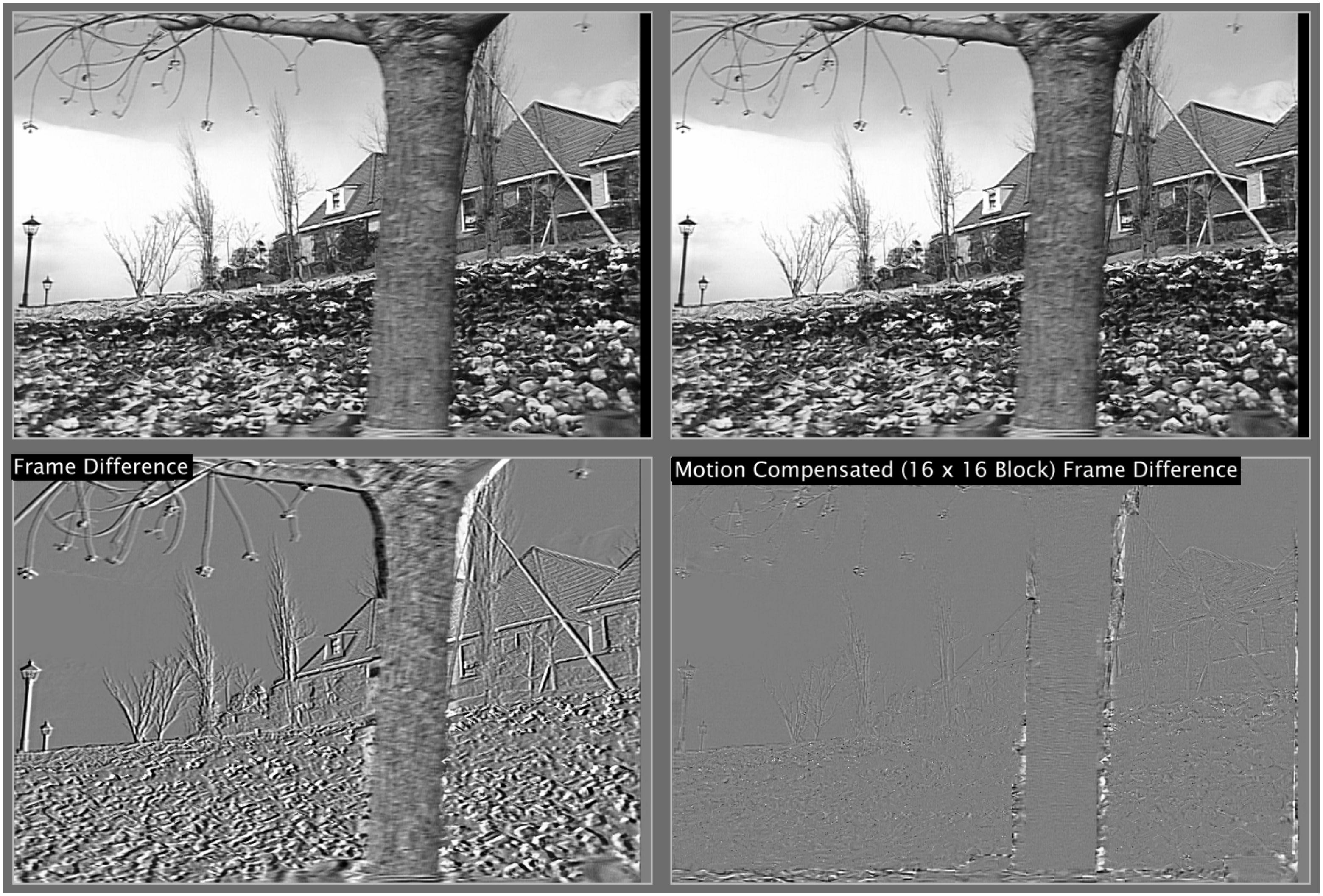
Original RGB



Transformed YCrCb



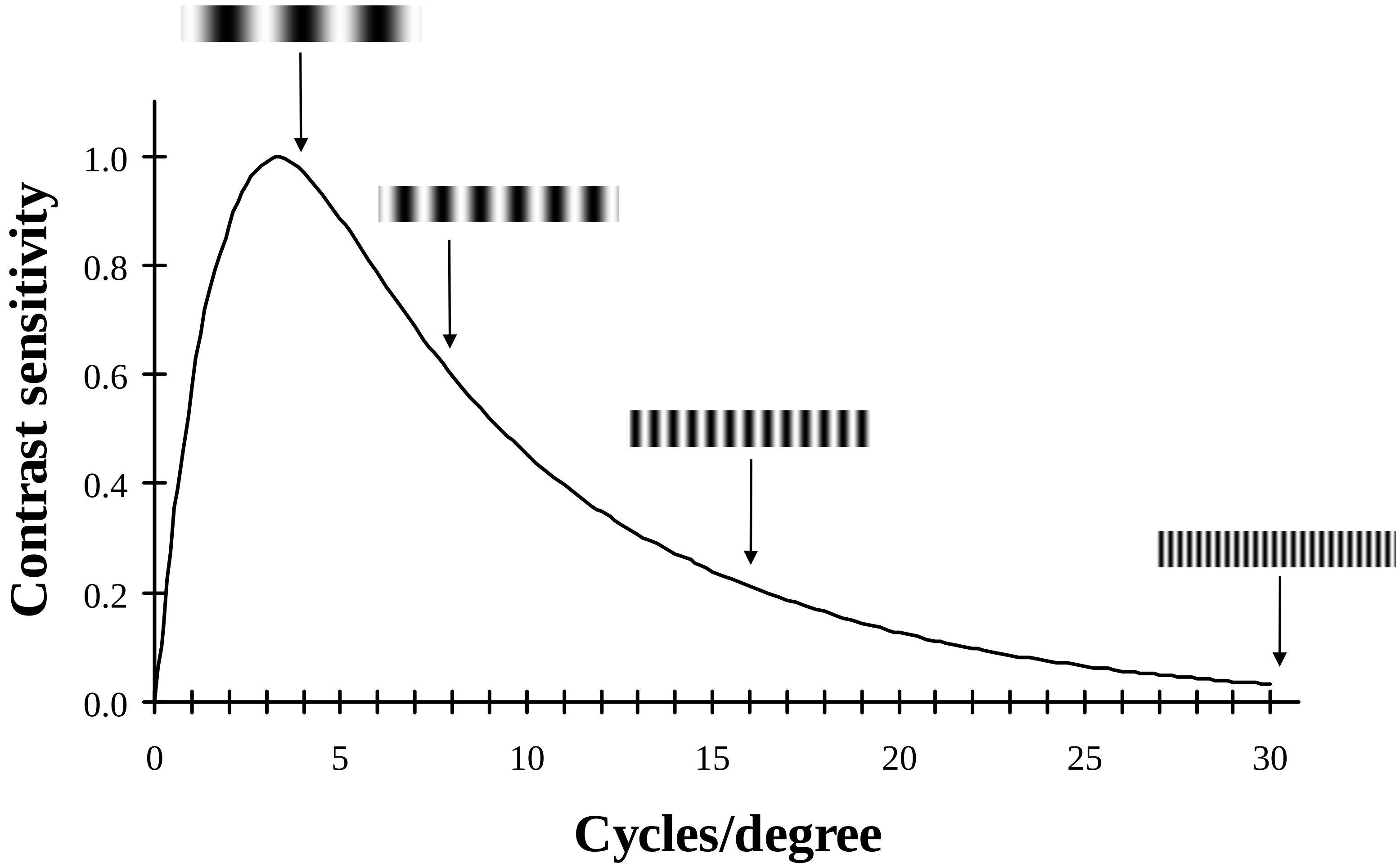
Temporal Redundancy

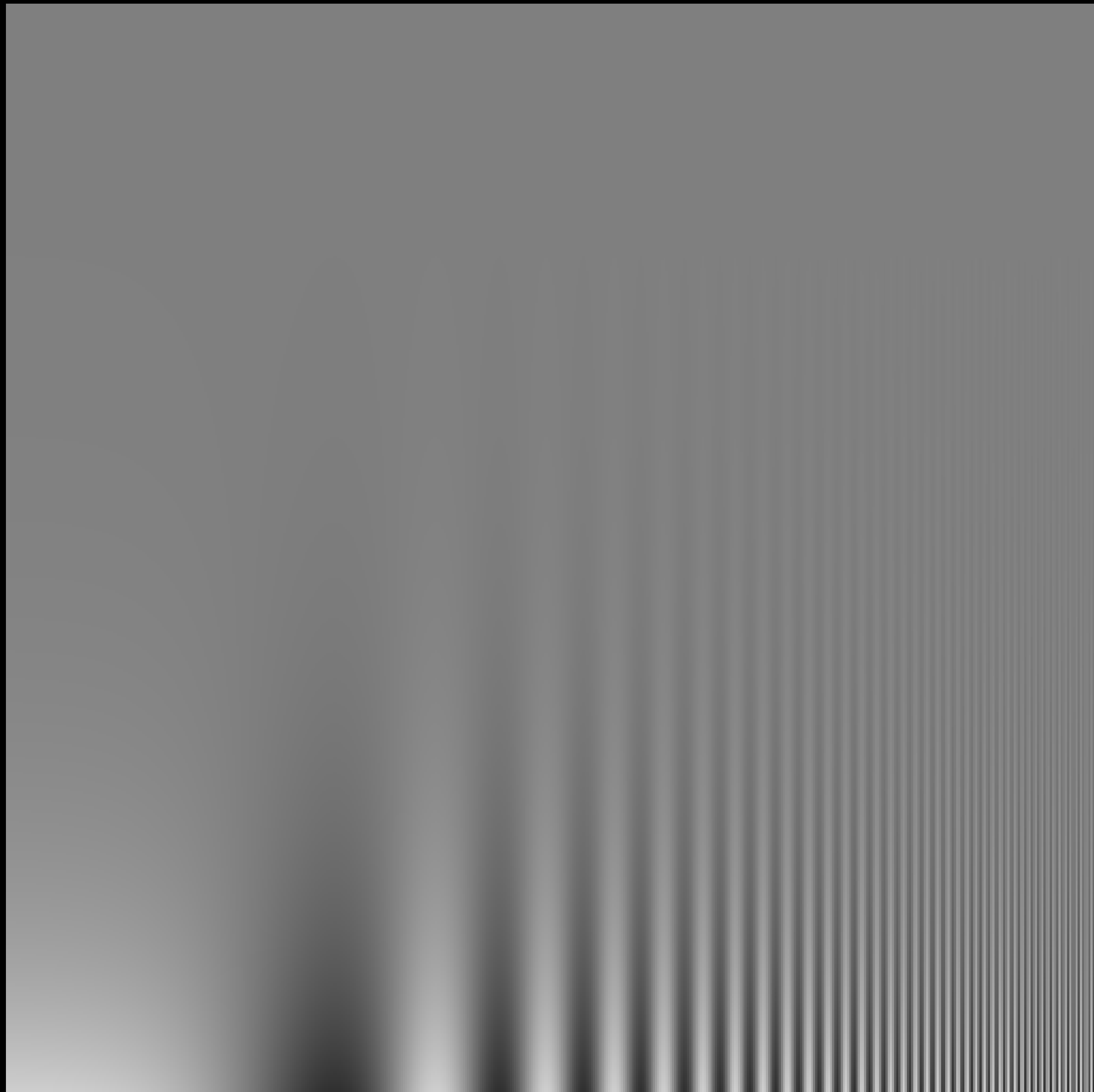


The Human Visual System CSF

- The frequency-dependent behavior of the human visual system (**HVS**) can be characterized by its response to harmonic (sinusoidal) functions.
- For each sinusoid with a given frequency, the amount of contrast needed to elicit a criterion level of response from a neuron is called the **contrast threshold**.
- The inverse of the contrast threshold is called the **contrast sensitivity** and when plotted as a function of frequency is referred to as the **contrast sensitivity function (CSF)**.
- The luminance CSF peaks at around 5 cycles/degree, and rapidly drops off to almost zero at 50 cycles/degree. The chrominance CSF drops even faster.

Example of Luminance CSF

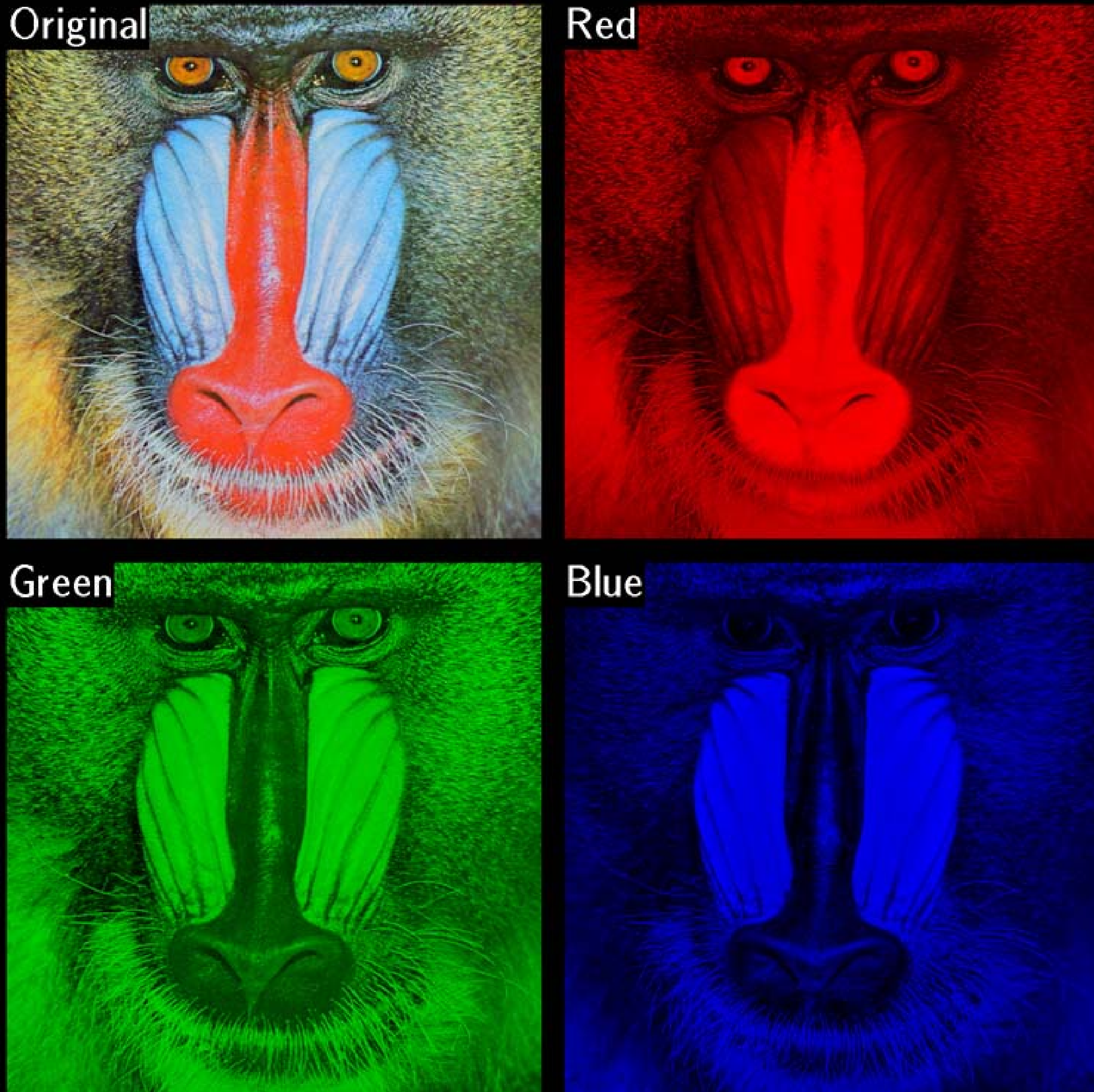


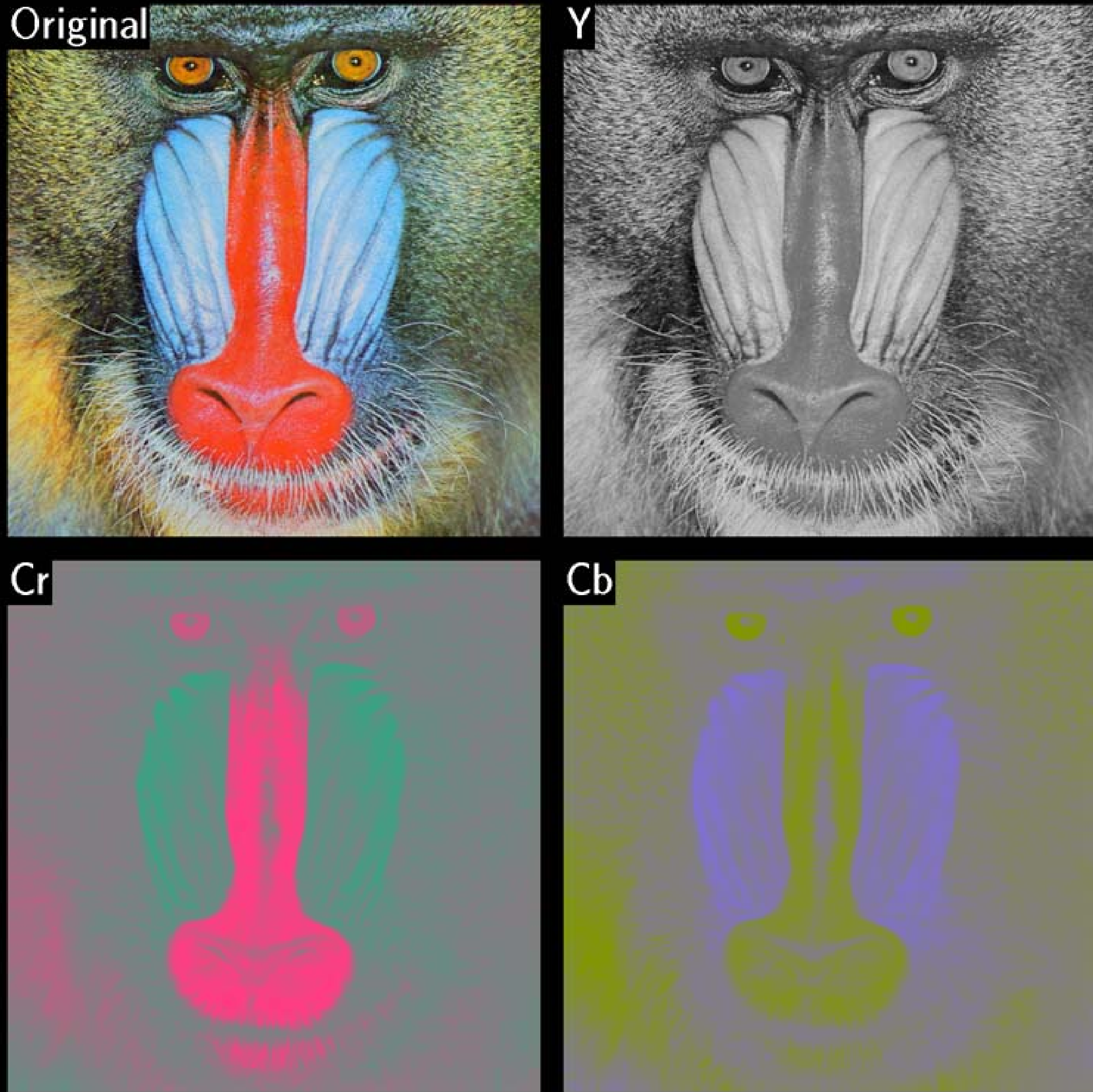


Original (512x512)


Original + Distortion

Distortion


Original
Red
Green
Blue


Original
Y
Cr
Cb

NATIONAL GEOSPATIAL-INTELLIGENCE AGENCY

Kodak

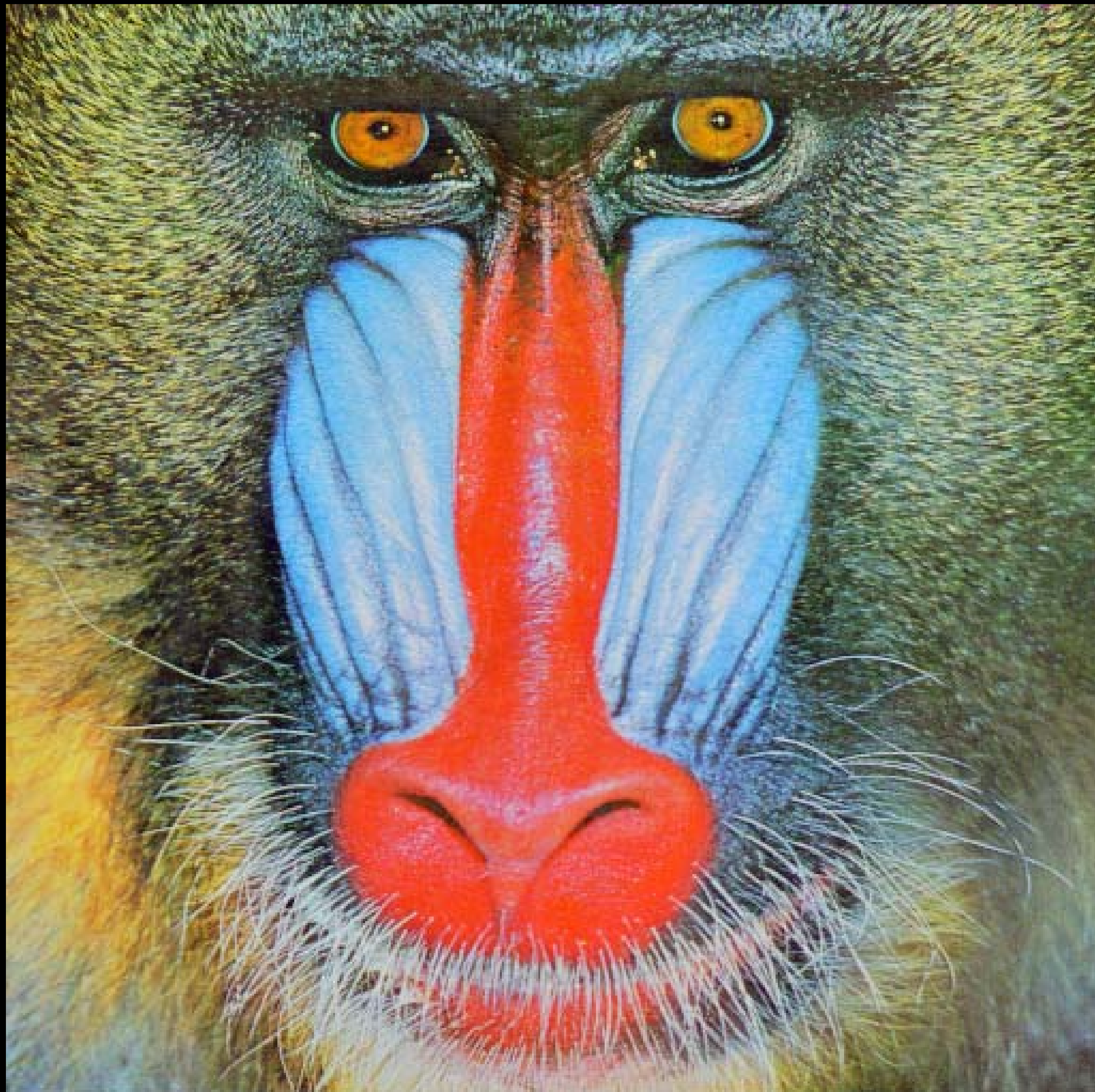
Luma Subsampled 2X

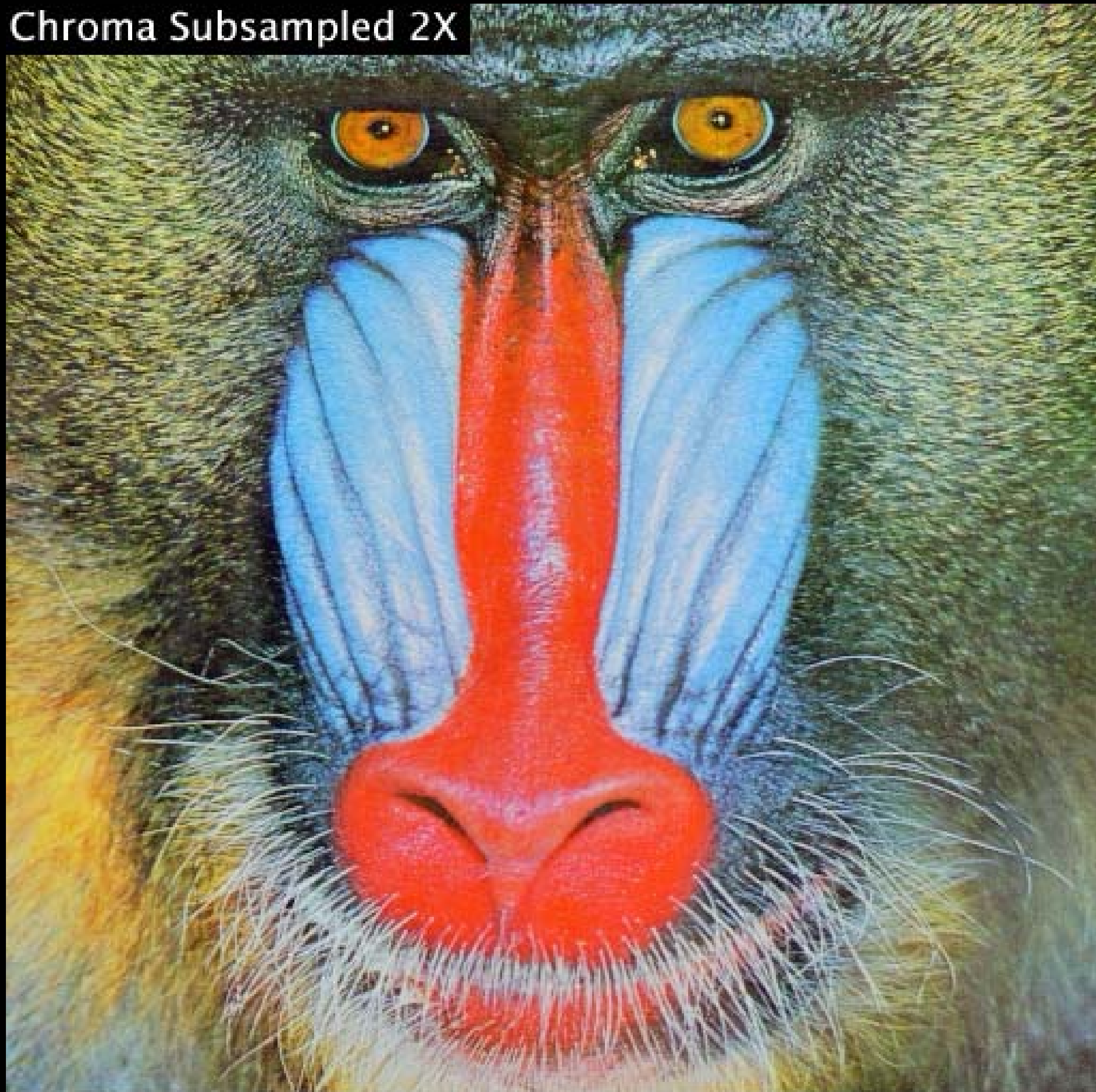
Luma Subsampled 4X


Luma Subsampled 8X

NATIONAL GEOSPATIAL-INTELLIGENCE AGENCY

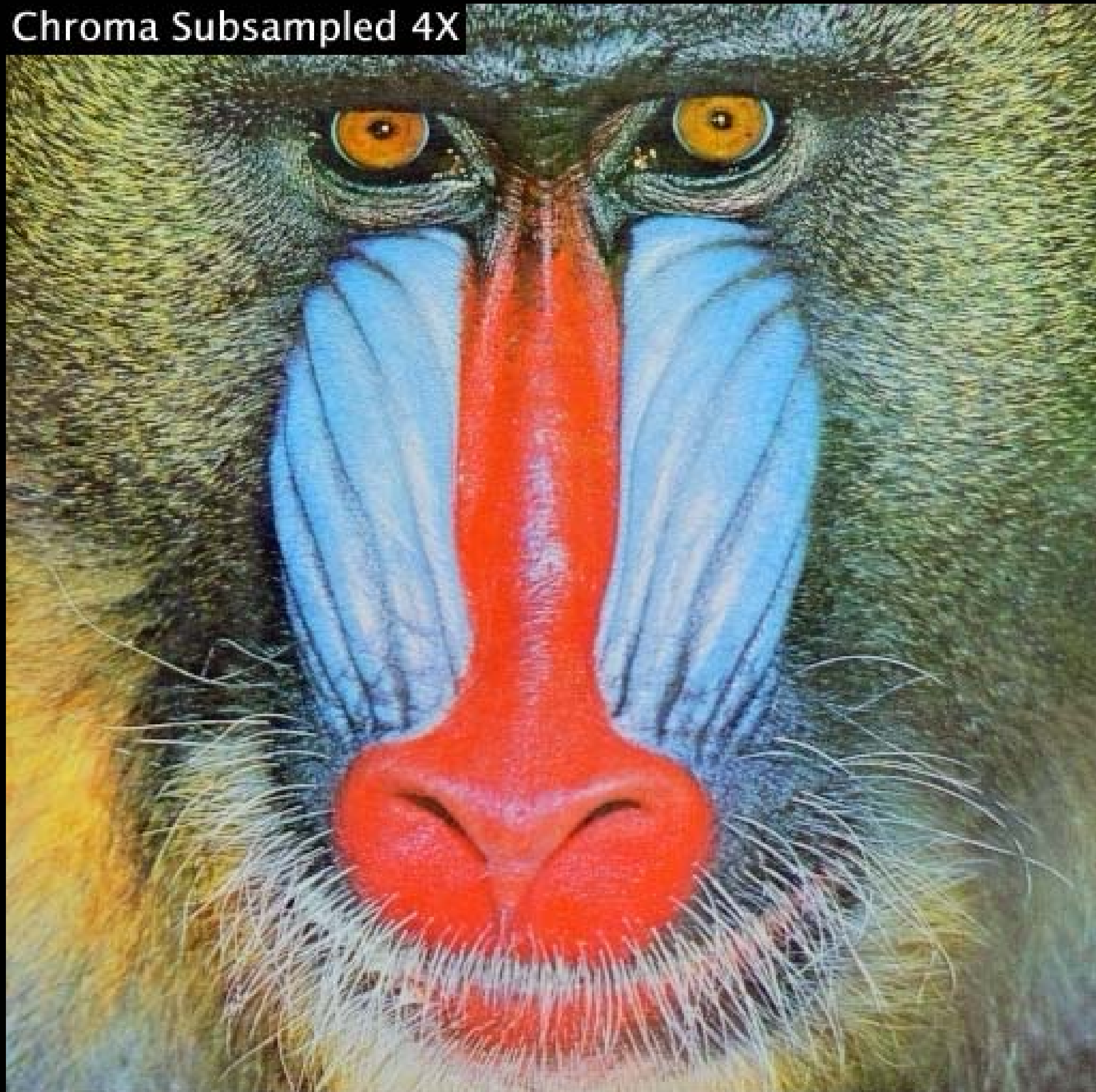
Kodak




Chroma Subsampled 2X

NATIONAL GEOSPATIAL-INTELLIGENCE AGENCY

Kodak



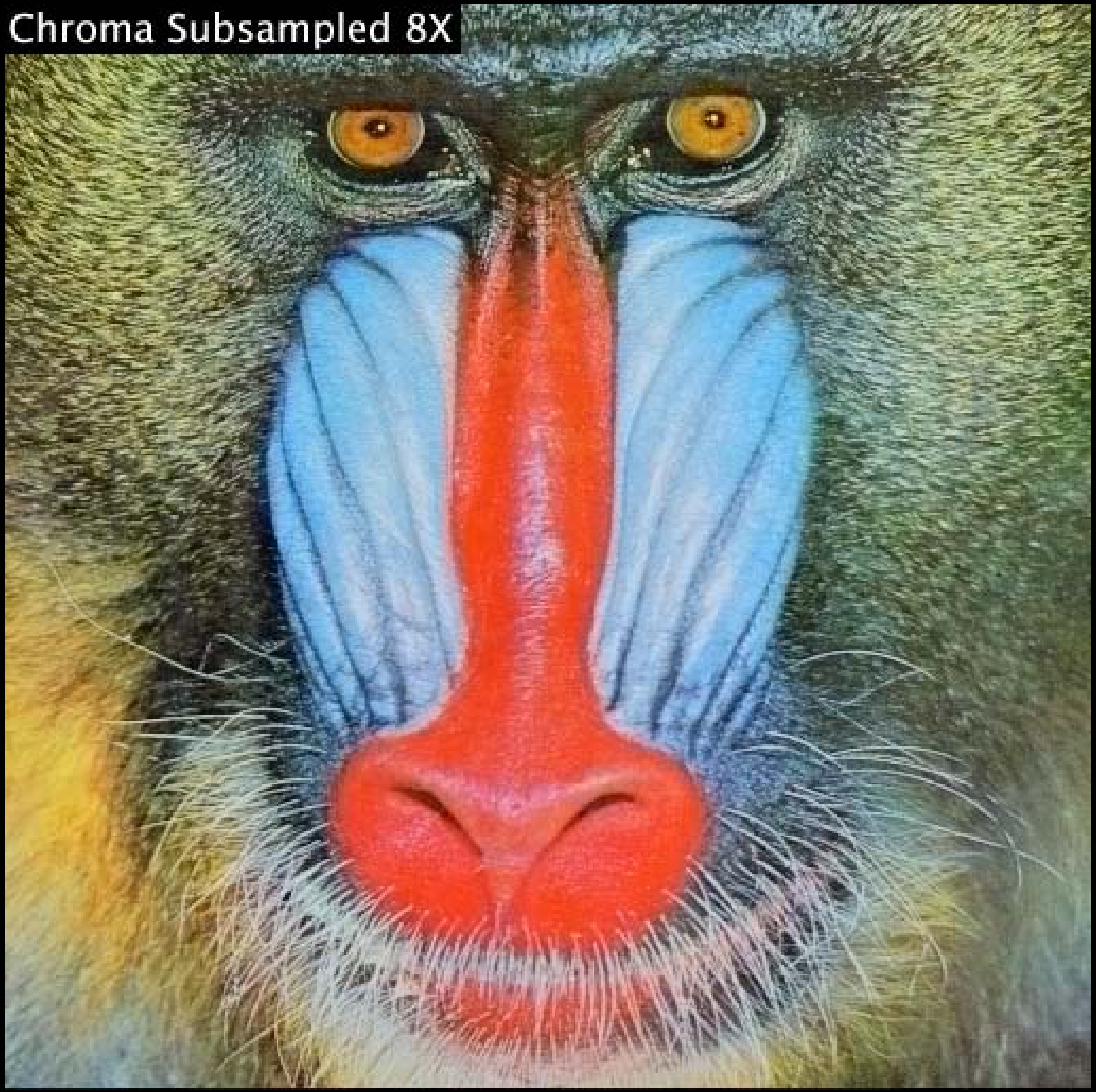

Chroma Subsampled 8X

Image Compression Basics

Basic Strategy in Image Compression

- Ideally, an image compression technique removes redundant and/or irrelevant information, and efficiently encodes what remains.

- Practically, it is often necessary to throw away both non-redundant information and relevant information to achieve the required compression.

- In either case, the trick is finding methods that allow important information to be efficiently extracted and represented.

Some Factors Affecting Achievable Compression

- Sample parameters (spatial resolution, bit depth).
- Sensor characteristics (noise, spectral response).
- Scene content, including noise.
- Image size and viewing distance.
- Display characteristics (noise, light level, non-linearities)
- Post Processing (Sharpening, Dynamic Range Adjustment (DRA), Tone Transfer Curve (TTC))
- Pre-Processing (image formation, registration)
- Observer (IA, machine)
- Required task

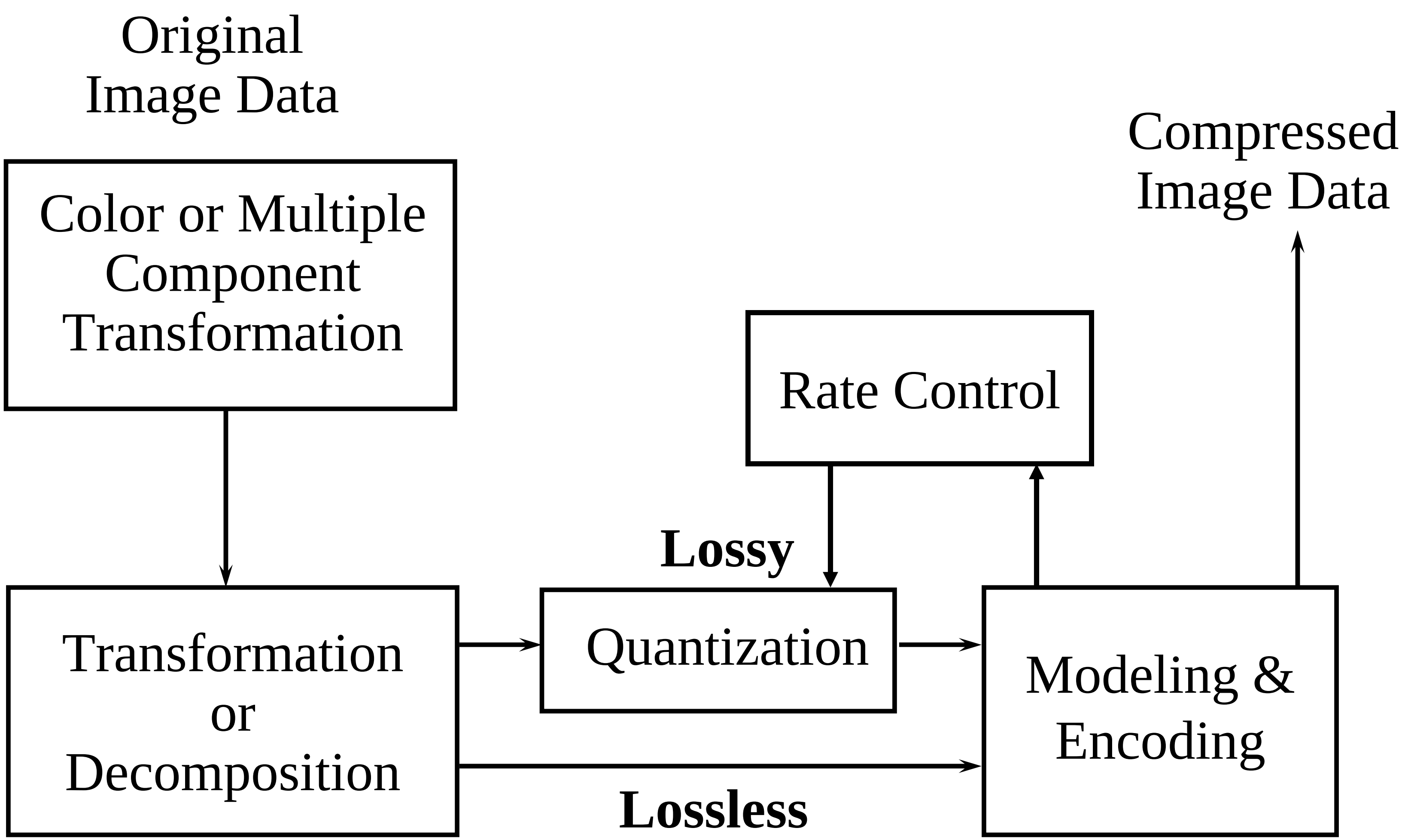
Lossless (Reversible) Compression

- The image after compression and decompression is identical to the original.
- Only the statistical redundancy is exploited to achieve compression.
- Data compression techniques such as LZW or LZ77 are used in GIF, PNG, and TIFF file formats and the Unix "Compress" command.
- Image compression techniques such as lossless JPEG or JPEG-LS perform slightly better.
- Compression ratios are typically ~2:1 for natural imagery but can be much larger for document images.

Lossy (Irreversible) Compression

- The reconstructed image contains degradations with respect to the original image.
- Both the statistical redundancy and the perceptual irrelevancy of image data are exploited.
- Much higher compression ratios compared to lossless.
- Image quality can be traded for compression ratio.
- The term **visually lossless** is often used to characterize lossy compression schemes that result in no visible degradation under a set of designated viewing conditions..

Compression Framework



Transformation

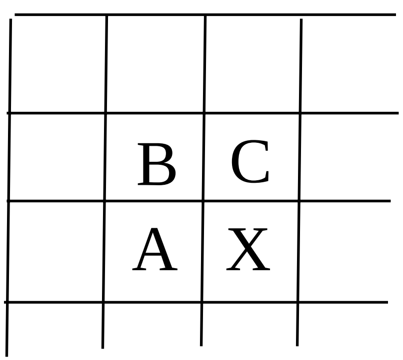
Decomposition or Transformation

- A reversible process (or near-reversible, due to finite precision arithmetic) that reduces redundancy and/or provides an image representation that is more amenable to the efficient extraction and coding of relevant information.

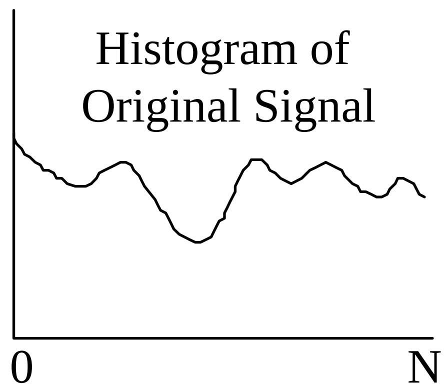
- Examples
  - Block-based linear transformations, e.g. Discrete Cosine Transform (DCT)
  - Wavelet decompositions.
  - Prediction/residual formation, e.g. Differential Pulse Code Modulation (DPCM)
  - Color space transformations, e.g. RGB to YCrCb.
  - Model prediction/residual formation, e.g. Fractals

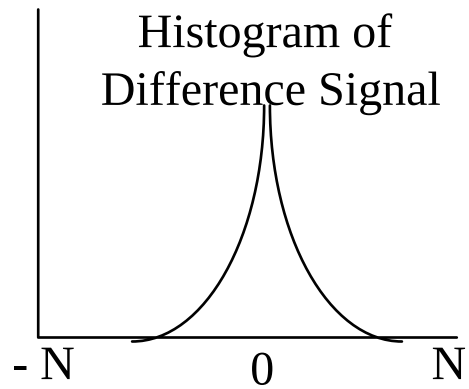
DPCM

- Lossless JPEG and 4.3 DPCM are based on differential pulse code modulation (DPCM).
  - In DPCM, a combination of previously encoded pixels (A, B, C) is used as a prediction (χ) for the current pixel (X).
  - The difference between the actual value and the prediction (χ – X) is encoded using Huffman coding.
    - The quantized difference is encoded in lossy DPCM
  - Properties
    - Low complexity
    - High quality (limited compression)
    - Low memory requirements

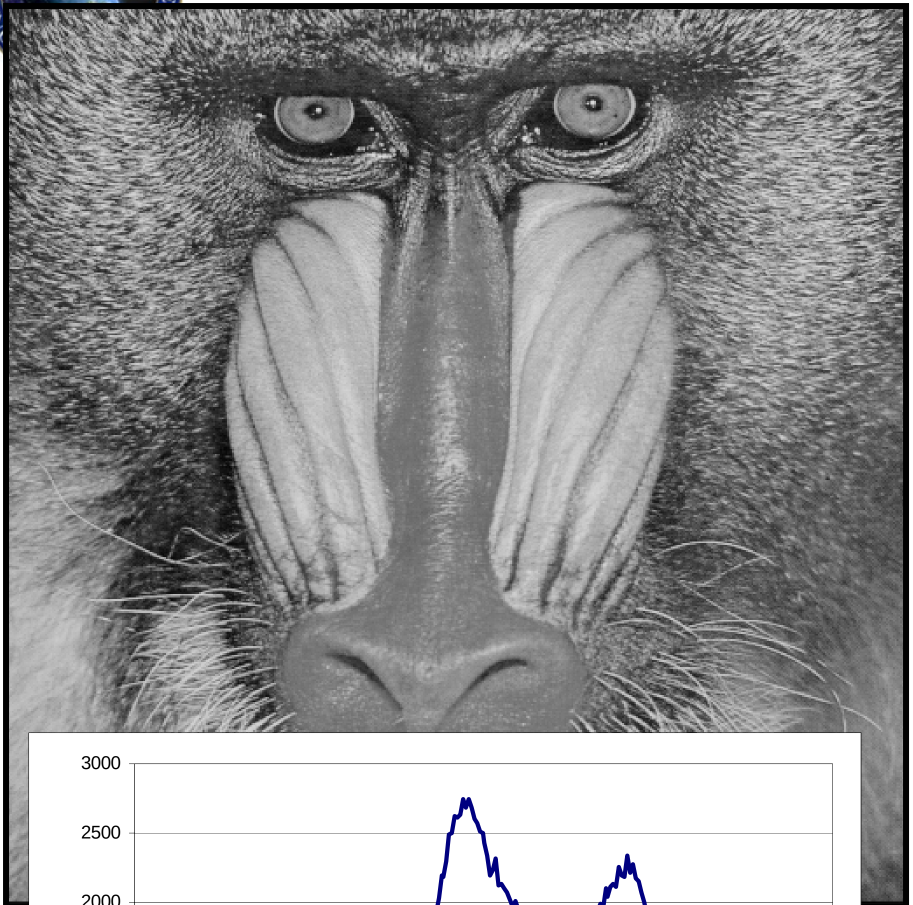


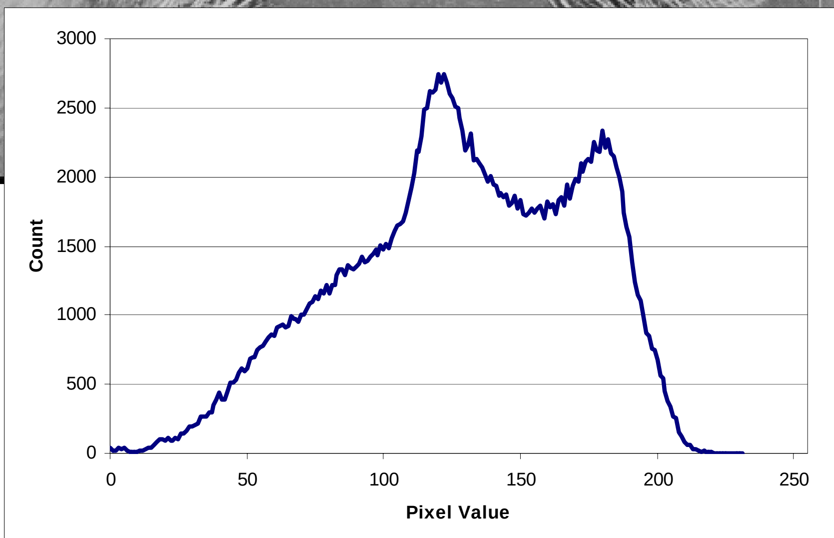
χ = A
χ = (A +C)/2
χ = (A + C - B)



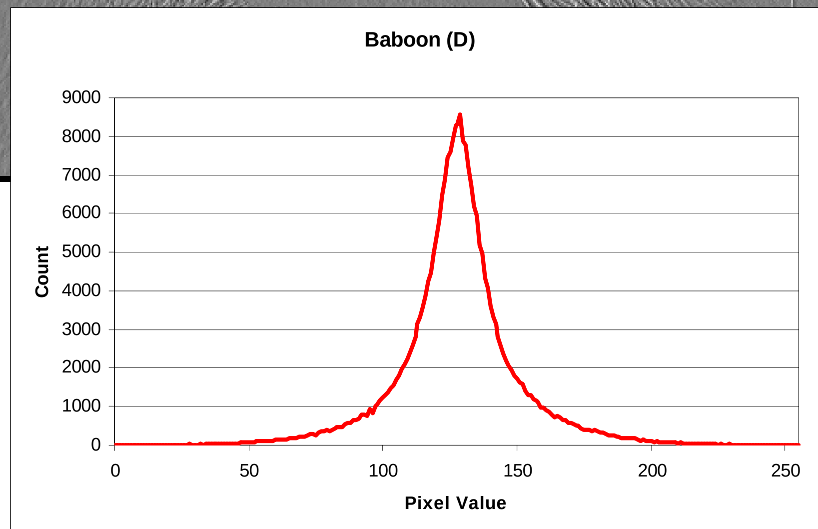


Original





DPCM output



Example Block From Lena Image

Following is an 8x8 block of the Lena image where each pixel value has been level-shifted by subtracting a value of 128.

$$x(k,l) =$$

8	14	23	37	52	68	73	82
6	14	24	37	46	67	74	81
3	11	28	35	48	62	72	82
4	13	22	28	44	61	69	86
5	11	18	30	40	59	72	86
5	9	16	29	39	58	74	83
-1	8	16	31	38	59	75	80
2	11	18	30	37	57	69	82

2-Dimensional 8 x 8 DCT Basis Functions

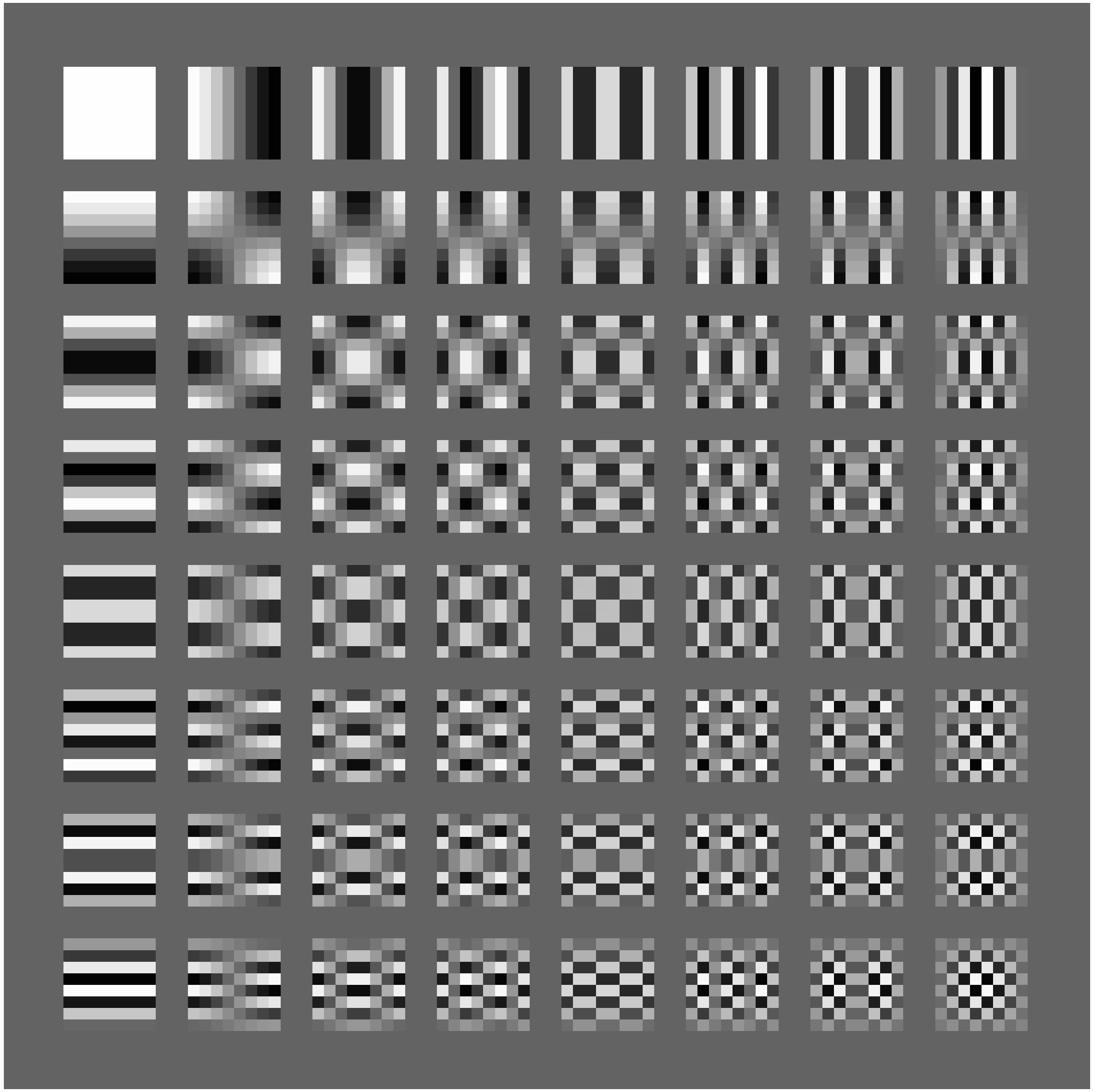
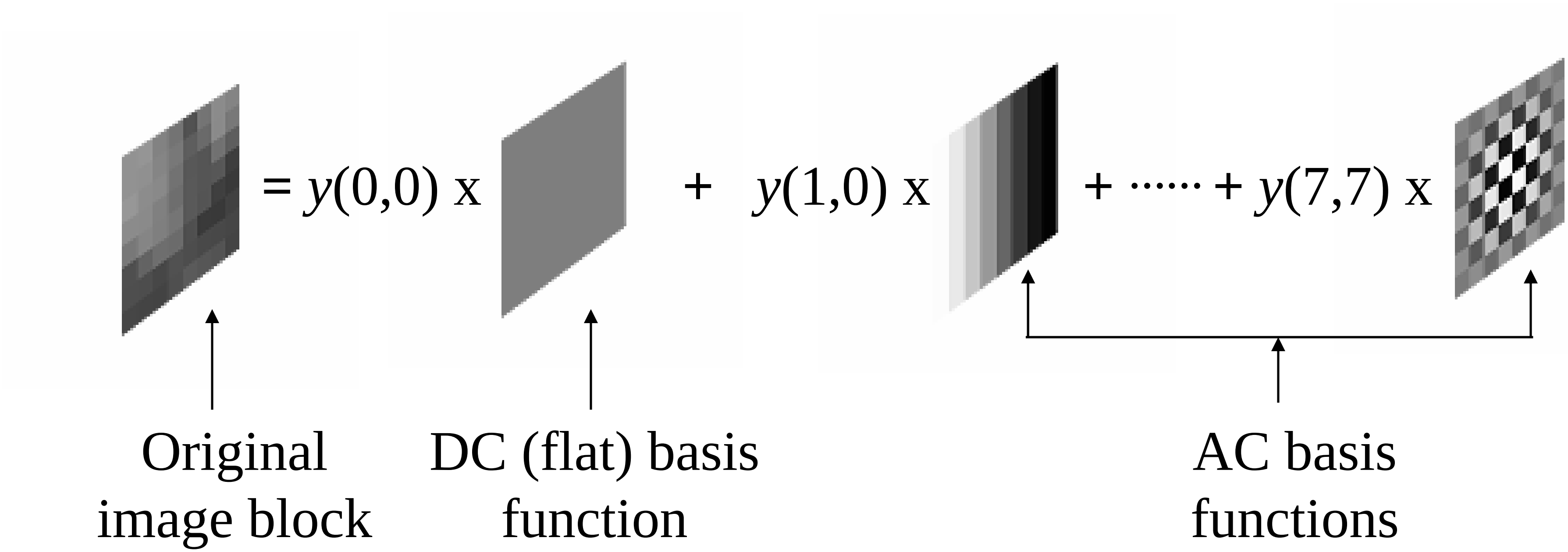


Image Representation with DCT

- DCT coefficients can be viewed as weighting functions that, when applied to the 64 cosine basis functions of various spatial frequencies (8 x 8 templates), will reconstruct the original block.



DCT of 8 x 8 Image Block

The 8 x 8 DCT of the block preserves the block's energy (sum of the squared amplitudes), but it packs the block energy into a small number of DCT coefficients by removing the pixel redundancy or correlation.

DC Value → 327.5

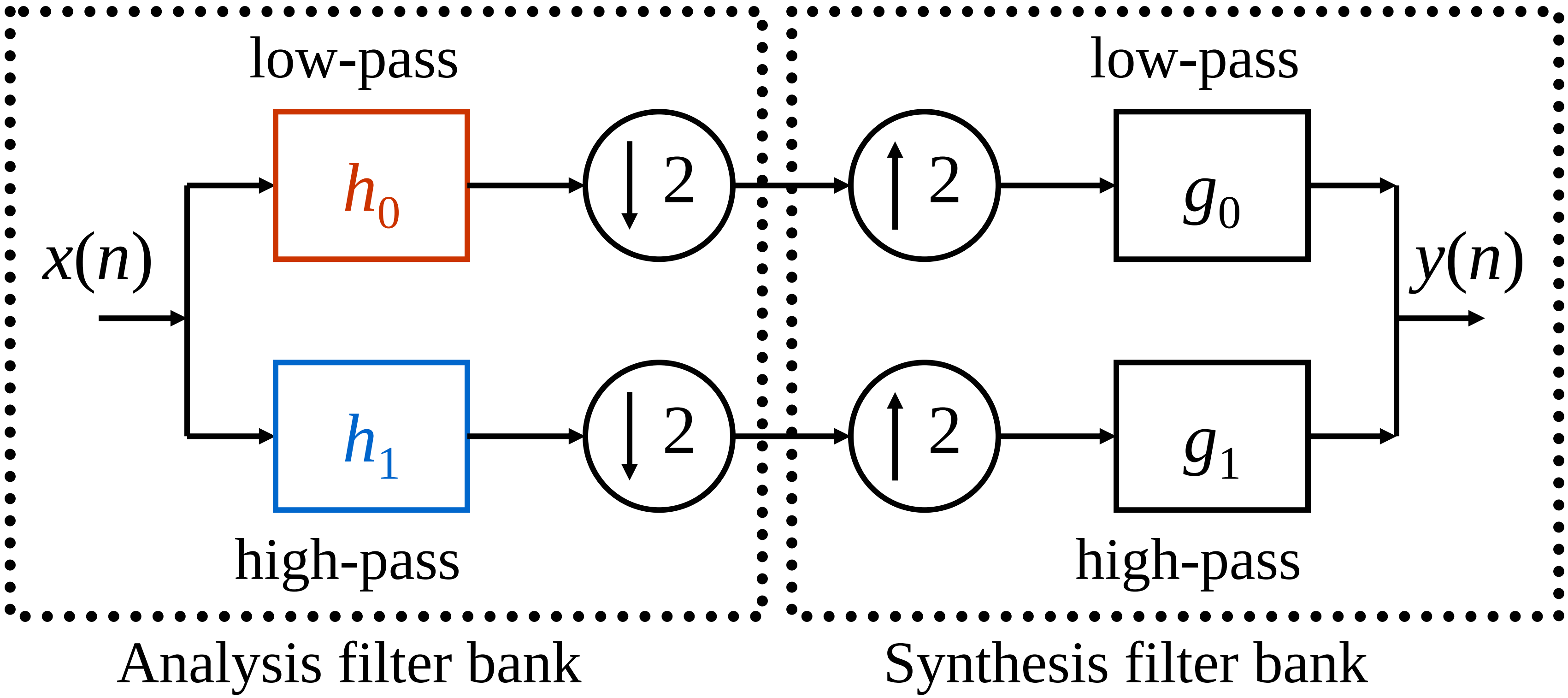
$y(u,v) =$

327.5	-215.8	16.1	-10.7	-3.7	-1.5	4.2	-6.7
18.1	3.4	-9.9	3.7	0.5	-3.2	3.5	2.2
2.5	1.3	-5.4	2.8	-1.0	2.3	-1.6	-2.6
0.6	-2.5	3.0	5.0	1.8	2.2	-2.6	-1.4
0.3	1.6	3.4	0.0	2.5	-5.1	1.6	-0.7
-0.6	-1.8	-2.4	0.5	-0.4	-1.6	-0.1	2.1
0.9	1.6	-0.6	-0.7	2.1	-0.5	0.9	2.8
0.6	-1.0	-2.9	-1.4	0.2	1.9	-0.6	0.7

1-D Discrete Wavelet Transform (DWT)

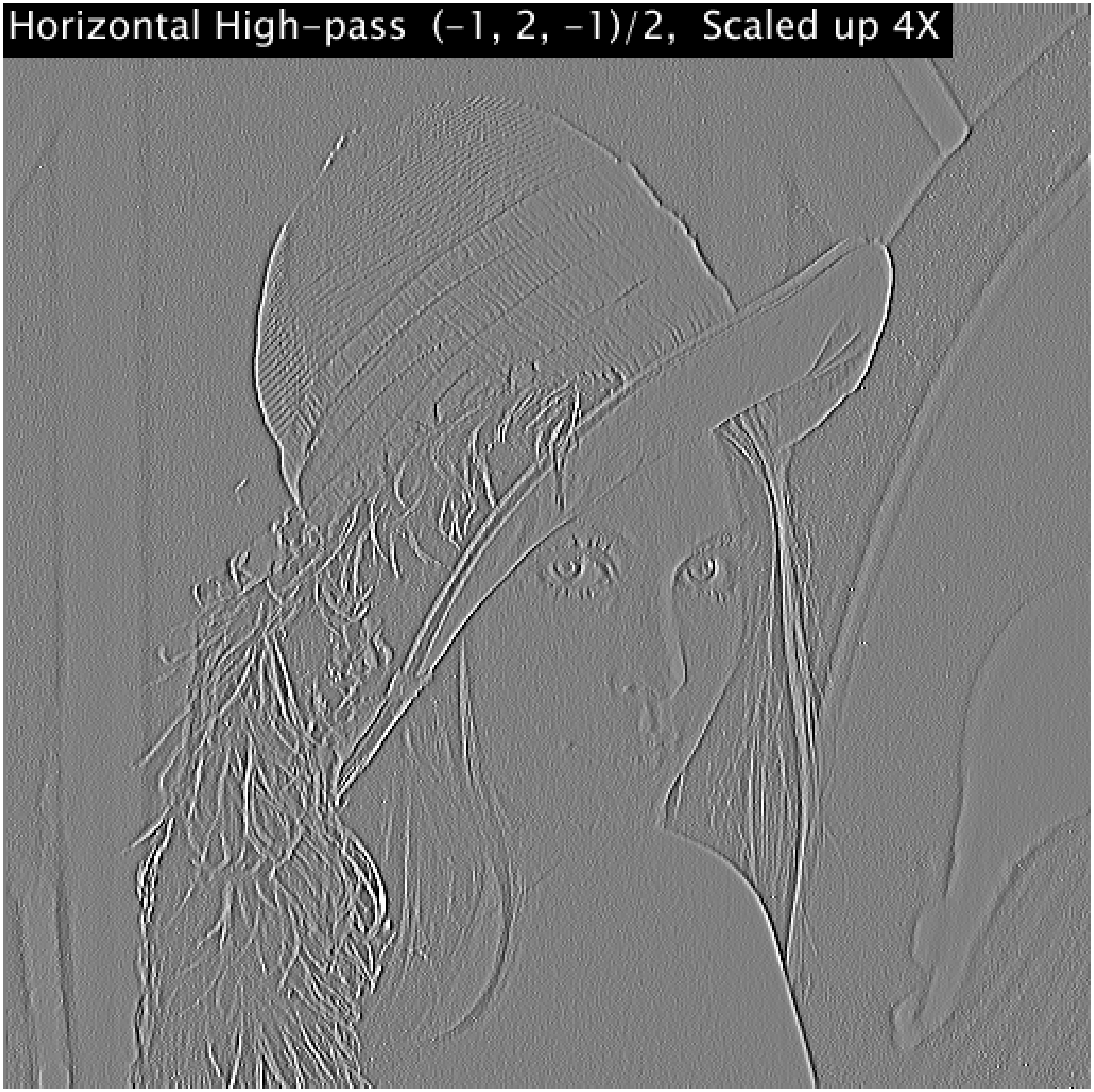
- The **forward discrete wavelet transform** (DWT) decomposes a one-dimensional (1-D) sequence (e.g., line of an image) into two sequences (called **subbands**), each with half the number of samples, according to the following procedure:
  - The 1-D sequence is separately **low-pass** and **high-pass** filtered.
  - The filtered signals are downsampled by a factor of two to form the low-pass and high-pass subbands.
  - The two filters are called the **analysis filter bank**.

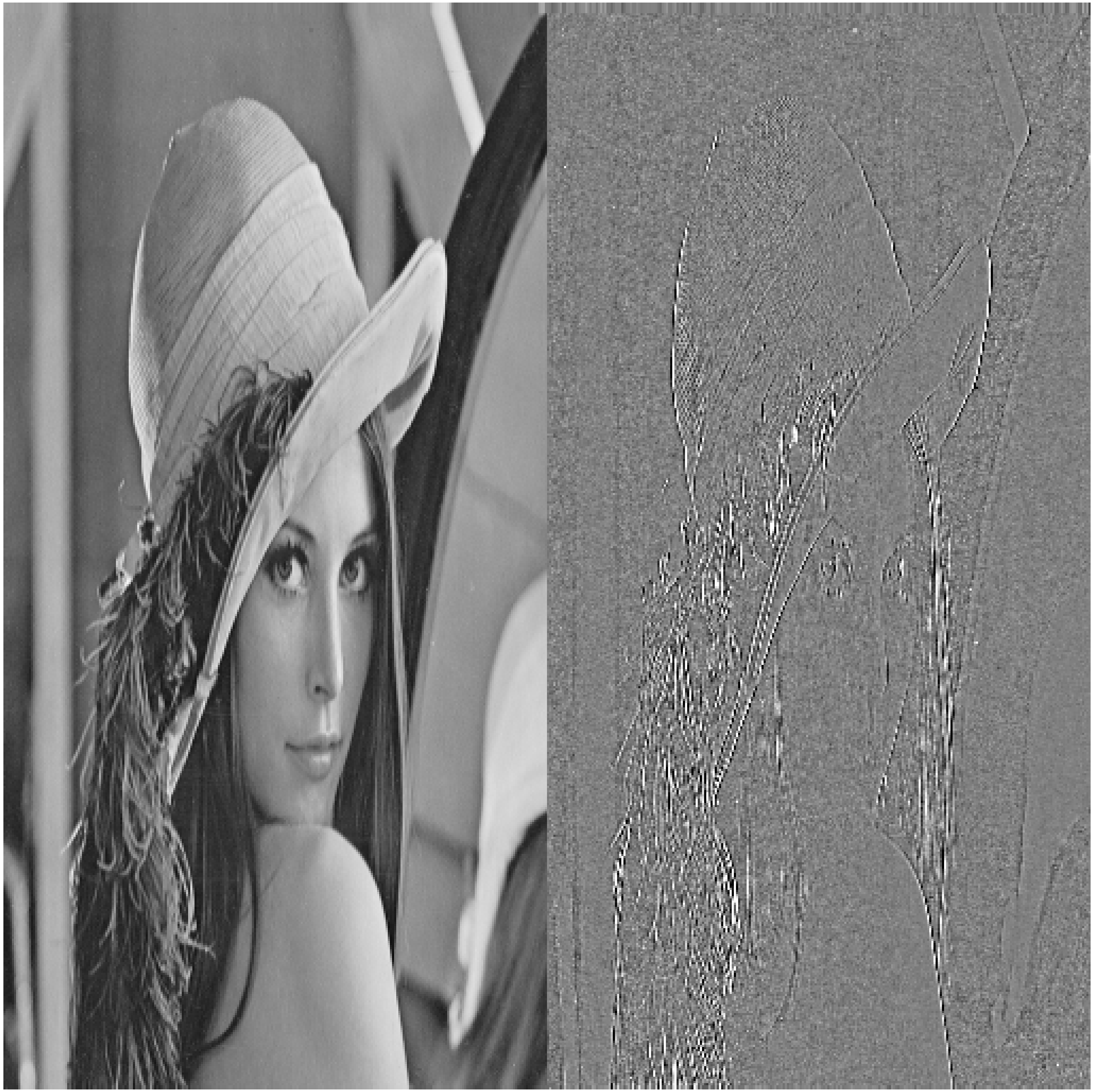
The 1-D Two-Band DWT



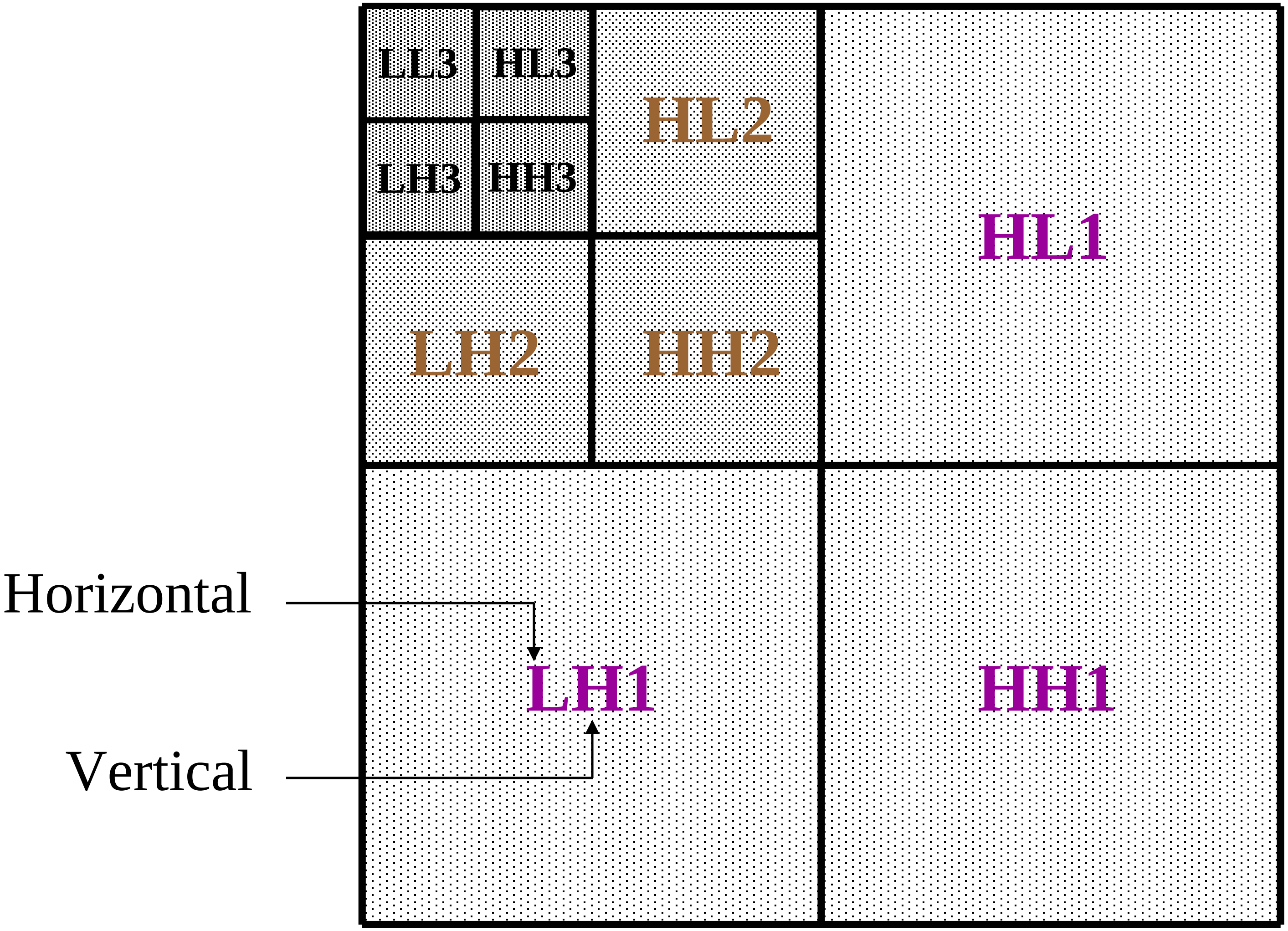
Ideally, it is desired to choose the analysis filter banks (h_0 and h_1), and the synthesis filter banks (g_0 and g_1), in such a way so as to make the overall distortion zero, i.e., $x(n) = y(n)$. This is called the **perfect reconstruction** property.


Horizontal Low-pass (-1, 2, 6, 2, -1)/8


Horizontal High-pass (-1, 2, -1)/2, Scaled up 4X



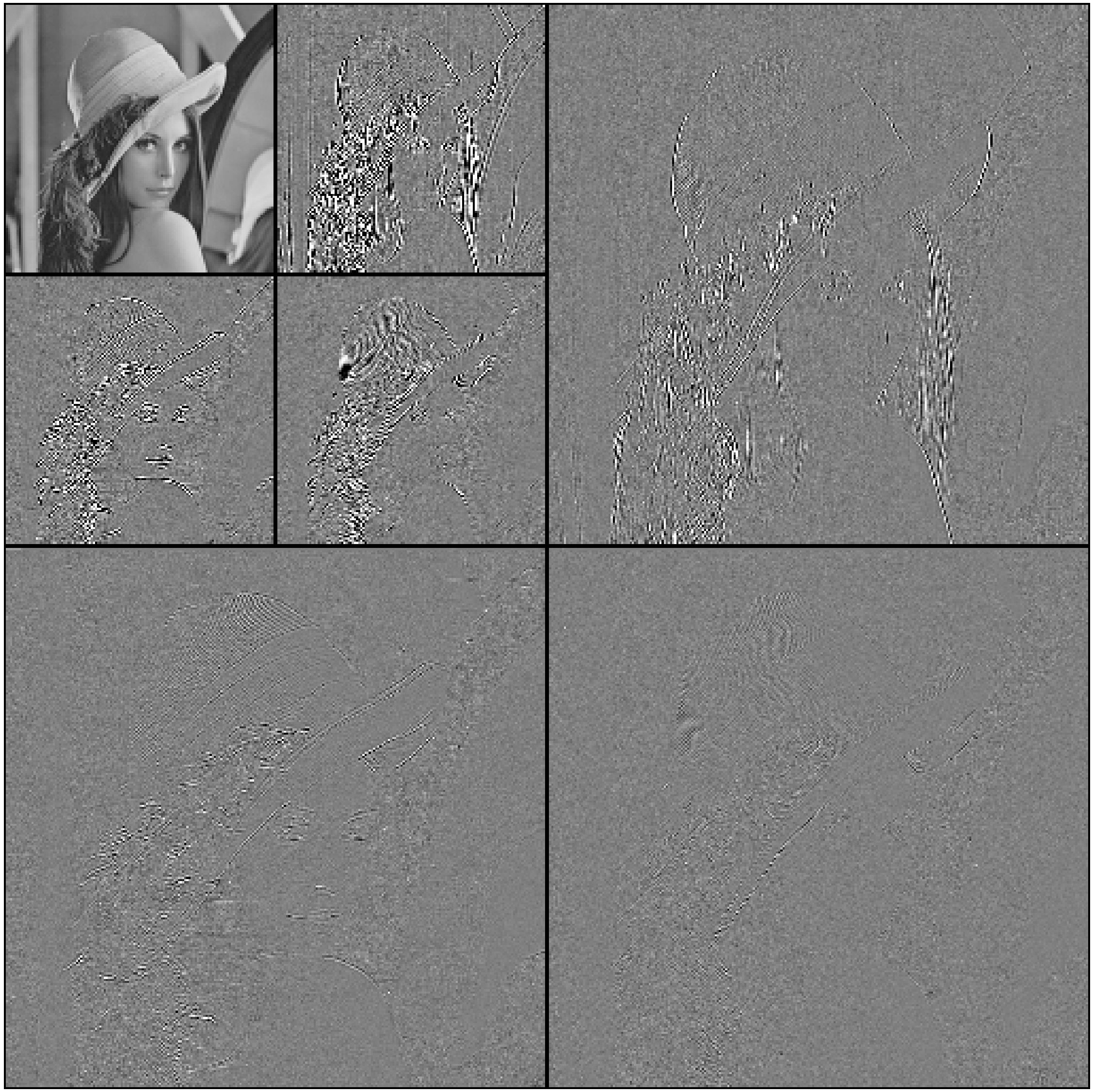
2-D Wavelet Decomposition



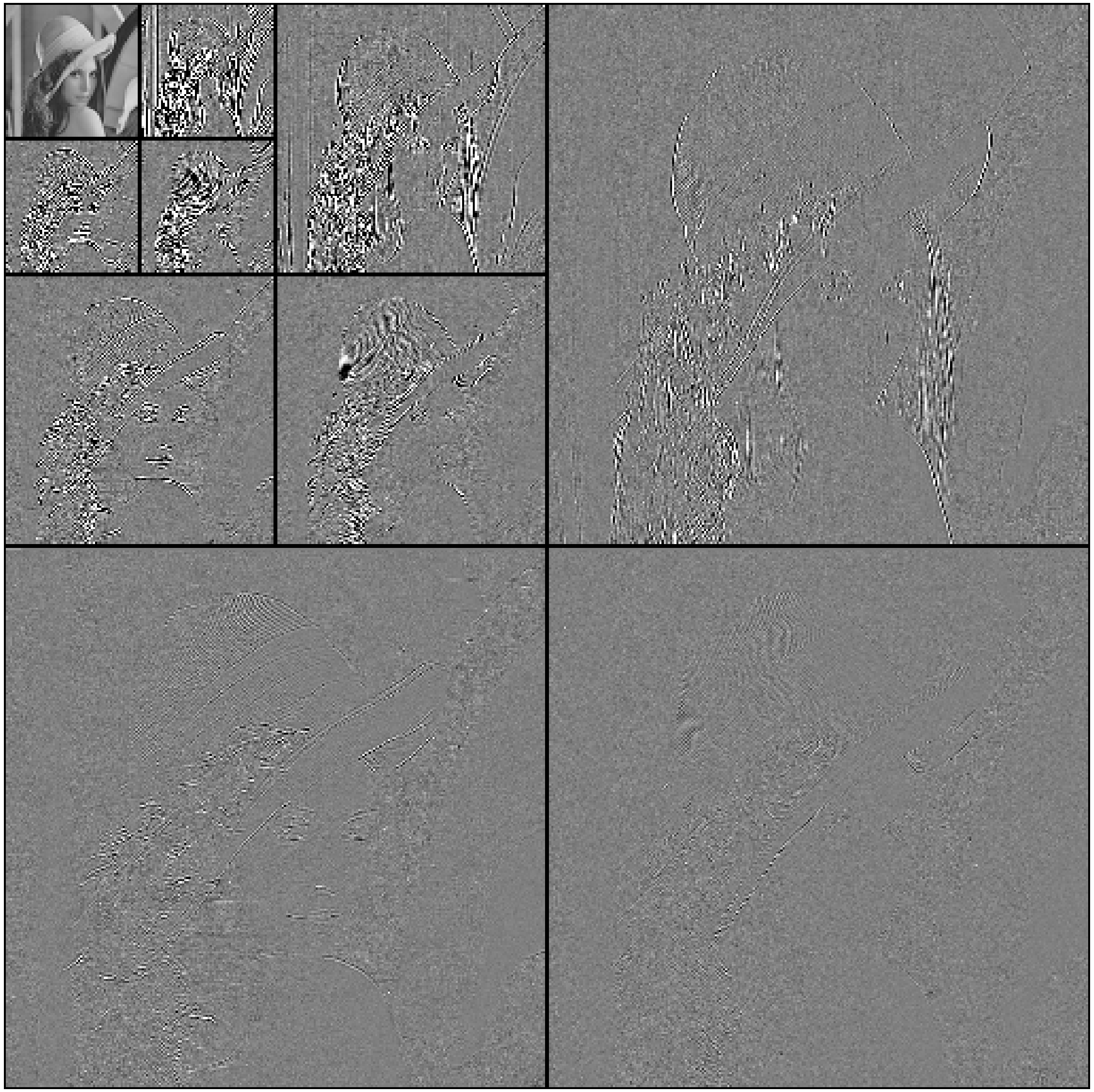
Original Lena Image

1-Level, 2-D Wavelet Decomposition of Lena

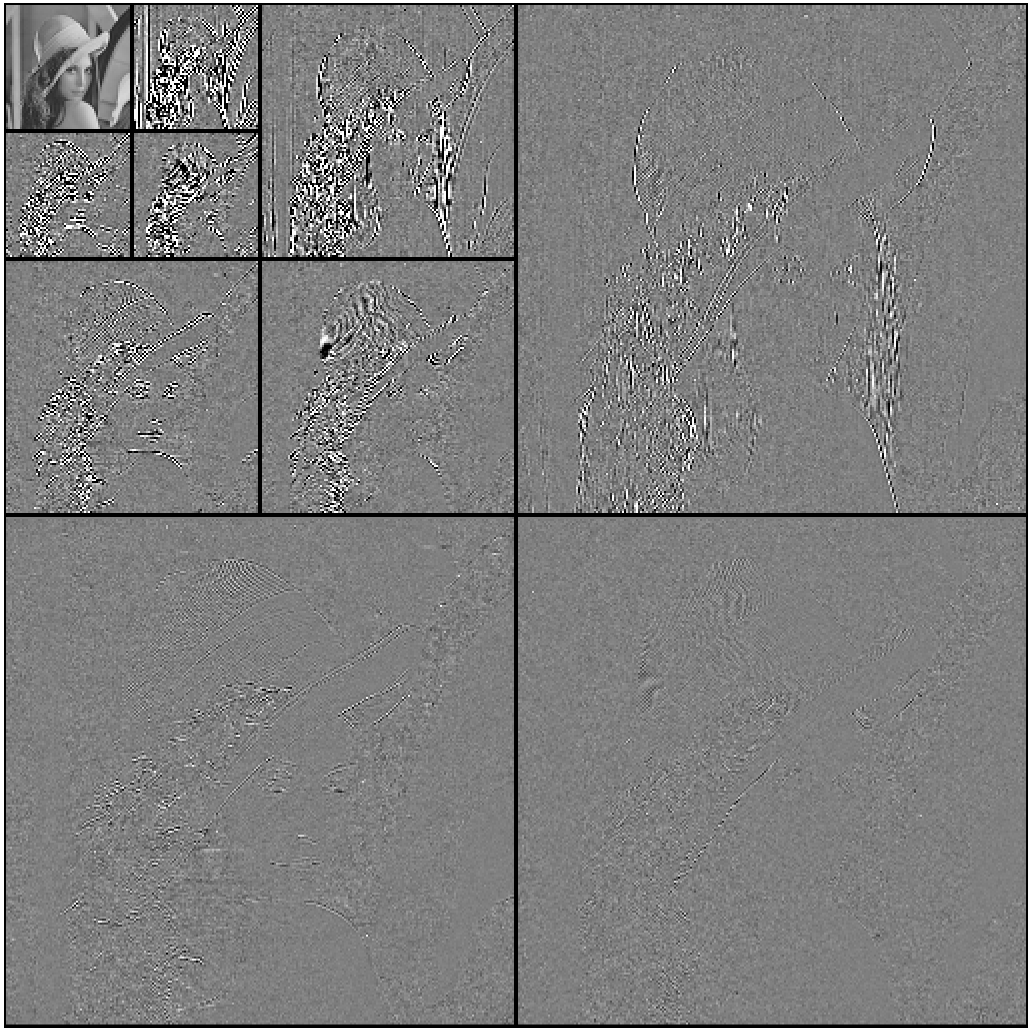
2-Level, 2-D Wavelet Decomposition of Lena



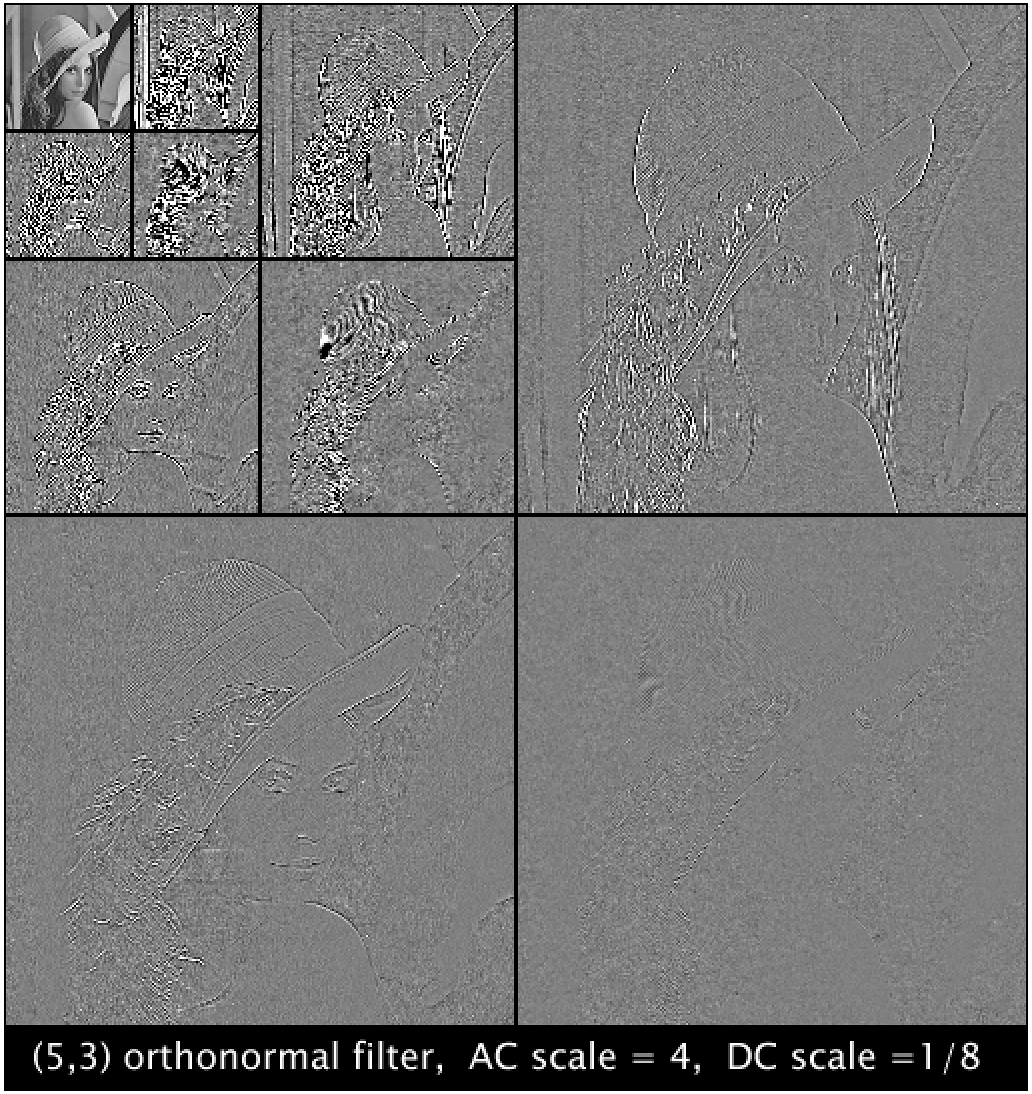
3-Level, 2-D Wavelet Decomposition of Lena



3-Level, 2-D DWT with (9,7) Filter


(9,7) orthonormal filter, AC scale = 4, DC scale =1/8

3-Level, 2-D DWT with (5,3) Filter

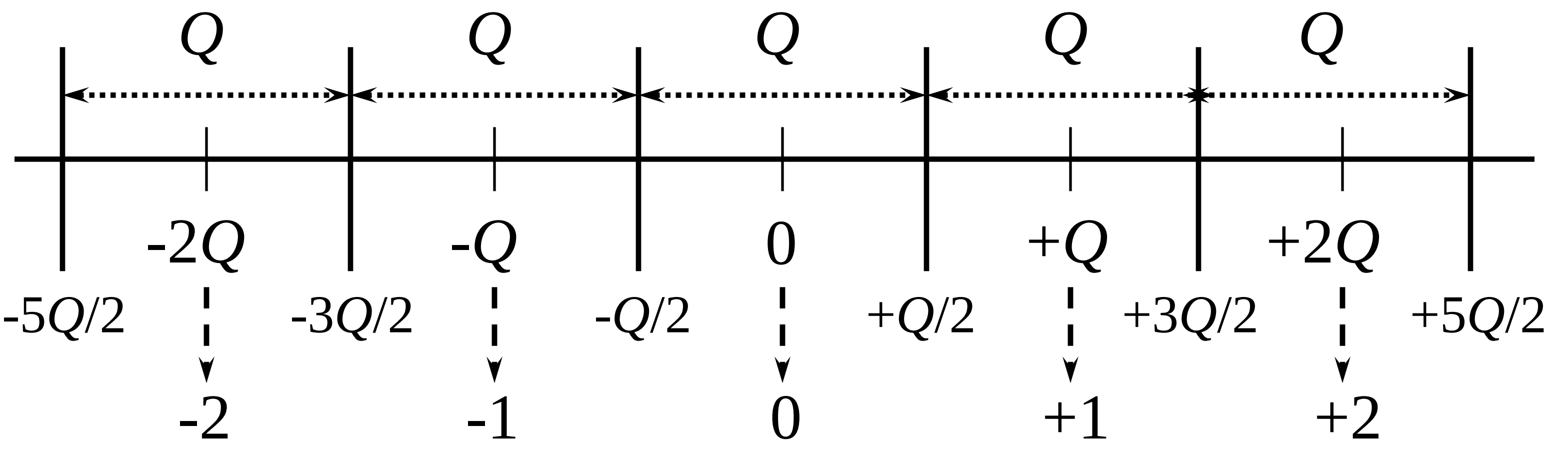


Quantization

Quantization

- A many-to-one mapping that reduces the number of possible signal values at the cost of introducing errors.
- The simplest form of quantization (also used in all the compression standards) is **scalar quantization** (SQ), where each signal value is individually quantized.
- The joint quantization of a block of signal values is called **vector quantization** (VQ). It has been theoretically shown that the performance of VQ can get arbitrarily close to the rate-distortion (R-D) bound by increasing the block size.
- In lossy compression schemes, quantization acts as a control knob for trading off image quality for bit rate (compression ratio).

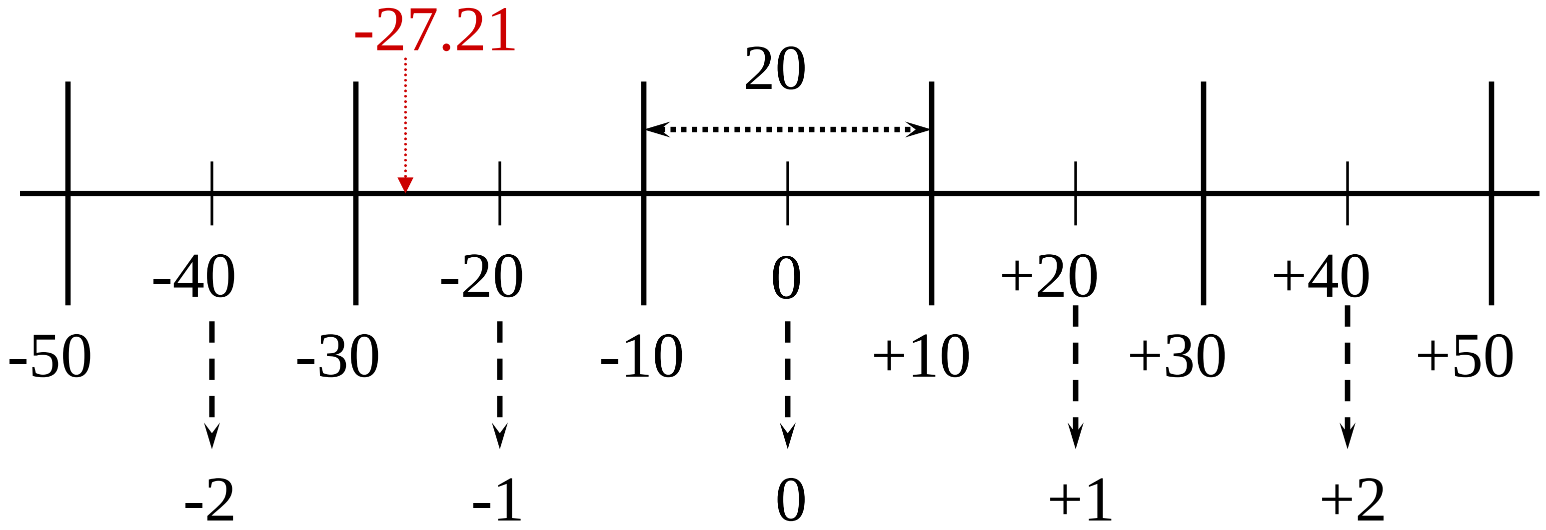
Uniform Threshold Quantizer (UTQ)



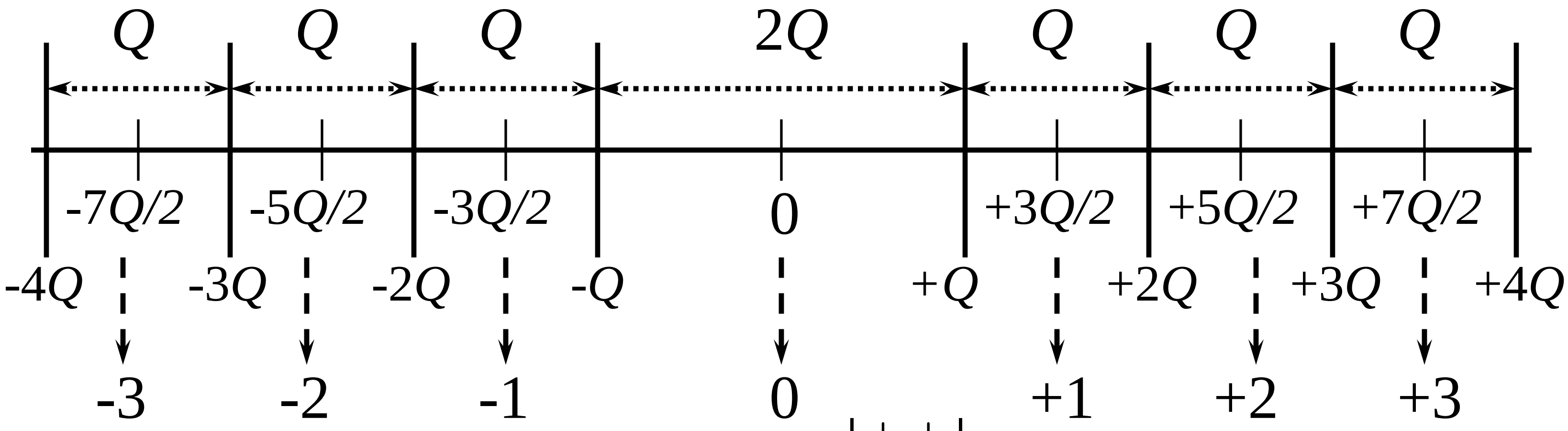
- In a **UTQ** quantizer, all bins have the same size. The bin size *Q* is called the quantizer **step size**. The quantization dequantization rule for a midpoint reconstruction is given by:
  - Quantization rule: $q = \mathrm{NINT}[y/Q]$
  - Dequantization rule: $z = q * Q$
  - Where *y* is the input signal, *q* is the resulting quantizer index, *z* is the **reconstructed** (quantized) value, and the NINT operation denotes rounding to the nearest integer.

Example: UTQ

- Quantization: encoder input value = -27.21
  - Scale by the step size → (-27.21)/(20) = -1.3605
  - Round to the nearest integer to get quantizer index = -1
- Dequantization: decoder received index = -1, step size = 20
  - Multiply quantizer index by step size → -1 x 20 = -20
  - Error = -27.21 – (-20) = -7.21



Uniform Threshold Quantizer with Deadzone



- Quantization rule: $z = \text{sign}\left\lfloor \frac{|y|}{Q} \right\rfloor$
- Dequantization rule: $z = (q + r * \text{sign}(q)) * Q$

where y is the input signal, q is the quantizer index, z is the reconstructed signal value, sign(x) is sign of x, $\lfloor x \rfloor$ denotes the largest integer smaller than x, and r is the reconstruction bias (r = 0.5 corresponds to midpoint reconstruction).

Symbol Modeling And Encoding

Symbol Modeling and Encoding

- Symbol modeling and encoding involves the process of defining a statistical model for the symbols to be encoded (e.g., quantizer output levels or indices) and assigning a binary codeword to each possible output symbol based on its statistics.
- The resulting code should be **uniquely decodable**, i.e., each string of input symbols should be mapped into a unique string of output binary symbols.
- Examples are fixed-length coding, **Huffman** coding, **Golomb-Rice** coding, **arithmetic** coding, **Lempel-Ziv-Welch** (LZW) coding.

Huffman Codes

Pixel Value	Probability	Code 1 Fixed	Code 2 Huffman
0	0.60	00	0
1	0.30	01	10
2	0.05	10	110
3	0.05	11	111

Example

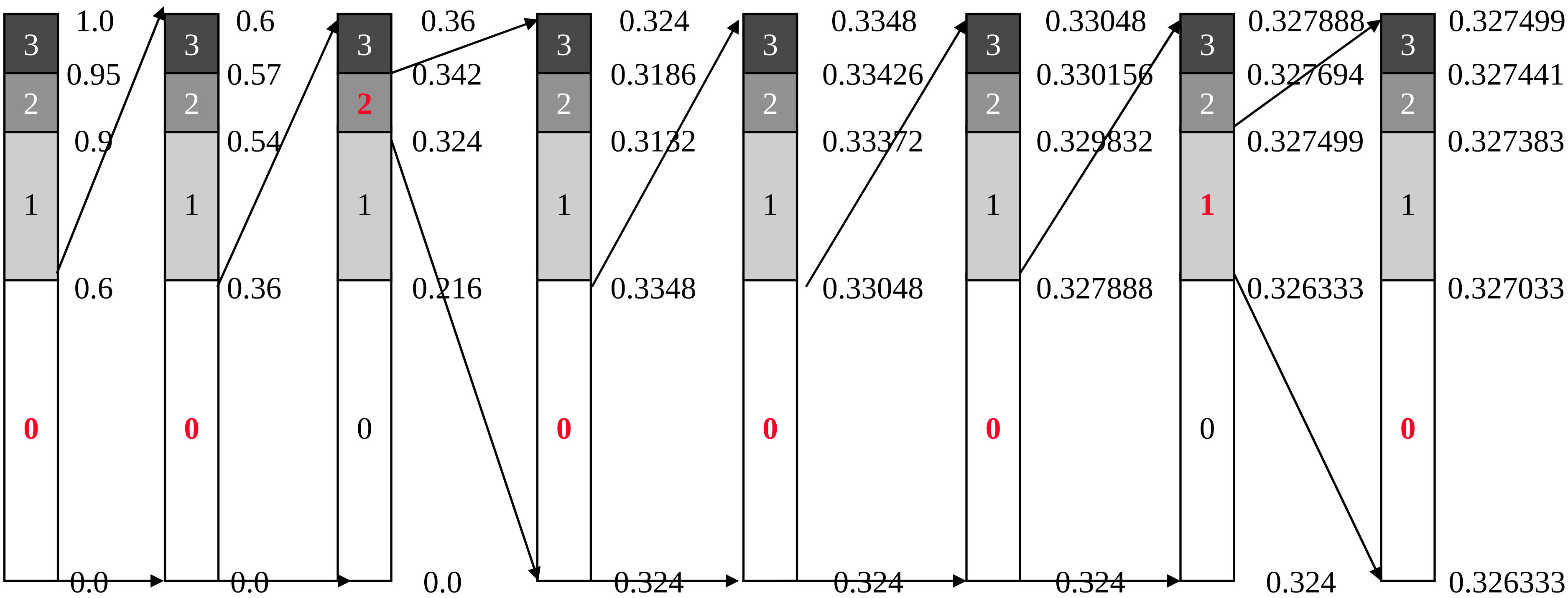
Line 1	0	0	3	0	0	0	1	0	1	1
Code 1	00	00	11	00	00	00	01	00	01	01
Code 2	0	0	111	0	0	0	10	0	10	10
Line 2	0	0	2	0	0	0	1	0	1	1
Code 1	00	00	10	00	00	00	01	00	01	01
Code 2	0	0	110	0	0	0	10	0	10	10

- Average length of Code 1 = 2.0 bits/symbol.
- Average length of Code 2 = 1.5 bits/symbol.
- Code 2 is a prefix code, i.e., no codeword is a prefix of any other codeword (uniquely decodable)
- A Huffman code has an average length that is less than, or equal to, the average length of all other uniquely decodable codes for the same source and code alphabet.

Arithmetic Encoder Example

- Example m-ary (base 10, want base 2 and binary)
- Decoder stops with special symbol or by counting

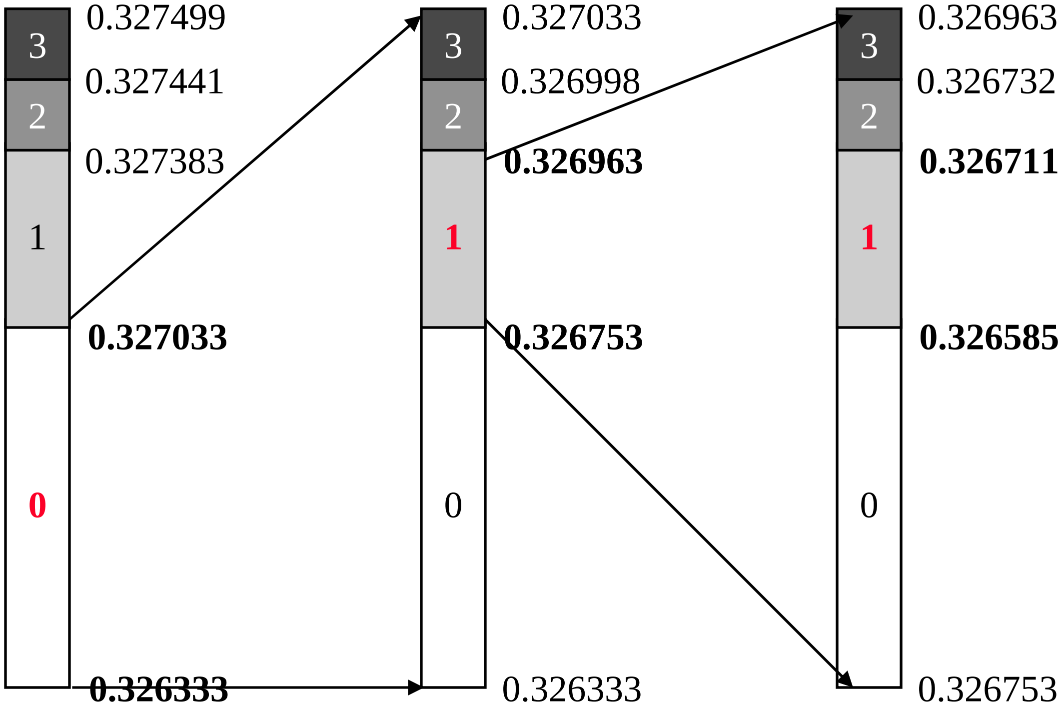
Sequence > 0020001011 (same as Huffman Example)



Arithmetic Encoder Example

Resulting Value is selected between
0.326711 and 0.326585
Could Select 0.3266

Sequence > 0020001011

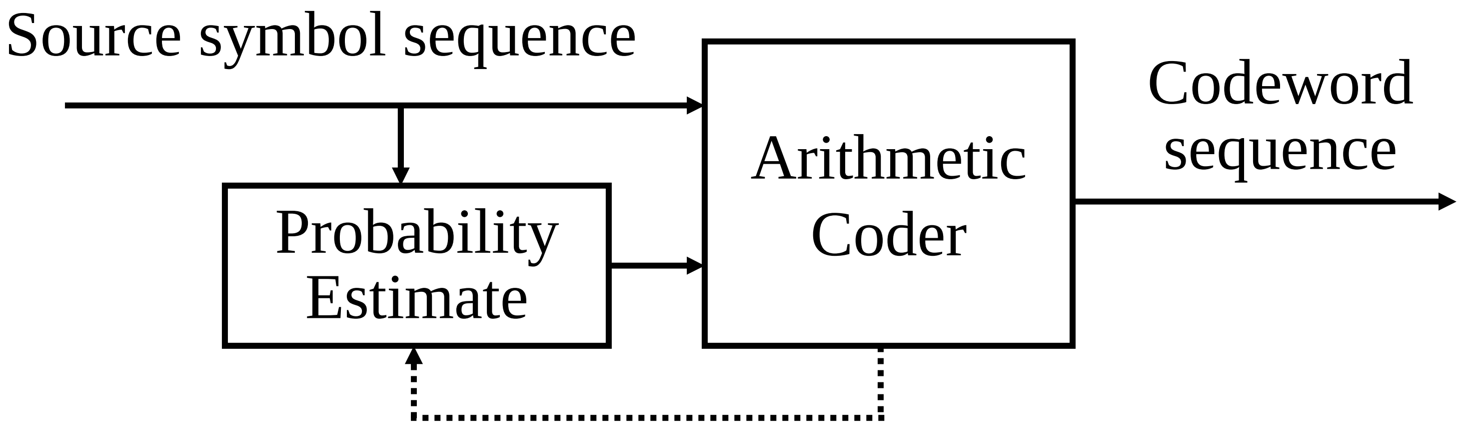


Arithmetic Encoder Example

- Line 2 (Sequence 0030001011)
- This results in the selection of data between 0.344942 and 0.344879 so we could select 0.3449
- So this results in the value of 0.3226 and 0.3449 to represent those data lines.
- For a different example (same probability)
- Line 1 > 0021111311 Line 2 > 0000000000 would result in values of 0.339448452 and 0.339447796 (0.339448) for line 1 and between 0 and 0.006046618 (select 0).

Arithmetic Coding (AC)

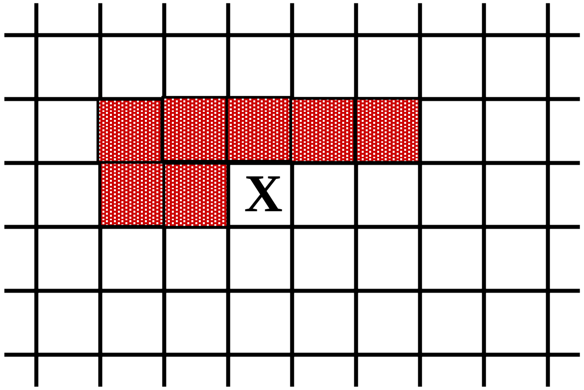
- An arithmetic coder accepts at its input the symbols in a source sequence along with their corresponding probability estimates, and produces at its output a code stream with a length equal to the combined ideal codelengths of the input symbols.
- Some implementations of arithmetic coding adaptively update the symbol probability estimate in each context as the symbols get encoded.
- Practical implementations of AC, such as the JBIG/JPEG **QM-Coder** or **MQ-Coder**



Conditioning Contexts

- In general, the probability of a sample having a certain value is influenced by the value of its neighbors. Thus, the symbol probabilities can be conditioned on the values of the symbols in a neighborhood surrounding them. For a given neighborhood configuration, each combination of the neighboring samples denotes a **conditioning context**.
- The **conditional entropy** of a correlated source can be significantly less than its zeroth-order entropy.

0	0	0	0	0	0	0	0
0	0	0	0	0	1	0	0
0	0	0	0	1	1	0	0
0	0	0	0	1	1	0	0
0	0	0	0	0	0	0	0



Example: Entropy of Lena MSB

Conditioning contexts can capture the redundancy in the image:

No conditioning contexts
Entropy = **1.0** bit/pixel

7-neighbor conditioning context
Entropy = **0.14** bits/pixel

Most significant bit plane

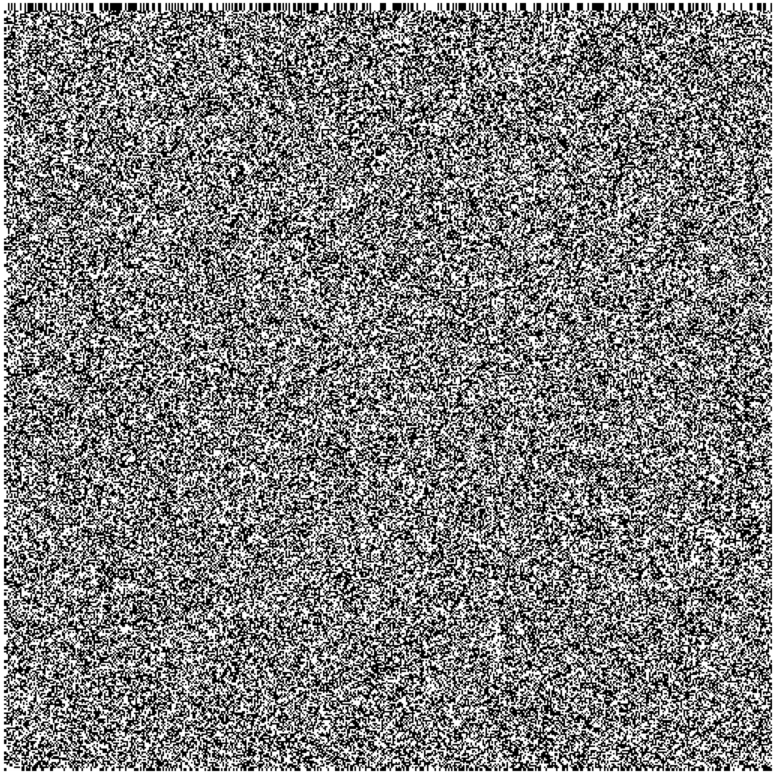
Example: Entropy of Lena LSB

No conditioning contexts

Entropy = **1.0** bit/pixel

7-neighbor conditioning context

Entropy = **1.0** bits/pixel



Least significant bit plane

Rate Controller

- A rate controller is used when an exact compression rate or image throughput is desired (e.g., DDS 1.3 DCT).
- The rate controller changes the amount of quantization dependent on the output bit rate and the desired bit rate.
  - The quantization is greater (i.e., bin size gets larger) when too many bits are coming out of the symbol encoder.
  - The quantization is reduced when too few bits are coming out of the symbol encoder.
- The rate control can be performed single-pass (the quantization step size changes as a function of location in the image) or multiple-pass (quantization step size is usually consistent throughout the image, tile or block).

Color and Multiple Component Transform

Color Image Representation

- Color image components are highly correlated due to:
  - Overlapping spectral responses of the sensors
  - Smooth spectral distribution of surfaces and illuminants
- The RGB color values are often transformed into a new set of values called **luminance** and **chrominance** (such as YCrCb, or YIQ), such that:
  - The transformed components are less correlated (reduced redundancy), and,
  - The sensitivity variations of the human visual system (irrelevancy) can be taken into account, e.g., the chromatic components may be subsampled or compressed more aggressively.

YC_bC_r Color Space

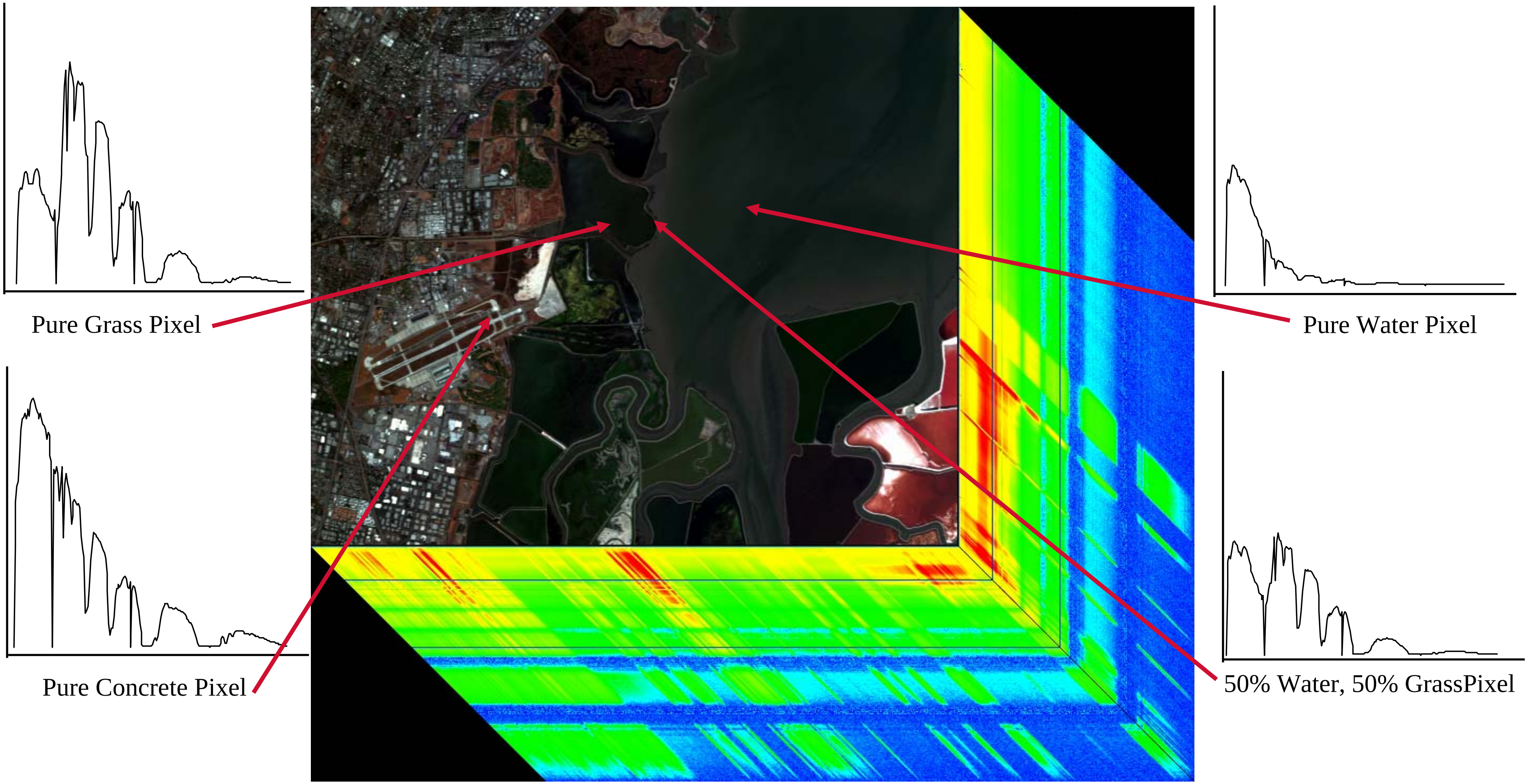
This is the most commonly used color coordinate system for the representation of image and video signals:

$$Y = 0.299(R-G) + G + 0.114(B-G)$$
$$C_b = 0.564(B-Y) \quad \text{and} \quad C_r = 0.713(R-Y)$$

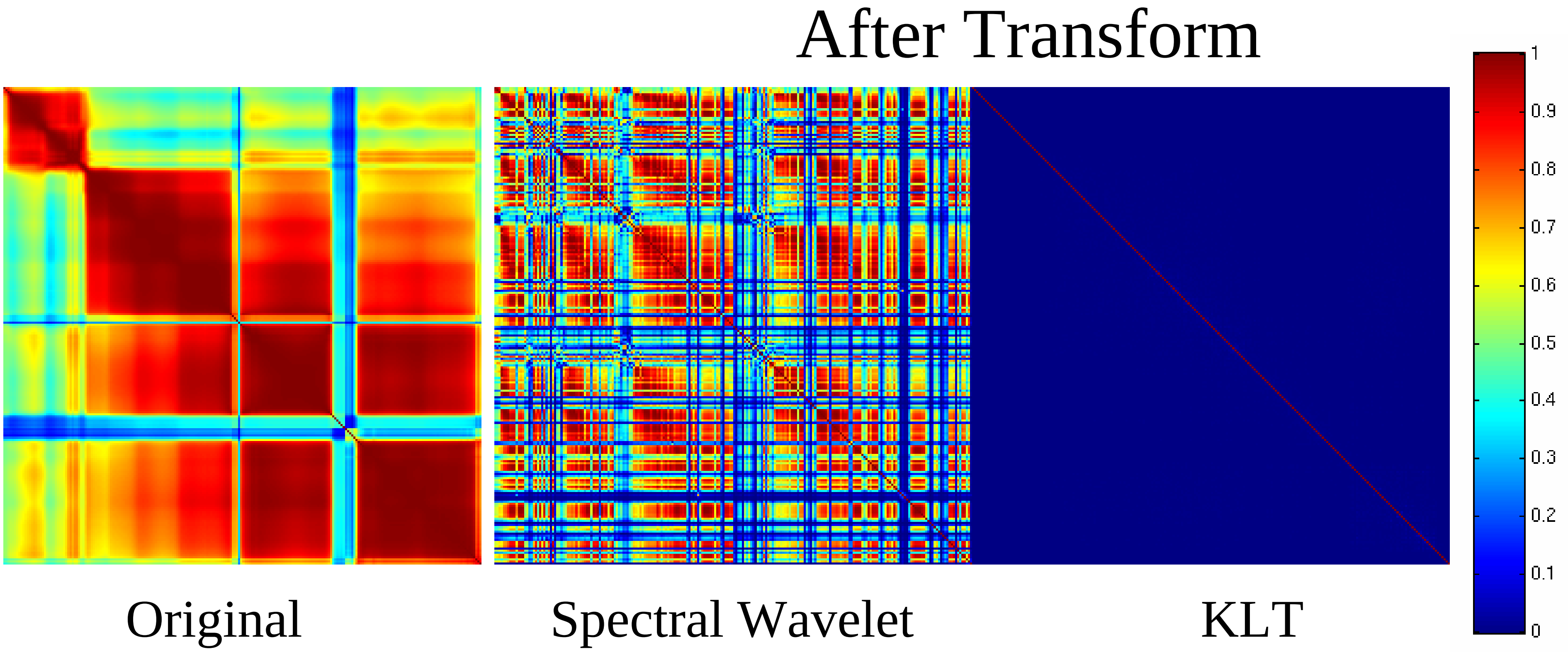
$$\begin{bmatrix} Y \\ C_b \\ C_r \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.500 \\ 0.500 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

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

Hyperspectral Information Cube (AVIRIS)



Spectral Correlation (AVIRIS)



How To Choose a Compression Algorithm

Standards

Requirements

Choosing a Compression Algorithm

- Q: What is the best compression technique?
- A: It depends on the application!
- Some factors to consider:
  - Image quality (lossless, visually lossless, visually lossy, acceptable loss)
  - Operational bit rate (transmission rate vs. image size/number requirements)
    - Constant bit rate(per block) vs. fixed bit rate(per pixel) vs. constant quantization
  - Computational complexity
  - Channel error tolerance
  - Encoder/decoder asymmetry
  - Artifacts (blocking, noise, edge blur)
  - System compatibility and compression standards
  - Input image characteristics
    - Data type and previous processing (sharpening, compression)
  - Output image applications
  - Spatial Accuracy

Digital Image Compression Standards

- Facilitate the exchange of compressed image data between various devices, applications and users.
- Permit common hardware/software to be used for a wide range of products, thus lowering costs and shortening development time.
- Several levels of standards:
  - Specification used in limited-access world
    - 1.3 DCT, 2.3 DCT, 4.3 DPCM
  - Military Standard used in DoD community
    - MIL-STD-188-198A NITFS JPEG DCT, NITFS Vector Quantization
  - International standards used in the commercial world
    - ISO/IEC 10918-1 (JPEG)
      - Very broad tool box; not all JPEG algorithms are the same

Commercial Image Compression Standards

- Binary (bi-level) images:
  - Group 3 & 4 facsimile (‘80); JBIG (‘94); JBIG2 (‘01)
- Continuous-tone still images:
  - JPEG (‘92); JPEG2000 (Part 1 IS ‘00, Part 2 IS ’01, other parts ongoing) – www.jpeg.org
- Image sequences (moving pictures):
  - H.261 (‘90); H.263 (‘95); H.263+ (‘97); H.26L (merged with MPEG4 AVC)
  - MPEG1 (‘94); MPEG2 (‘95); MPEG4 (‘99, AVC still ongoing) - http://mpeg.telecomitalialab.com/

Standards Background

- 4.3 DPCM
  - Developed for visually lossless, rate-controlled simple compression for storage and transmission
  - Old technology, current technology can significantly outperform
- 1.3 DCT/2.3 DCT
  - Significant development effort to produce a high quality (0.2/0.1 NIIRS loss) at low bit rates (1.3 BPP/2.3 BPP).
  - Old technology, still very competitive but not very flexible
- JPEG DCT/NITFS JPEG DCT
  - Developed as a commercial standard to run on commercial PCs (386s) and commercially viable hardware.
  - NITFS/DoD adopted because of quality, flexibility and COTS products
- Vector Quantization
  - Developed to compress maps with very fast decompression.
  - Used by NIMA to put maps and imagery out on CD.
- NIMA Method 4
  - Developed to achieve dissemination to warfighters with very low bandwidth communication lines

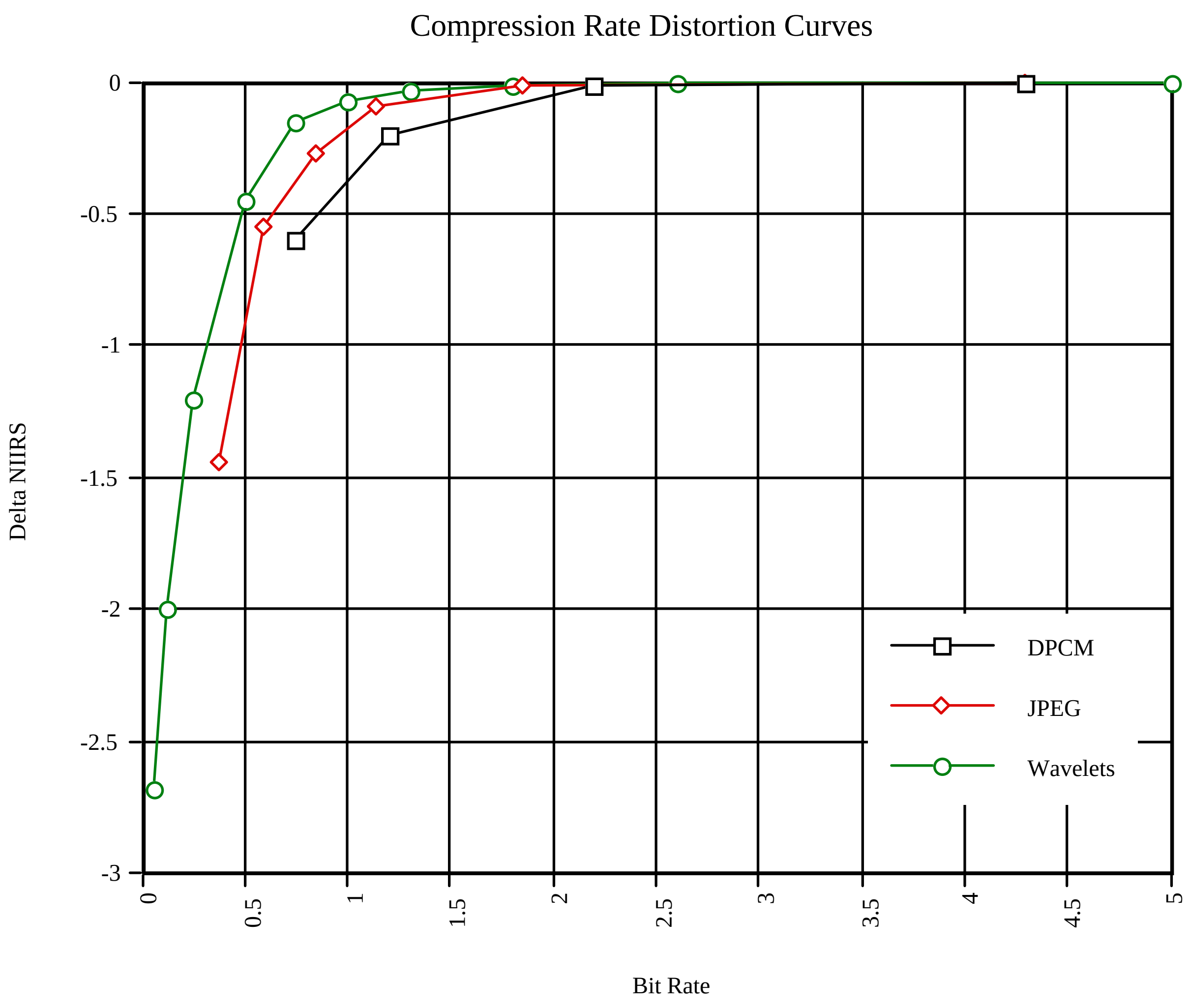
Current Requirements

- 4.3 DPCM
  - 0.0 NIIRS loss, 2:1 or better compression, fast decompression
- 2.3 DCT
  - 0.1 NIIRS loss, 3:1 or better compression, spatial accuracy
- 1.3 DCT
  - 0.2 NIIRS loss or less, 1.3 bpp or less, robust to channel errors
- NITFS/NIMA VQ
  - Fast decompression, variable compression, robust to channel errors
- NITFS JPEG DCT
  - 0.5 NIIRS loss at 8:1 compression, 2.0 min. decompression time Variable compression ratios, robust to channel errors
- NIMA Method 4
  - High compression ratios with minimal image quality loss

Compression Optimization

- Each compression algorithm has several parameters that can be modified to improve the quality, increase the compression ratio (at same quality) or reduce artifacts.
  - For example, JPEG optimization can give a 5% to 15% gain in compression with proper optimization of the quantization and Huffman tables or a 0.5 NIIRS improvement at the same compression rate
  - Parameters are optimized for the characteristics of the image and/or the requirements of the compression applications
    - Optimization is common for a class of imagery or image characteristics
      - Color, panchromatic, IR, SAR, noisy, graphic
    - Optimization is also common for a desired bit rate (1.3 bpp, 2.3 bpp)
      - Quantization tables, Huffman tables
  - Parameters can be modified to reduce identified artifacts which may be the interaction between the compression algorithm, the image characteristics, post processing and the display process.

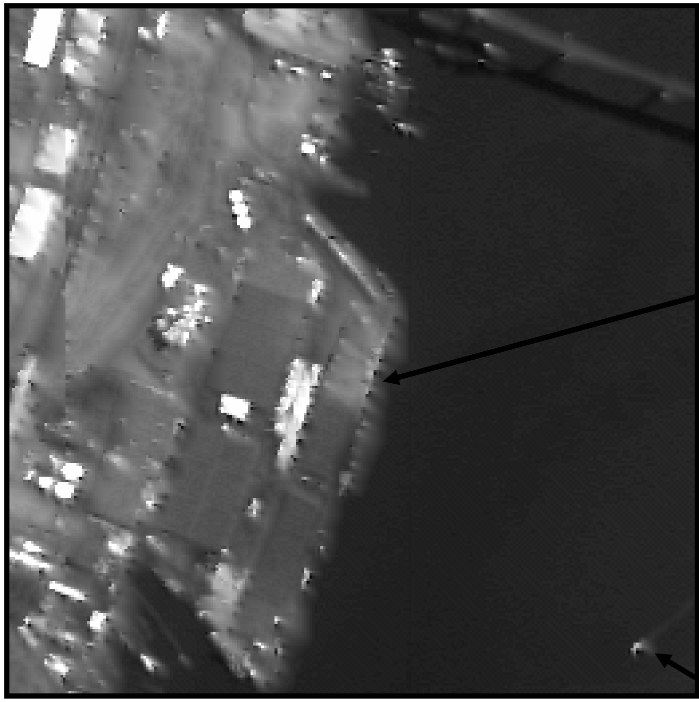
Compression Rate Distortion

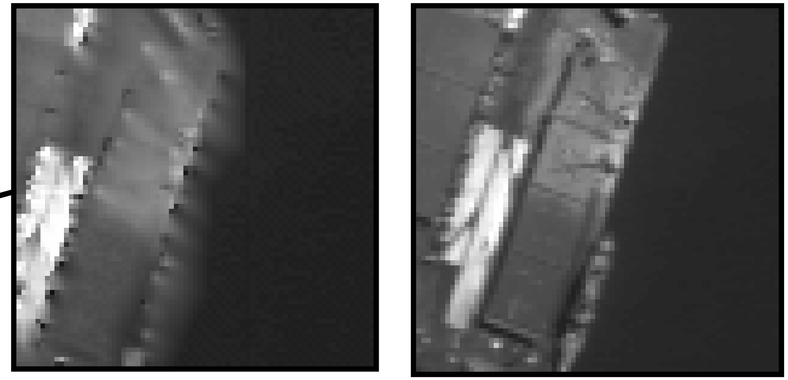


Compression Artifacts

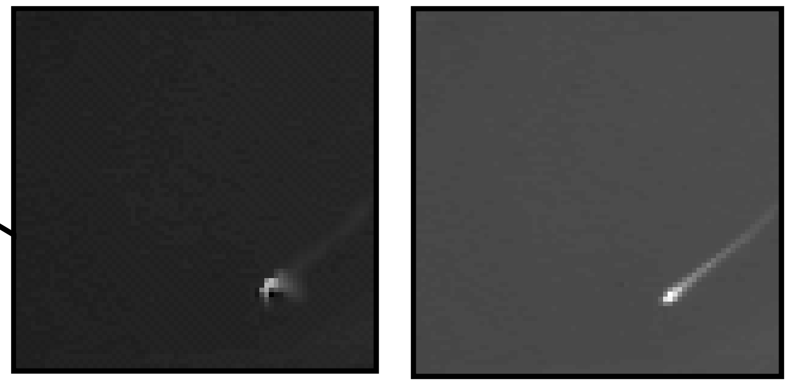
- Artifacts of compression are viewable when:
  - The compression ratio is pushed beyond the normal working environment of the given compression algorithm, or
  - the image is processed beyond the “normal” range of enhancements (i.e., sharpen, sharpen-more, DRA, TTC)
- Common artifacts include;
  - DPCM
    - Slope overload, water-fall artifact
  - DCT
    - Blocking, ringing around edges, DCT basis functions
  - Wavelets
    - False texture, reduction in resolution, ringing
  - VQ
    - Blocking, contouring

DPCM Example (1.8 bpp)





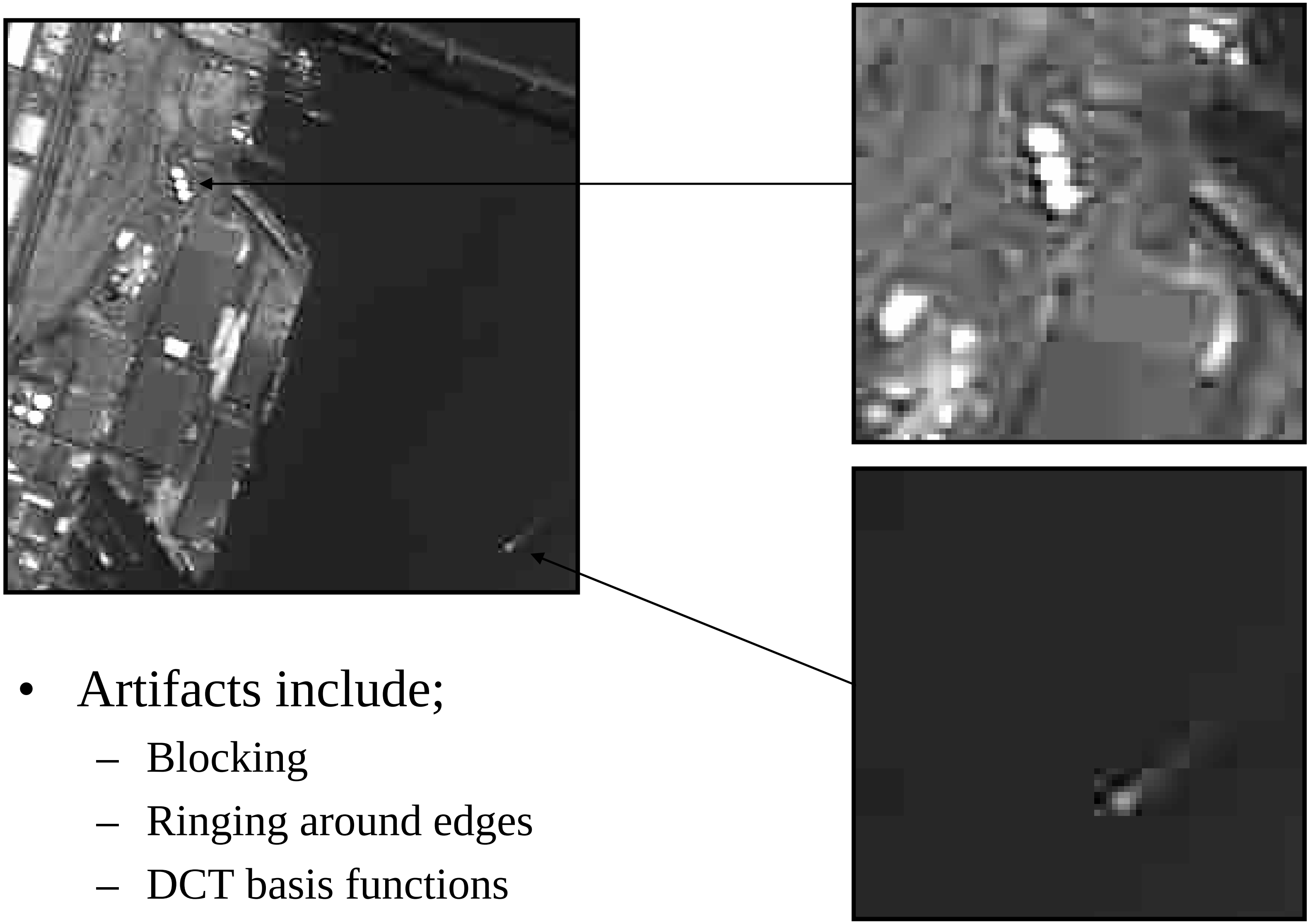
DPCM Original



DPCM Original

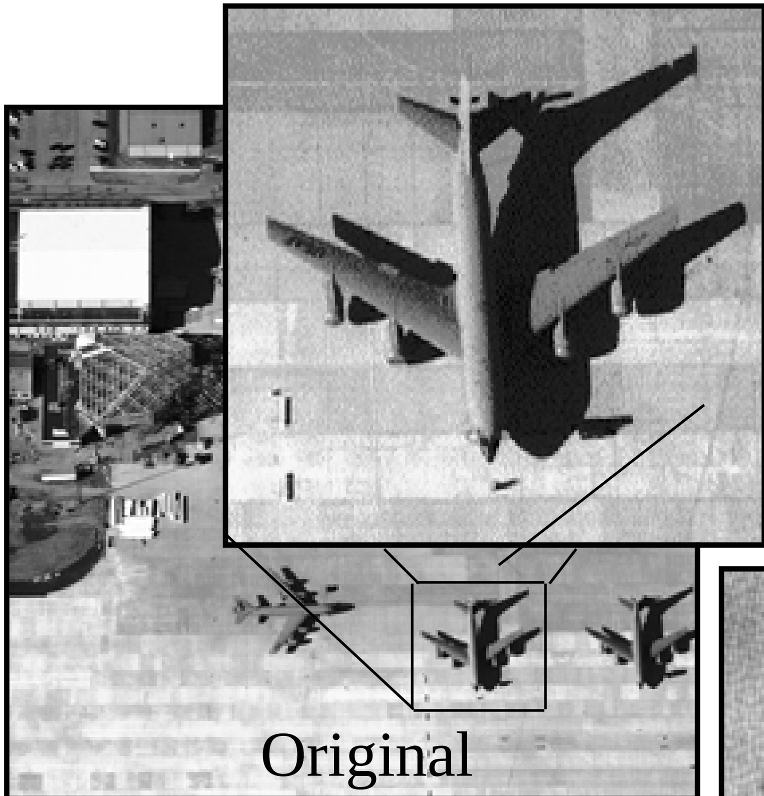
- Artifacts include;
  - Slope overload
  - Water-fall artifact

DCT Example (JPEG @ 0.4 bpp)



- Artifacts include;
  - Blocking
  - Ringing around edges
  - DCT basis functions

Wavelet Example (0.0625)




Algorithm 1


Algorithm 2


Algorithm 3

- Artifacts include;
  - False texture
  - Reduction in resolution
  - Ringing

Channel Errors

- Problems from channel errors are hard to characterize for each algorithm
- Several factors affect the image quality when a channel error is occurred
  - Variable length encoder vs. fixed length encoders
    - A channel error in a variable length encoder will propagate until the encoder resyncs or there is a restart interval
    - A channel error in a fixed length encoder only affects that value
  - Prediction/transform technique
    - Any incorrect value is propagated to surrounding value depending on the prediction or transform technique
      - Only the block of a given DCT is affected by an error in the AC components
      - Error is propagated from the error pixel to the lower and right for a DPCM
      - Depending on the level of the wavelet the error is propagated to the surrounding 2N by 2N pixels (N is the level to error occurred)

Overcoming Channel Errors

- Protection to channel errors
  - Restart markers
    - Restart markers are used to restart the algorithm to stop the propagation of any error that may have occurred before
  - Error Dection And Correction (EDAC)
    - Forward Error Correction (FEC)
      - Will correct errors automatically
    - Error Detection
      - Can detect errors for retransmission of data
  - Re-send data (the simplest of all methods)
    - Re-send data that is bad 2-3 times and make decision (2/3 rule)
  - These techniques can be used on the entire data or data that is determined to be critical
    - For example, the DC component, Huffman tables, quantization tables of JPEG DCT

Image Sequence Compression (Video)

- Image sequences (neighboring frames) are often highly correlated, particularly if object motion is taken into account (motion compensation)
- Motion-compensated frame differencing can be used very effectively to reduce redundant information in sequences.
- Finding corresponding points between frames (i.e., motion estimation) can be difficult because of occlusion, noise, illumination changes, etc.
- Motion vectors (x,y-displacements) are sent to the receiver to indicate corresponding points; these vectors are usually computed over blocks of pixels (e.g., 16x16) to minimize overhead.

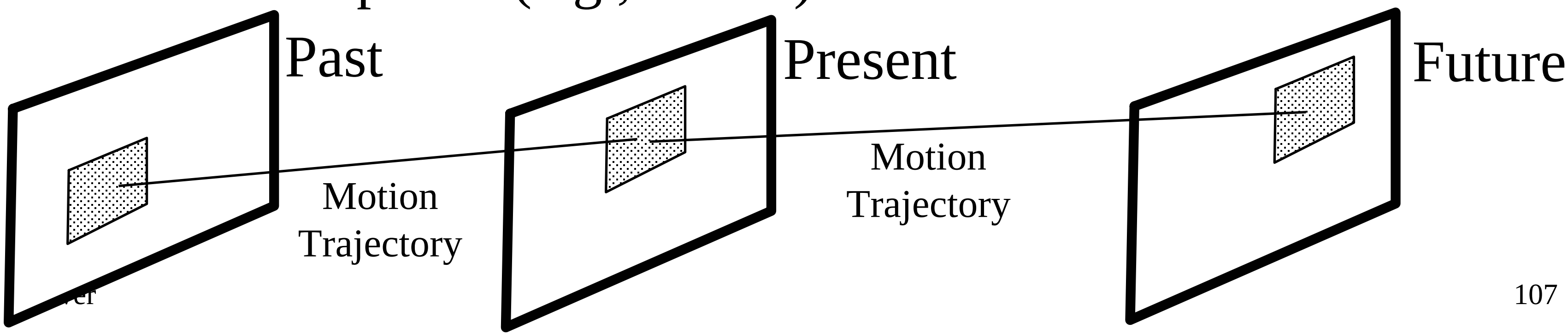
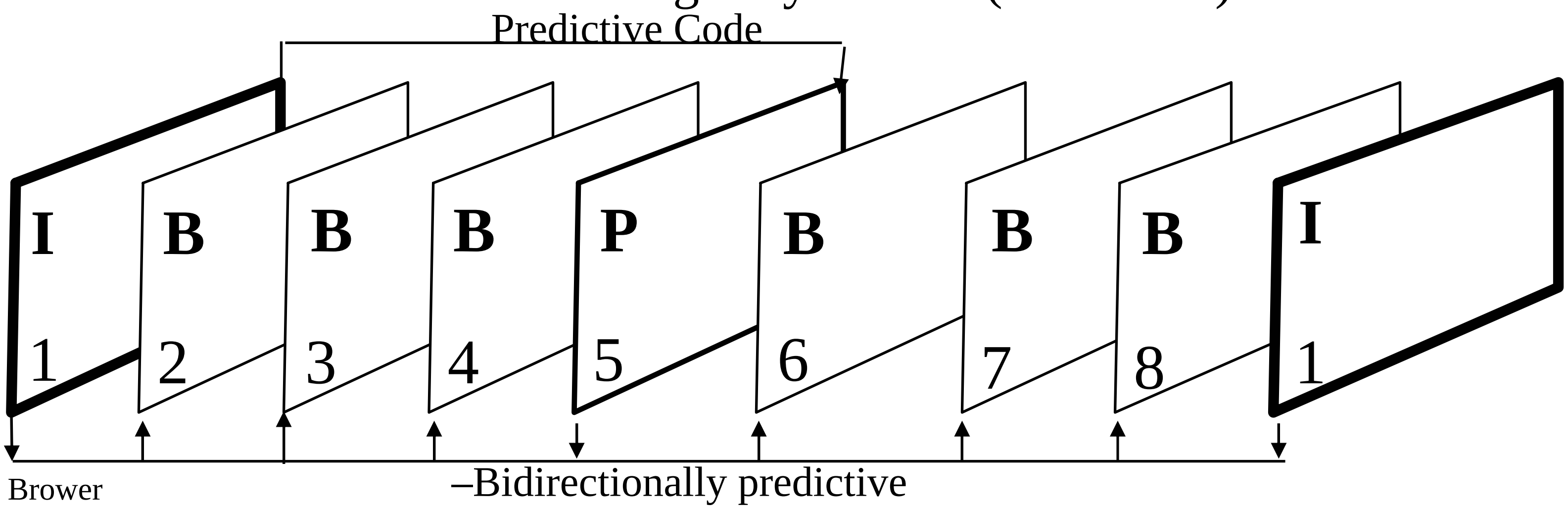


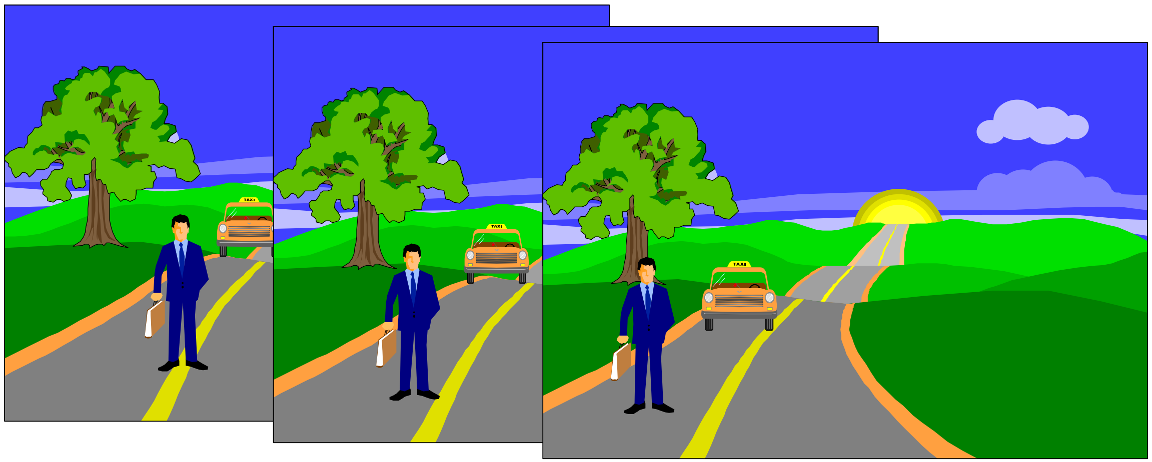
Image Sequence Compression (Video)

- The MPEG system specifies three types of frames within a sequence:
  - **Intra-coded picture** (I-frame): Coded independently from all other frames in the sequence. Uses the most number of bits.
  - **Predictive-coded picture** (P-frame): Coded based on a prediction from a past I- or P-frame. Uses less bits than an I-frame.
  - **Bidirectionally predictive coded picture** (B-frame): Coded based on a prediction from a past and/or future I- or P-frame(s). Uses the least number of bits and cannot be used as a reference for prediction.
  - Each frame is encoded using 8-by-8 DCT (JPEG like)



MPEG-7

- Object based motion compression
  - Separate objects (background, object 1, object 2)
  - Compress each object separate
    - Send updates to objects not background



Original Scene

Background

Object 1

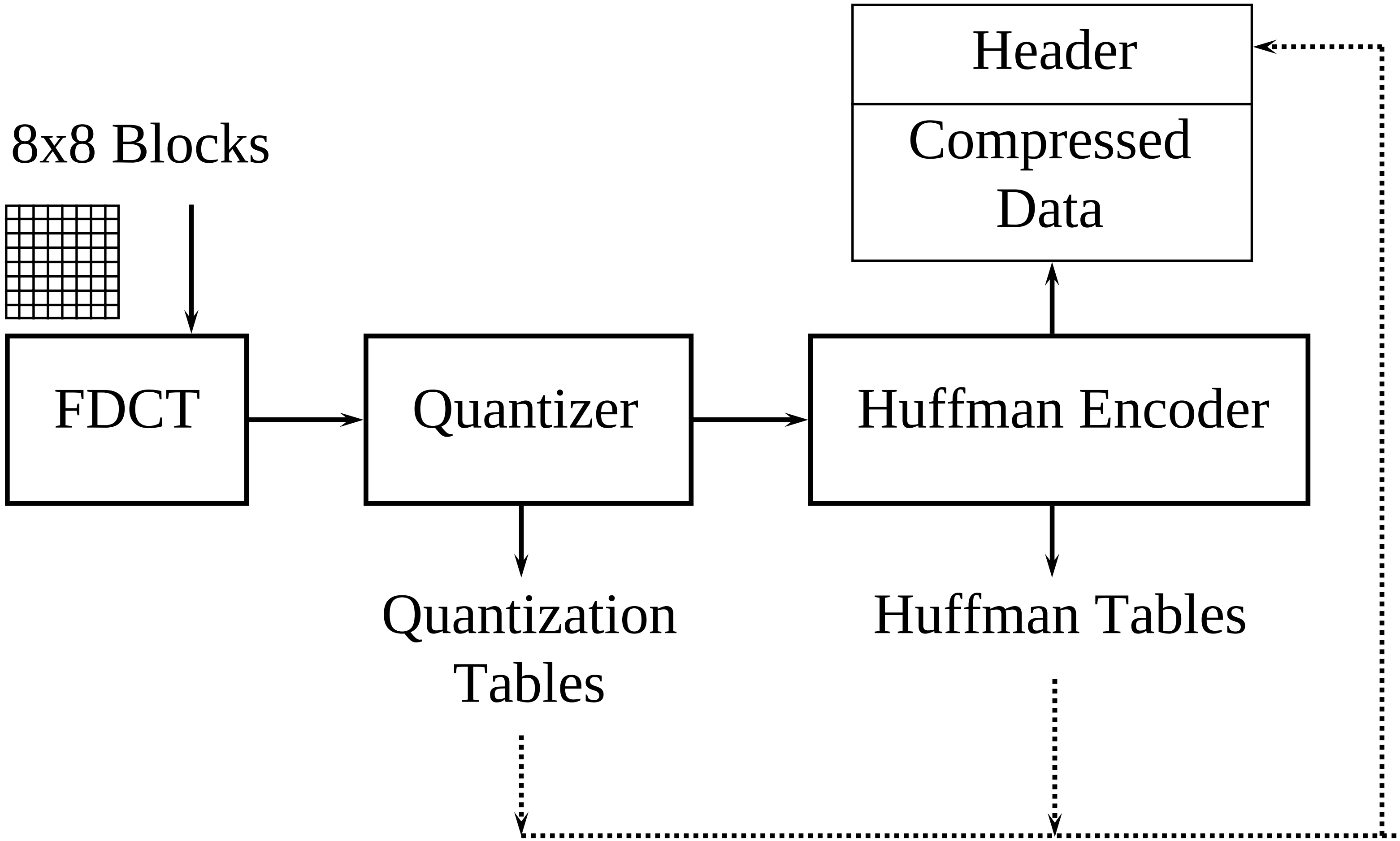
Object 2

The Existing DCT-Based JPEG Standard

What Is JPEG?

- The JPEG (**Joint Photographic Experts Group**) committee, formed in 1986, has been chartered with the
  - "*Digital compression and coding of continuous-tone still images*"
- Joint between ISO and ITU-T
- Has developed standards for the lossy, lossless, and nearly lossless of still images in the past decade
- Website: **www.jpeg.org**
- The JPEG committee has published the following standards:
  - ISO/IEC 10918-1 | ITU-T Rec. T.81 : *Requirements and guidelines*
  - ISO/IEC 10918-2 | ITU-T Rec. T.83 : *Compliance testing*
  - ISO/IEC 10918-3 | ITU-T Rec. T.84: *Extensions*
  - ISO/IEC 10918-4 | ITU-T Rec. T.86: *Registration of JPEG Parameters, Profiles, Tags, Color Spaces, APPn Markers Compression Types, and Registration Authorities (REGAUT)*

JPEG Encoder Block Diagram



JPEG Decoder Block Diagram

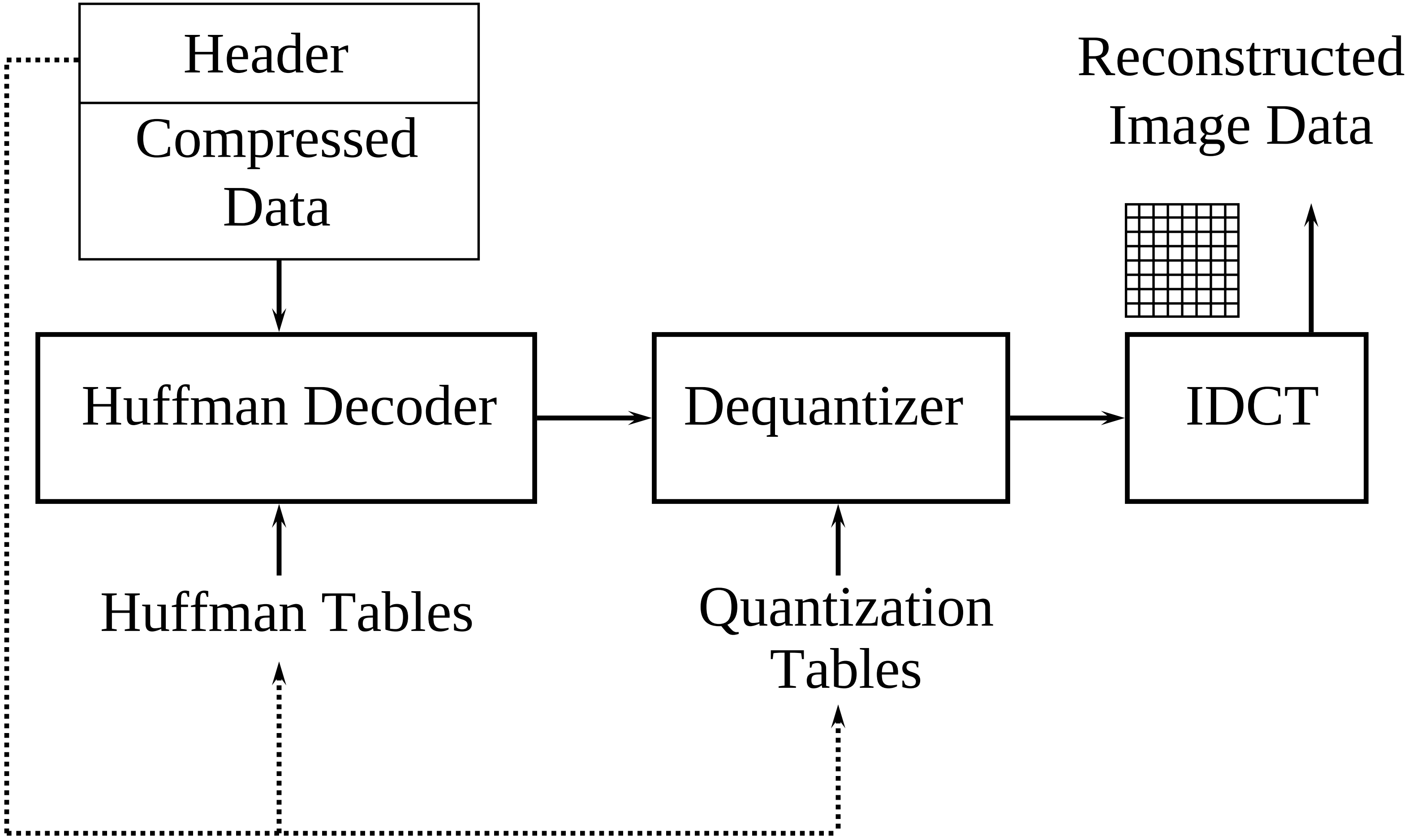
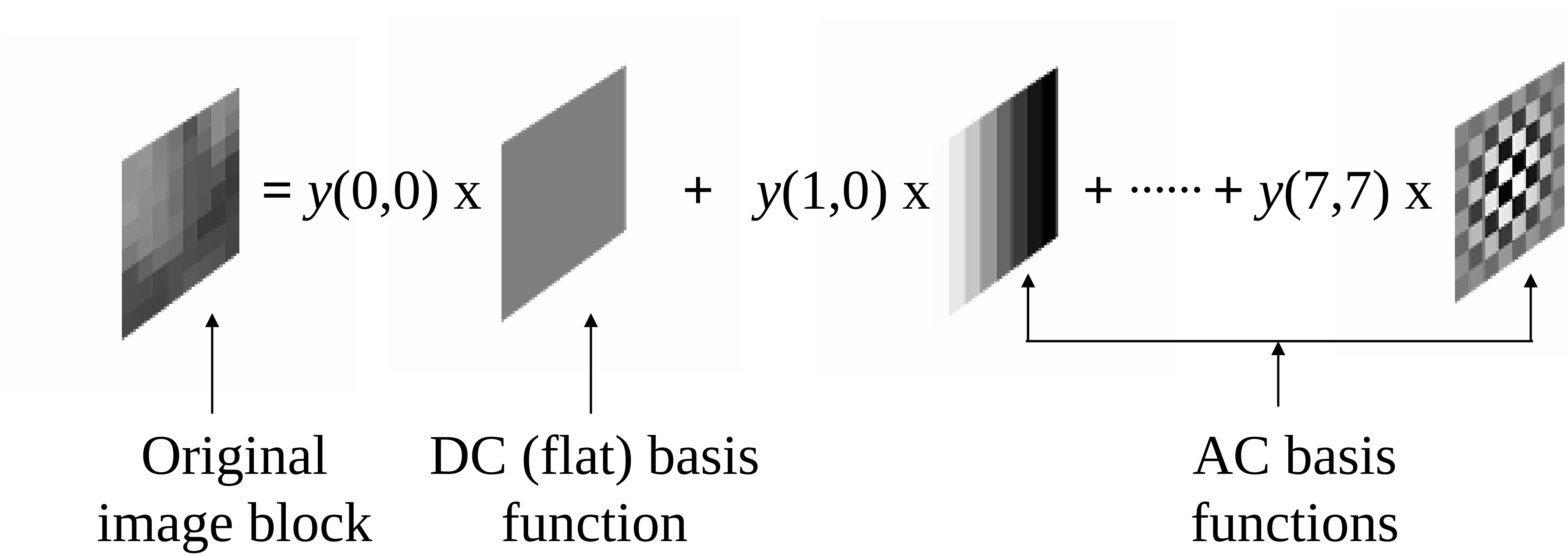


Image Representation with DCT

- DCT coefficients can be viewed as weighting functions that, when applied to the 64 cosine basis functions of various spatial frequencies (8 x 8 templates), will reconstruct the original block.



JPEG DCT Example

Original Image

139	144	149	153	155	155	155	155
144	151	153	156	159	156	156	156
150	155	160	163	158	156	156	156
159	161	162	160	160	159	159	159
159	160	161	162	162	155	155	155
161	161	161	161	160	157	157	157
162	162	161	163	162	157	157	157
162	162	161	161	163	158	158	158

8 BPP
64 pixels
512 bits

DCT Transformed Image

235.6	-1.0	-12.1	-5.2	2.1	-1.7	-2.7	1.3
-22.6	-17.5	-6.2	-3.2	-2.9	-0.1	0.4	-1.2
-10.9	-9.3	-1.6	1.5	0.2	-0.9	-0.6	-0.1
-7.1	-1.9	0.2	1.5	0.9	-0.1	0.0	0.3
-0.6	-0.8	1.5	1.6	-0.1	-0.7	0.6	1.3
1.8	-0.2	1.6	-0.3	-0.8	1.5	1.0	-1.0
-1.3	-0.4	-0.3	-1.5	-0.5	1.7	1.1	-0.8
-2.6	1.6	-3.8	-1.8	1.9	1.2	-0.6	-0.4

Quantized/ Scaled Transformed Data

15	0	-1	0	0	0	0	0
-2	-1	0	0	0	0	0	0
-1	-1	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

DCT Coefficient Quantization

- Each DCT coefficient is uniformly quantized with a quantization step that is taken from a user-defined **quantization table** (q-table or normalization matrix), characterized by 64, 1-byte elements.

- The quality and compression ratio of an encoded image can be varied by changing the q-table elements (usually by scaling up or down the values of an initial q-table).

- The q-table is often designed according to the perceptual importance of the DCT coefficients (e.g., by using the HVS CSF data) under the intended viewing conditions.

- For the baseline system, in order to meet the needs of the various color components, up to four different quantization tables are allowed.

Example of Luminance Quantization Table

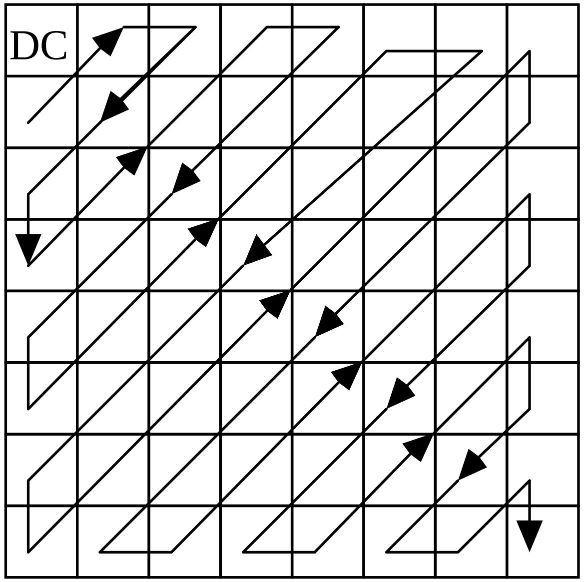
The JPEG committee has listed the following luminance quantization table as an example in Annex K (informative) of the IS. It is obtained by measuring the DCT coefficient "visibility threshold" using CCIR-601 images and display, at a distance of six picture-heights away.

$Q_L(u,v) =$

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

JPEG DCT Example

- After Quantization, the DCT is separated into a DC coefficient and AC coefficients, which are reordered into a 1-D format using a zigzag pattern in order to create long runs of zero-valued coefficients.
  - The DC coefficient is directly correlated to the mean of the 8-by-8 block (upper-left corner).
    - All DC coefficients are combined into a separate bit stream.
  - The AC coefficients are the values of the cosine basis functions (all other values).



JPEG DCT Example

DC	AC								
15	0	-2	-1	-1	-1	0	0	-1	EOB

- The DC coefficient is encoded using Huffman encoded 1D-DPCM
- The AC coefficients are encoded using Huffman coding on magnitude/runlength pairs (magnitude of a nonzero AC coefficient plus runlength of zero-valued AC coefficients that precede it).
- The end-of-block (EOB) symbol indicates that all remaining coefficients in the zigzag scan are zero. This allows many coefficients to be encoded with only a single symbol.

JPEG DCT Example

Dequantized DCT Coefficients

240	0	-10	0	0	0	0	0
-24	-12	0	0	0	0	0	0
-14	-13	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Reconstructed Image

144	146	149	152	154	156	156	156
148	150	152	154	156	156	156	156
155	156	157	158	158	157	156	155
160	161	161	162	161	159	157	155
163	163	164	163	162	160	158	156
163	163	164	164	162	160	158	157
160	161	162	162	162	161	159	158
158	159	161	161	162	161	159	158

Error Image

-5	-2	0	1	1	-1	-1	-1
-4	1	1	2	3	0	0	0
-5	-1	3	5	0	-1	0	1
-1	0	1	-2	-1	0	2	4
-4	-3	-3	-1	0	-5	-3	-1
-2	-2	-3	-3	-2	-3	-1	0
2	1	-1	1	0	-4	-2	-1
4	3	0	0	1	-3	-1	0

RMSE= 2.26

The Emerging JPEG2000 Standard

JPEG2000 Objectives

- Advanced standardized image coding system to serve applications into the next millenium
- Address areas where current standards fail to produce the best quality or performance
- Provide capabilities to markets that currently do not use compression
- Provide an open system approach to imaging applications

JPEG2000: Requirements and Profiles

- Internet applications (World Wide Web imagery)
  - Progressive in quality and resolution, fast decode
- Mobile applications
  - Error resilience, low power, progressive decoding
- Digital photography
  - Low complexity, compression efficiency
- Hardcopy color facsimile, printing and scanning
  - Compression efficiency, strip or tile processing
- Digital library/archive applications
  - Metadata, content management
- Remote sensing
  - Multiple components, fast encoding, region of interest
- Medical applications
  - Region of interest coding, lossy to lossless

Old Compression Paradigm (JPEG Baseline)

Encode

<u>Encoder choices</u>

color space

quantization

entropy coder

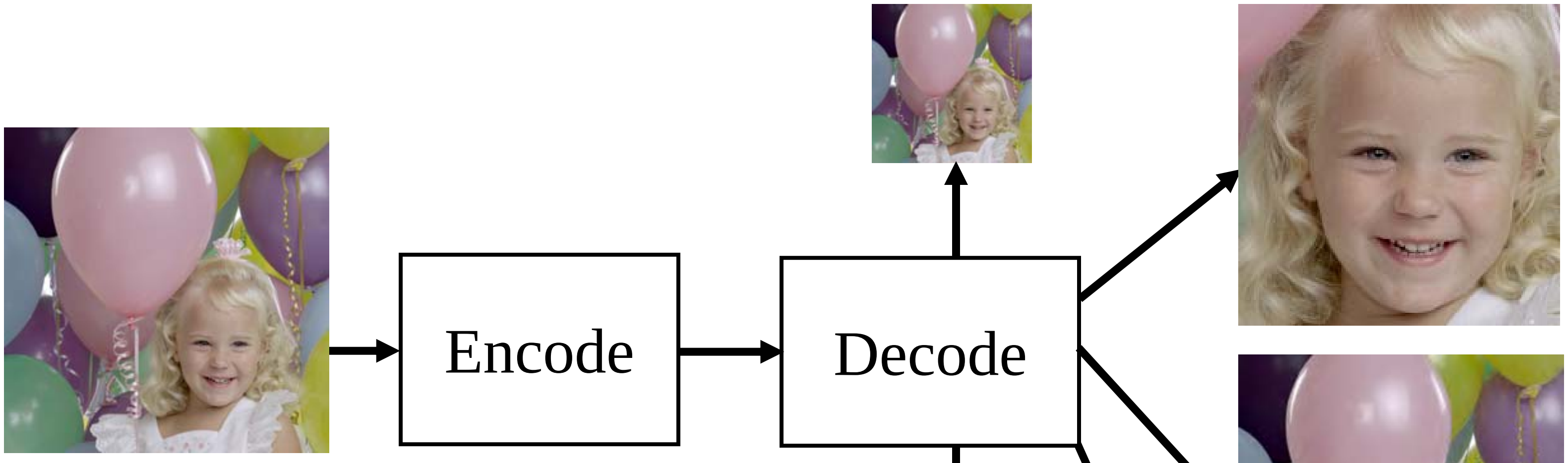
pre-processing

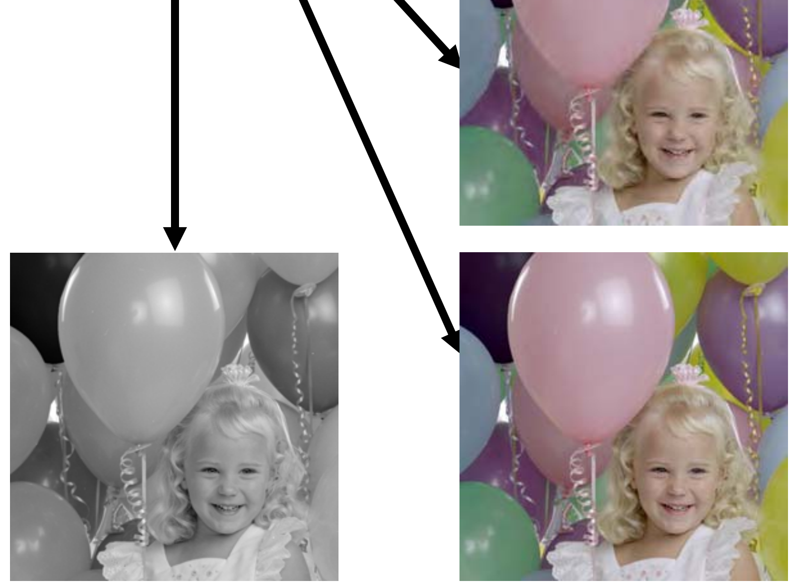
<u>No decoder choices</u>

only one image

post-processing

JPEG2000 Compression Paradigm





<u>Decode Choices</u>

Image resolution
SNR fidelity
Visual fidelity
Target filesize
Lossless/lossy
Region-of-interest
Tiles

JPEG2000 Features

- Improved compression efficiency (estimated 5-30% depending on the image size and bit rate)
- Lossy to lossless
- Multiple resolution
- Embedded bit stream (progressive decoding)
- Region of interest coding (ROI)
- Error resilience
- Bit stream syntax
- File format

Status of Current ISO Standards

- JPEG 2000 Image Compression Standards
  - Part 1, Core coding system (intended as license & fee free but not patent-free)*
  - Part 2, Extensions (adds more features and sophistication to the core)+
  - Part 3, Motion JPEG 2000+
  - Part 4, Conformance*
  - Part 5, Reference software (Java and C implementations are available)
  - Part 6, Compound image file format (document imaging)
  - Part 7 has been abandoned
- Currently being developed
  - Part 8, JPSEC: working draft (security aspects)
  - Part 9, JPIP: Final CD – DIS in March (interactive protocols and API)+
  - Part 10, JP3D: working draft (volumetric imaging)
  - Part 11, JPWL: working draft (wireless applications)
  - Part 12, Common format text (Motion J2K, MPEG 4) DIS
  - Part 13, Standard Encoder: Working Draft

JPEG2000 Compression Standard

- The standard only specifies a decoder and a bitstream syntax and is issued in several parts:
  - **Part I:** specifies the minimum compliant decoder (e.g., a decoder that is expected to satisfy 80% of applications); International Standard (IS) was passed 12/2000.
    - Full description of compression technology
      - Includes wavelet transform, quantization, encoding, compression bit stream, and file format
    - Baseline of all other parts
  - **Part II:** Describes optional, value added extensions; IS was passed 12/2001.
    - This includes multiple component compression
      - Take advantage of the correlation between components
      - Currently being evaluated by NGA and DICOM
    - Includes other wavelets, TCQ, . . .

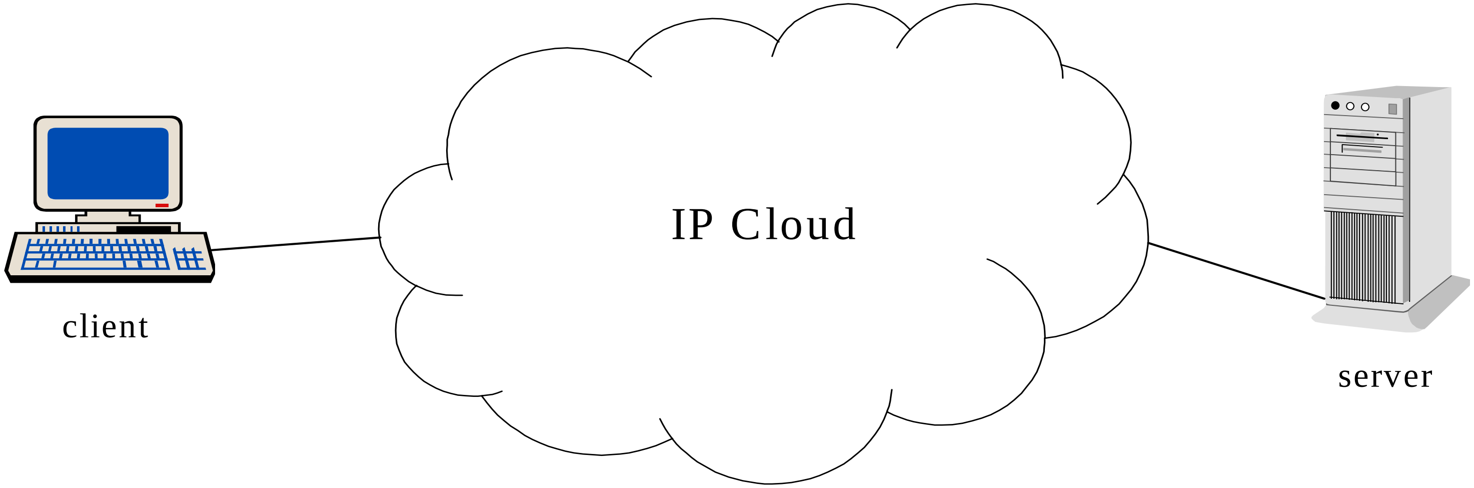
JPEG2000 Compression Standard

- Other parts:
  - **Part III:** Motion JPEG 2000.
    - Uses baseline JPEG 2000 (Part 1)
    - Uses file format from MPEG-4
    - **<u>No</u>** frame-to-frame motion compensation
    - Has timing and audio support
    - Selected as the format by SIMPTE for digital cinema!
  - **Part IV:** Describes compliance testing of JPEG 2000.
    - Open standard (can download of the web)
    - Includes test imagery
    - Used for baseline JITC testing
  - **Part V:** Reference software JPEG 2000.
    - JAVA and C++ example software
    - Compliance tested (not fully compliant)

JPEG2000 Compression Standard

- Other parts:
  - **Part VI:** Compound Document (IS)
    - Segmentation of scanned image
    - Separate image, text, and graphics in to different files
    - Compress each one separate.
    - Should be looked at for ADRG
  - **Part VII:** Abandoned
  - **Part VIII:** JPSEC- Security. (CD)
    - Security for JPEG 2000 compressed images
    - Includes the techniques and file format to define security definitions and techniques within a file
    - For example: an encryption can be used for each quality or resolution layer
      - A key can be used for each layer or resolution

JPIP Simple Functional Goal

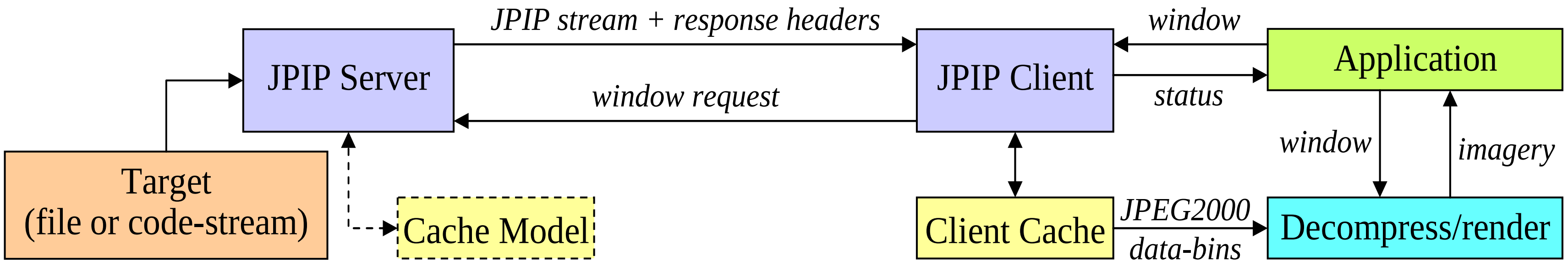


- Client makes “simple” requests to view a part of an image
  - Zoom, Pan, Get metadata
- Server makes a sensible response
  - Optimise for data transmission efficiency – i.e. relevant data to request
  - Plus extra important data the client should know (e.g. metadata titles)

JPIP Fundamental Concepts

- JPEG 2000 Internet Protocol - ISO 15444–9
  - Standardizes protocol for interactive and scalable access to JPEG 2000 coded image data and file format header and metadata
- Self-describing response data-stream sent by server
  - Equivalent as an "extension" to the code-stream structure
  - Allow arbitrary ordering of compressed data elements
  - Can build up information across multiple sessions, etc.
  - Mime types:
    - Precinct based - image/jpp-stream
    - Tile based - image/jpt-stream
  - Can be stored and read as a file in its own right
- Client requests
  - Simple description by image parameters (e.g. size, offset)
  - Explicit description of JPEG2000 elements (headers, packets)

JPIP Fundamental Concepts



- Data-Bins
  - All code-stream data is arranged into data-bins with unique ID
    - Imagery data (headers, precinct packets)
    - Metadata boxes
    - All tables of offsets
- File equivalents
  - Servers can condense complex structures (such as Motion JPEG2000 files MJ2, Compound JPEG2000 Documents JPM) into more simple equivalent files (either static or dynamically created) to reduce complexity of client requests

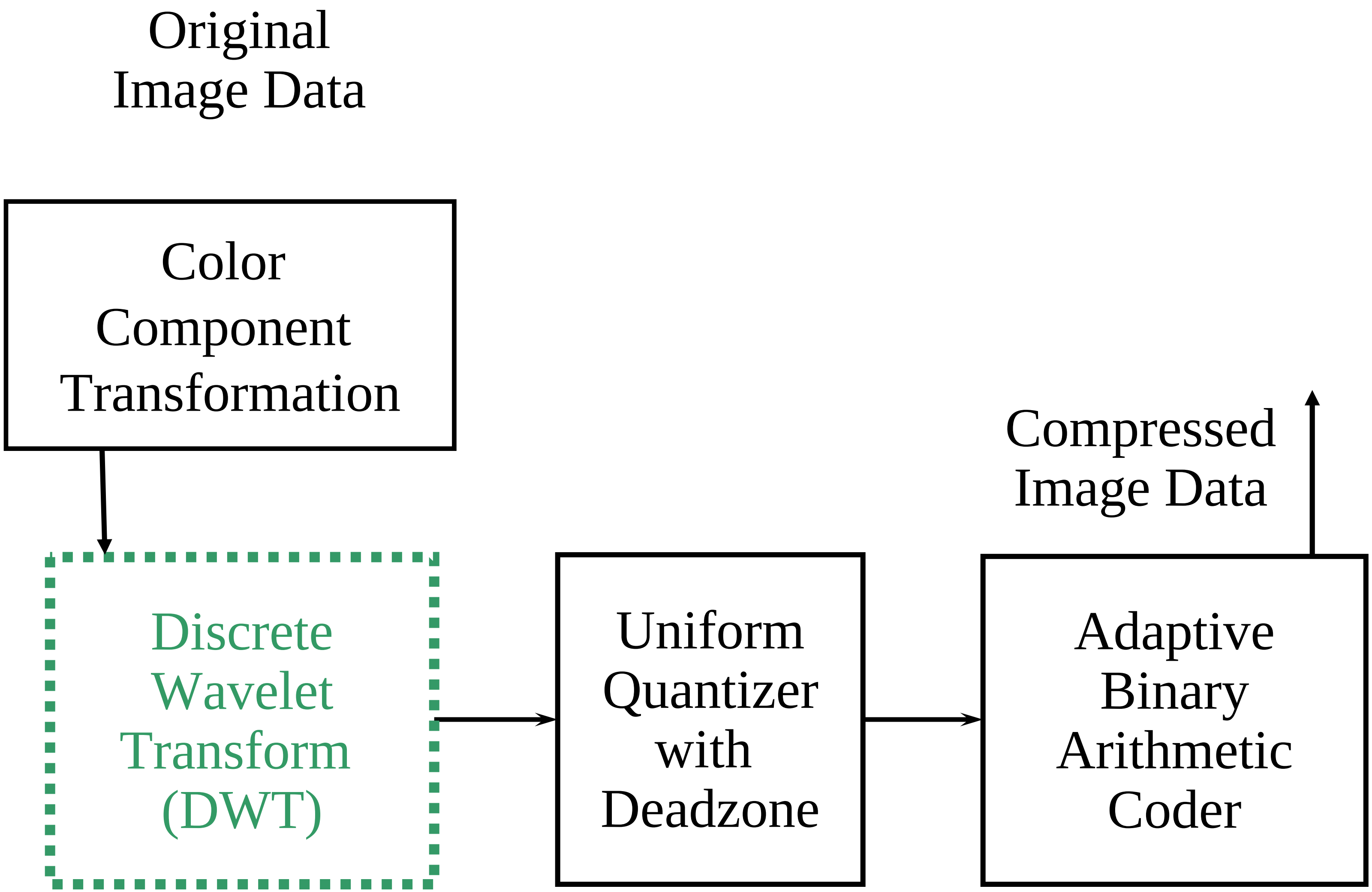
JPEG2000 Compression Standard

- Other parts:
  - **Part X:** Floating point and 3-D compression (WD)
    - Compression of floating point data
    - Current work stalled because of lack of funding for Los Alamos Labs
  - **Part XI:** JP Wireless (WD)
    - Compression dissemination and compression for wireless delivery
    - Mainly protection and other aspects for wireless communication
  - **Part XII:** JPEG 2000 Encoder (WD)
    - Baseline encoder
    - Reduce IP issues for encoder
    - Part 1 is only a decoder standard
    - Issue is that the IP statements are only for compliant standard

WG 1 New Work Items

- There are three new proposed work items in the JPEG committee
- Advanced Image Coding
  - Develop procedures and test to evaluate Advanced Image Coding techniques
  - Review the current status of Advanced Image Coding techniques
- Still Image Search
  - Development of technique to search for image content
  - Mainly based on search of metadata and standard metadata
  - Based on technology of MPEG-7 search
- Image Based Authentication
  - Using images as passwords
  - Humans remember imagery and imagery content better than random strings of numbers and letters

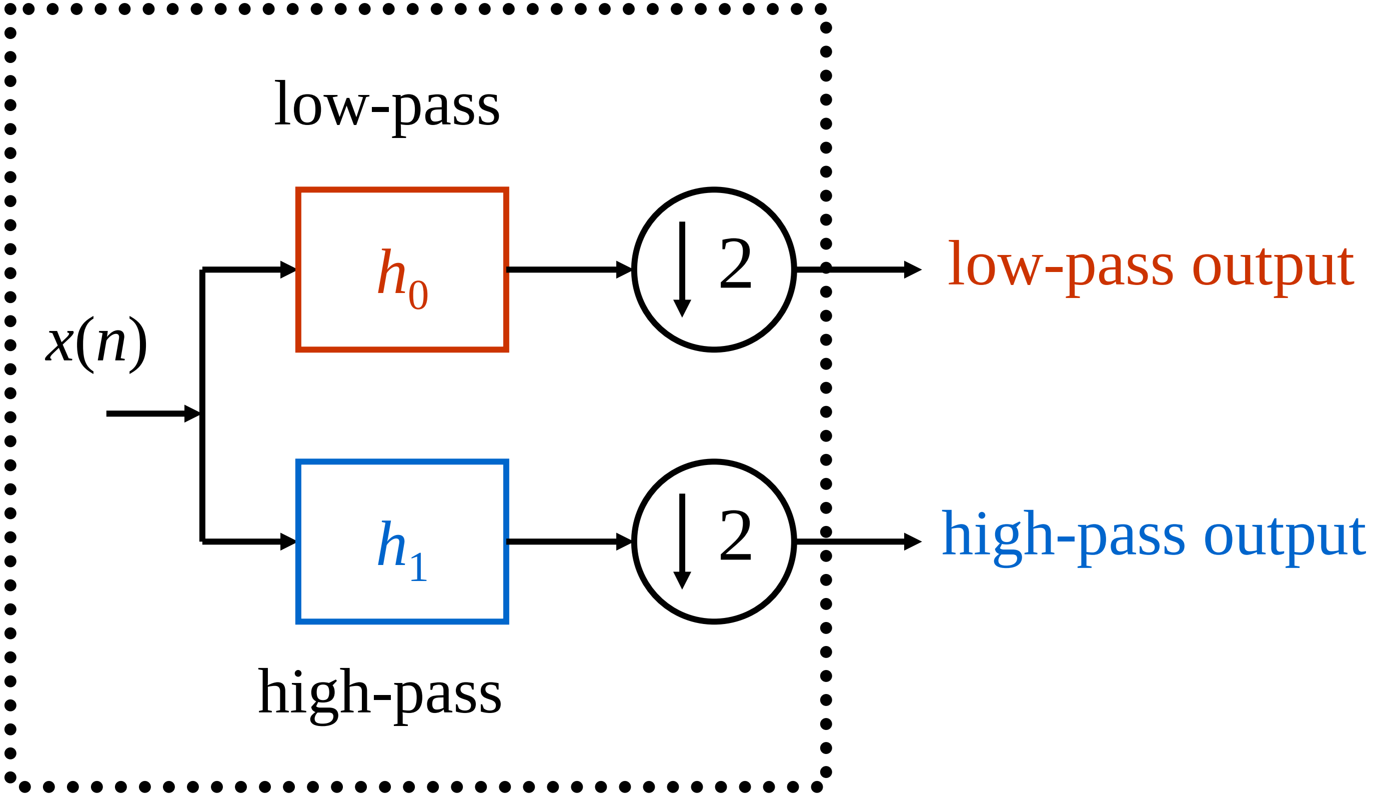
JPEG2000 Part I Encoder



Benefits of DWT

- Multiple resolution representation
- Lossless representation with integer filters
- Better decorrelation than DCT, resulting in higher compression efficiency
- Use of visual models
  - DWT provides a frequency band decomposition of the image where each subband can be quantized according to its visual importance (similar to the quantization table specification in JPEG-DCT)

The 1-D Two-Band DWT



Analysis filter bank

Example of Analysis Filter Bank

- 1-D signal:

  …100 100 100 100 200 200 200 200...

- Low-pass filter h_0: (-1 2 6 2 -1)/8

- High-pass filter h_1: (-1 2 -1)/2

- Before downsampling:

  … 100 100 87.5 112.5 187.5 212.5 200 200...

  … 0 0 0 -50 50 0 0 0…

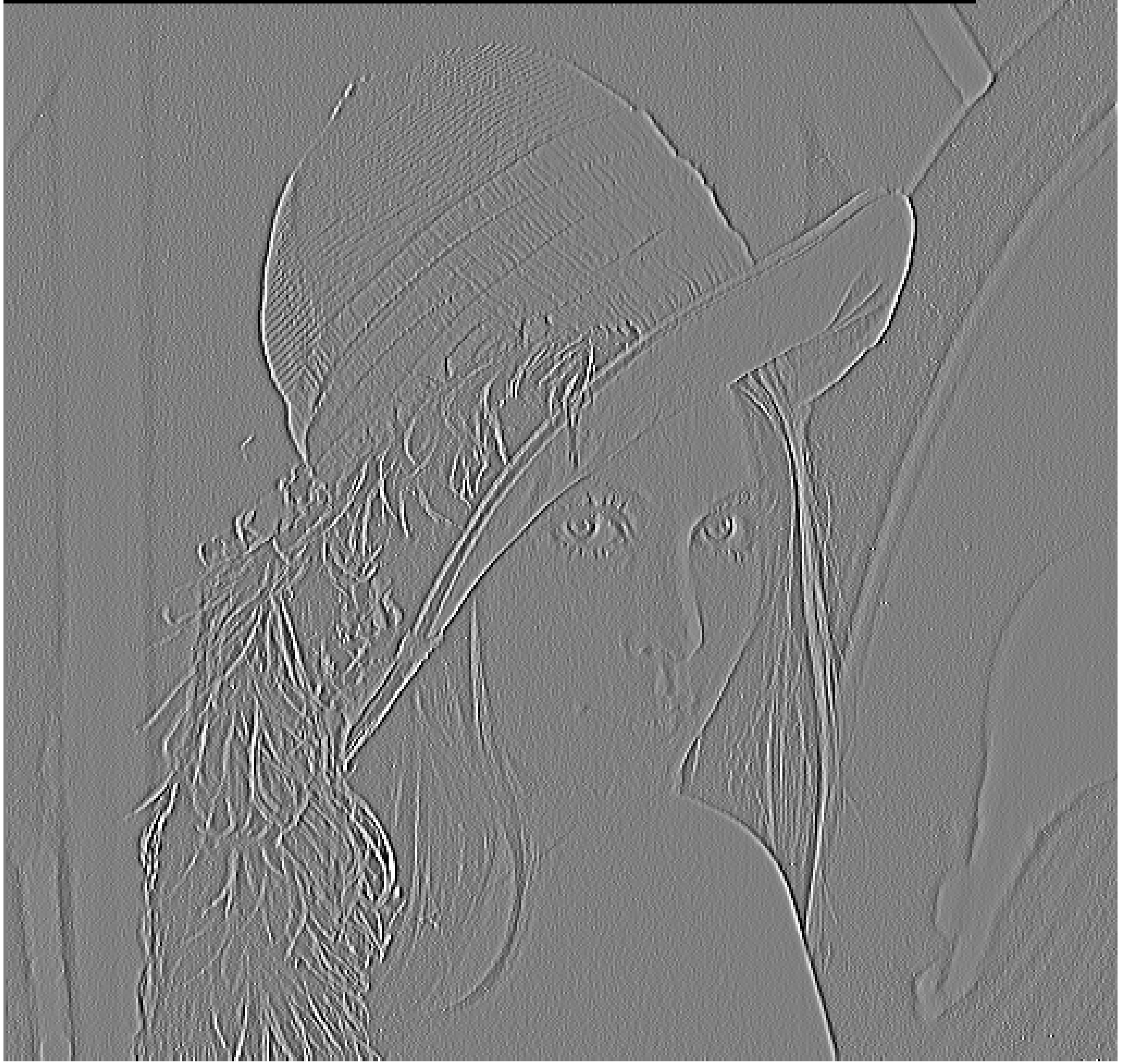
- After downsampling:

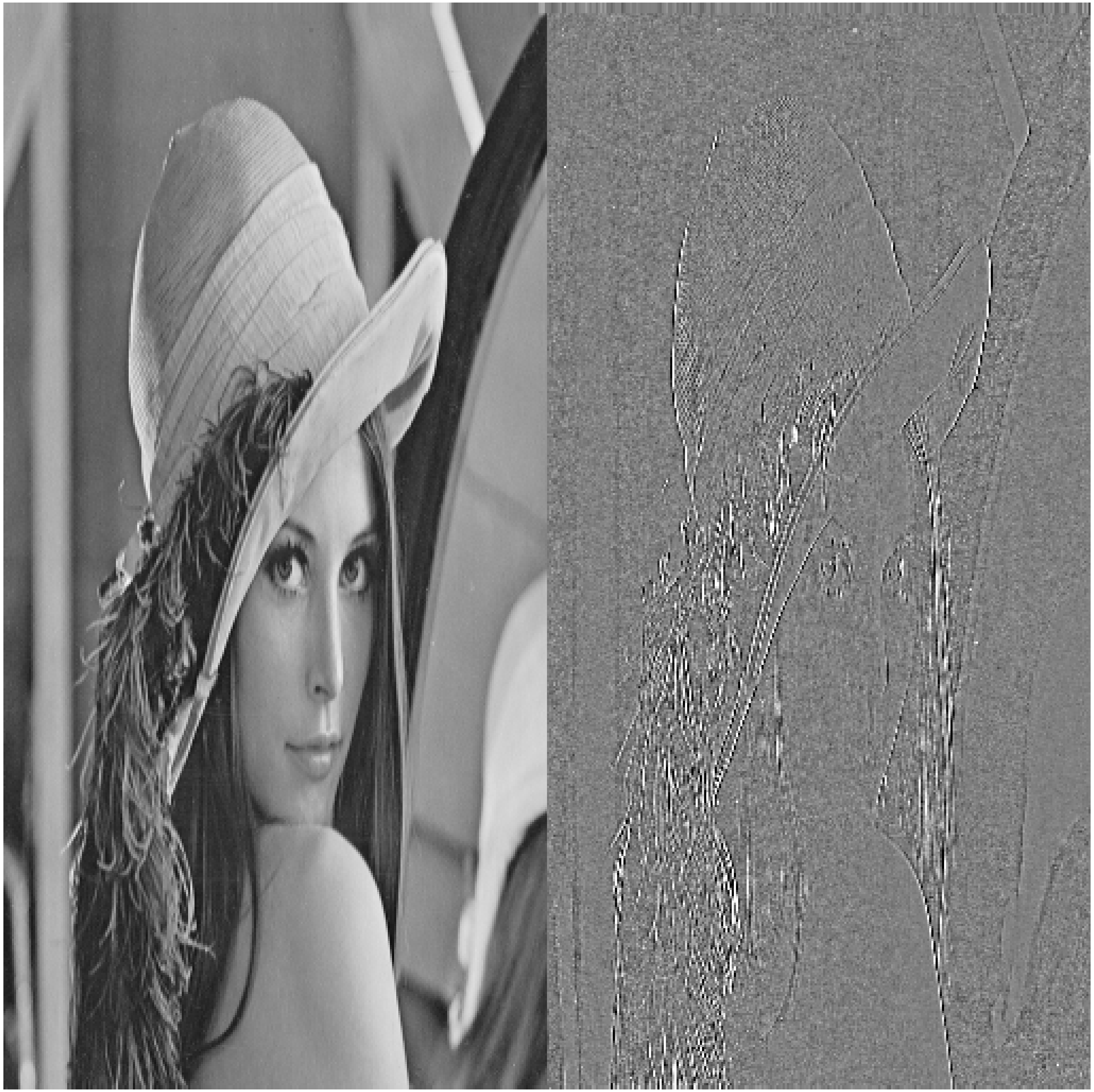
  … 100 112.5 212.5 200...

  … 0 0 50 0 …

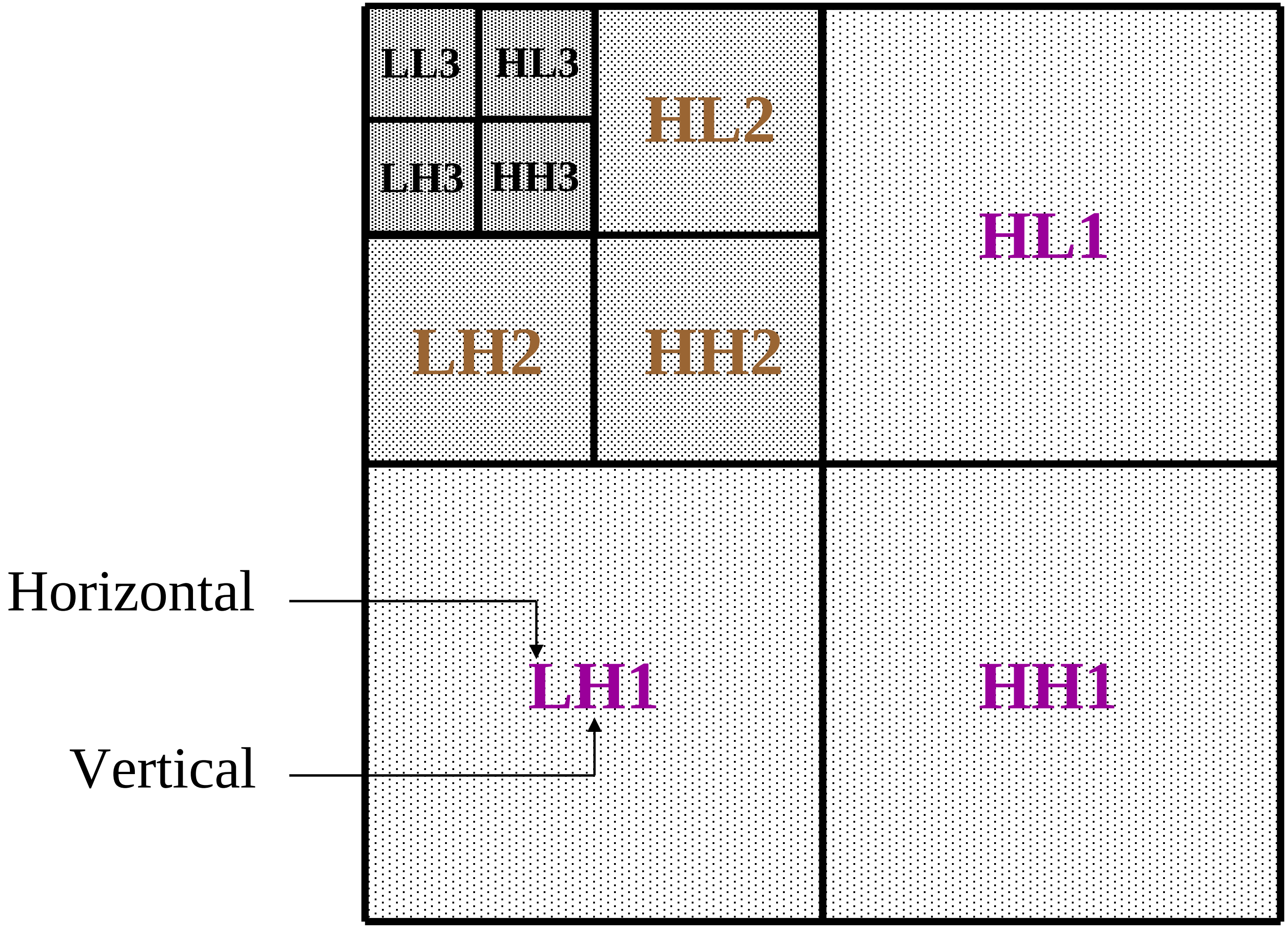
Horizontal Low-pass (-1, 2, 6, 2, -1)/8

Horizontal High-pass (-1, 2, -1)/2, Scaled up 4X





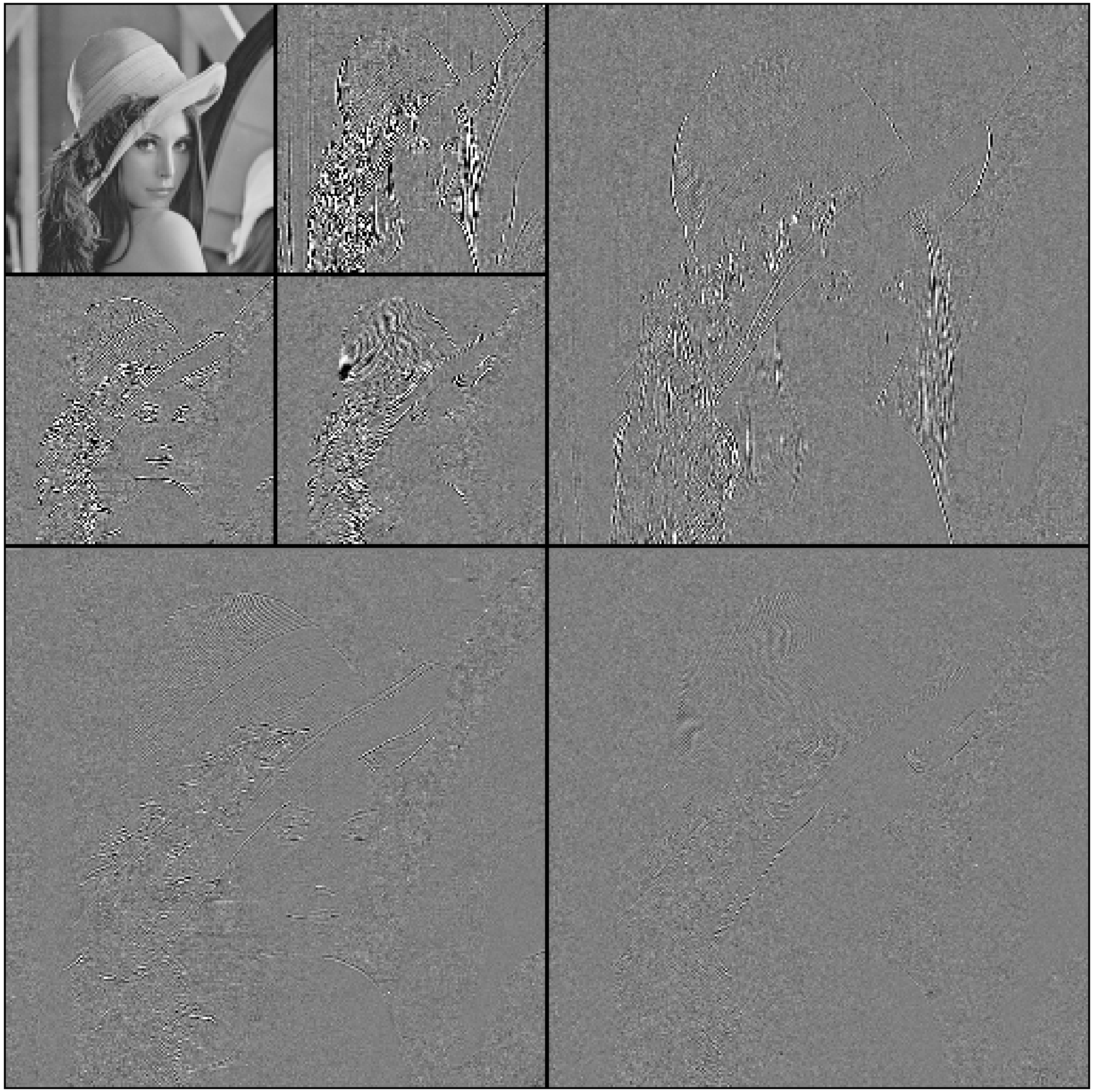
2-D Wavelet Decomposition



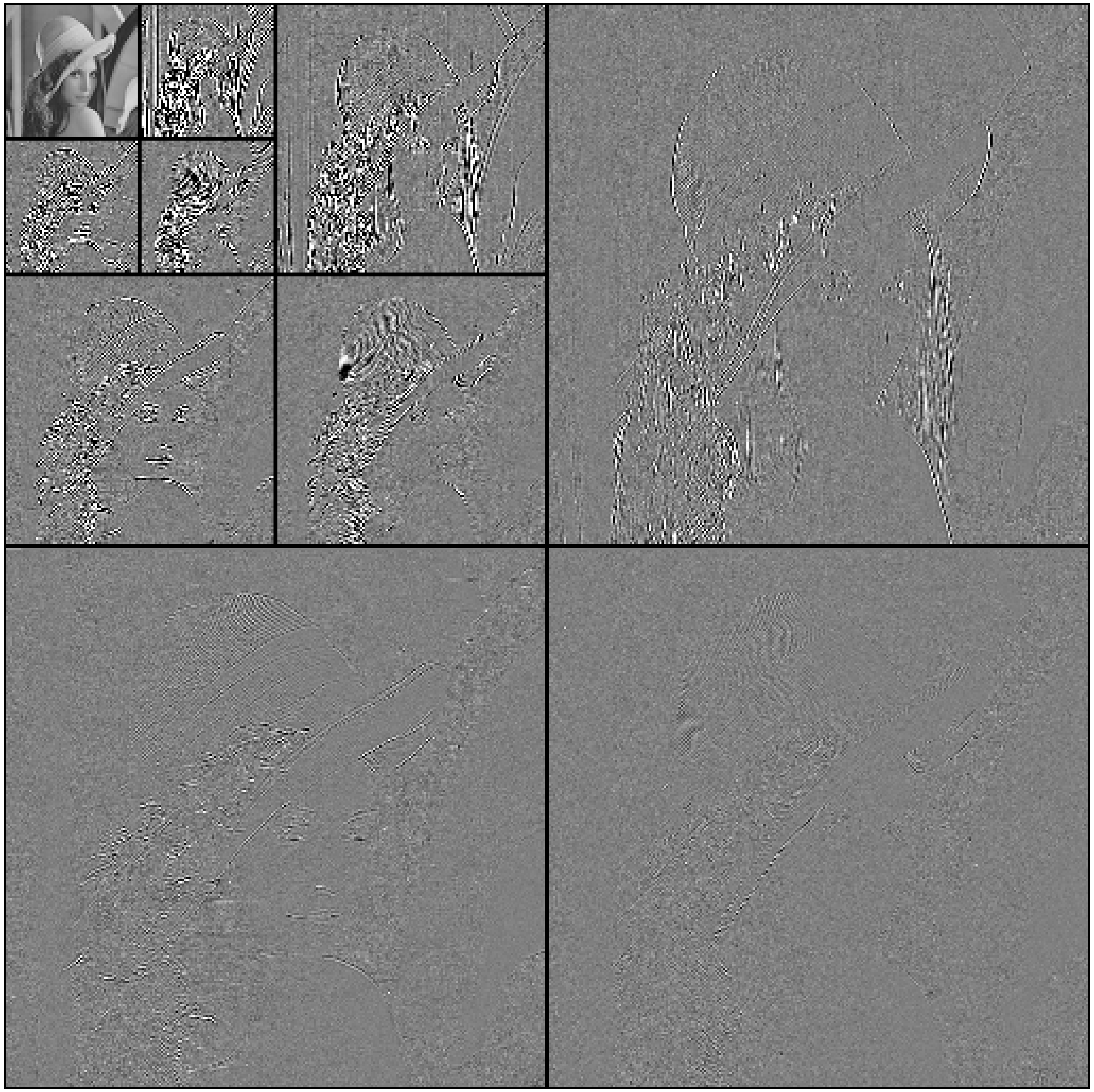
Original Lena Image

1-Level, 2-D Wavelet Decomposition of Lena

2-Level, 2-D Wavelet Decomposition of Lena



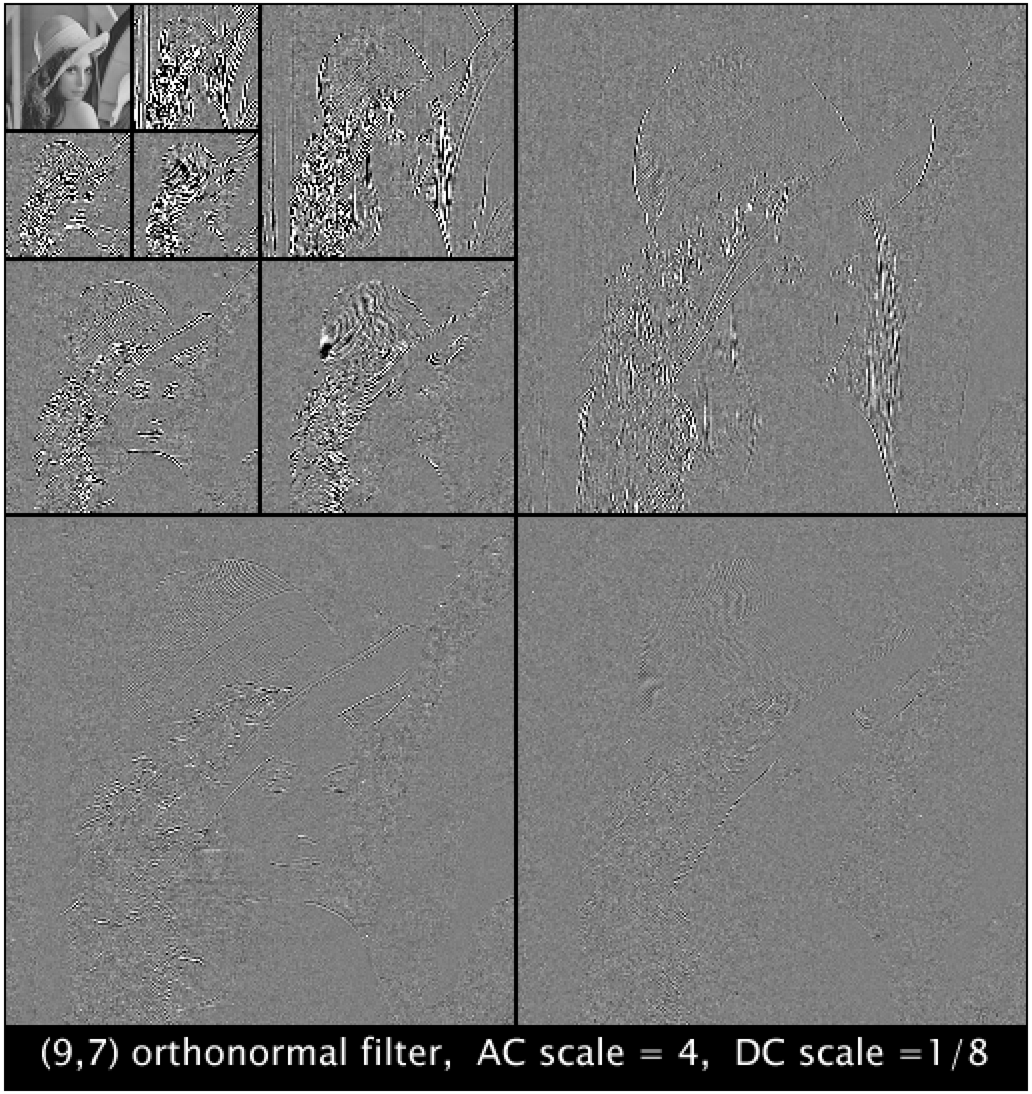
3-Level, 2-D Wavelet Decomposition of Lena



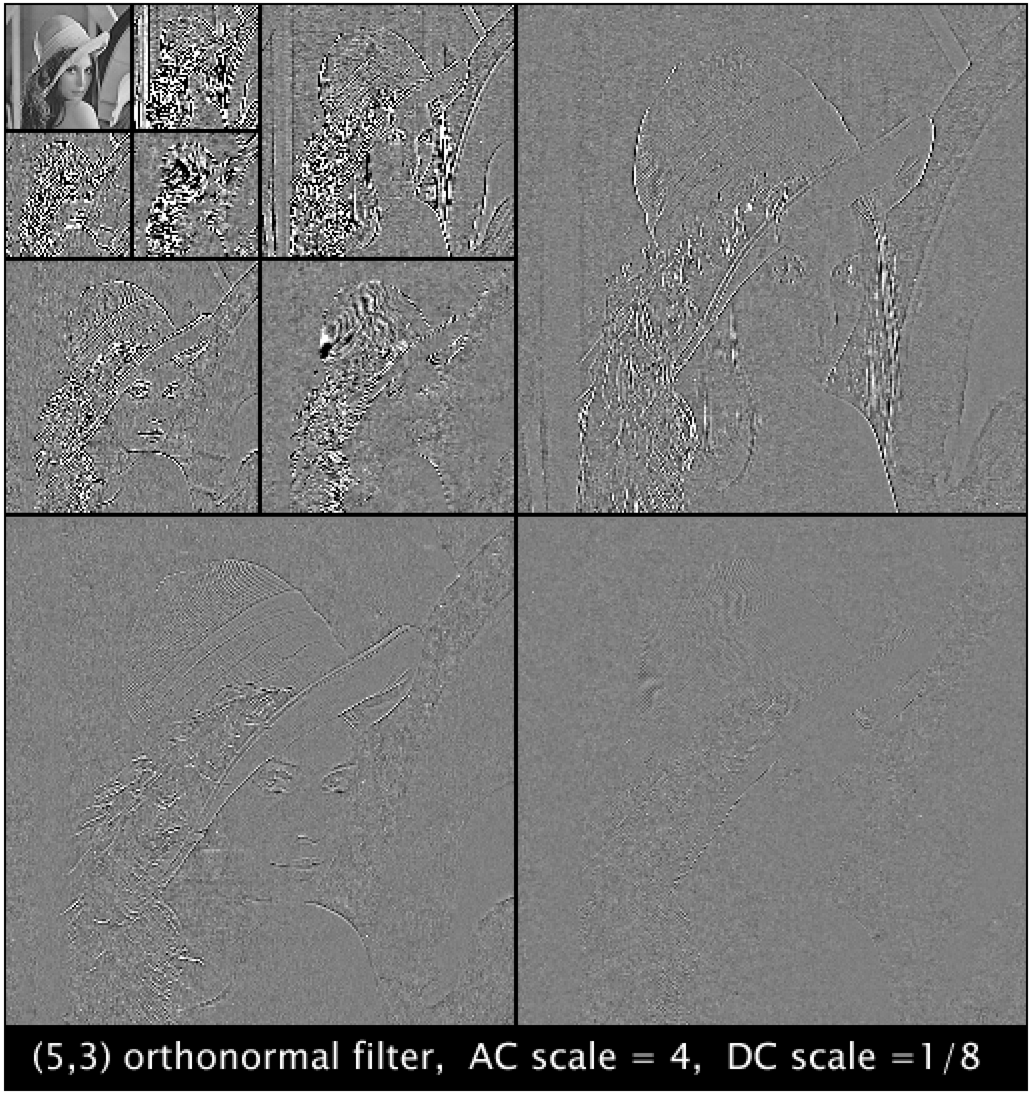
JPEG2000 DWT Choices

- JPEG-2000 Part I only allows successive powers of two splitting of the LL band and the use of two DWT filters:
  - The integer (5,3) filter provides fast implementation (faster than DCT) and lossless capability, but at the expense of some loss in coding efficiency.
  - The Daubechies (9,7) floating-point filter that provides superior coding efficiency. The analysis filters are normalized to a DC gain of one and a Nyquist gain of 2.
- Part II allows for arbitrary size filters (user-specified in the header), arbitrary wavelet decomposition trees, and different filters in the horizontal vs. vertical directions.

3-Level, 2-D DWT with (9,7) Filter



3-Level, 2-D DWT with (5,3) Filter



Inverse DWT

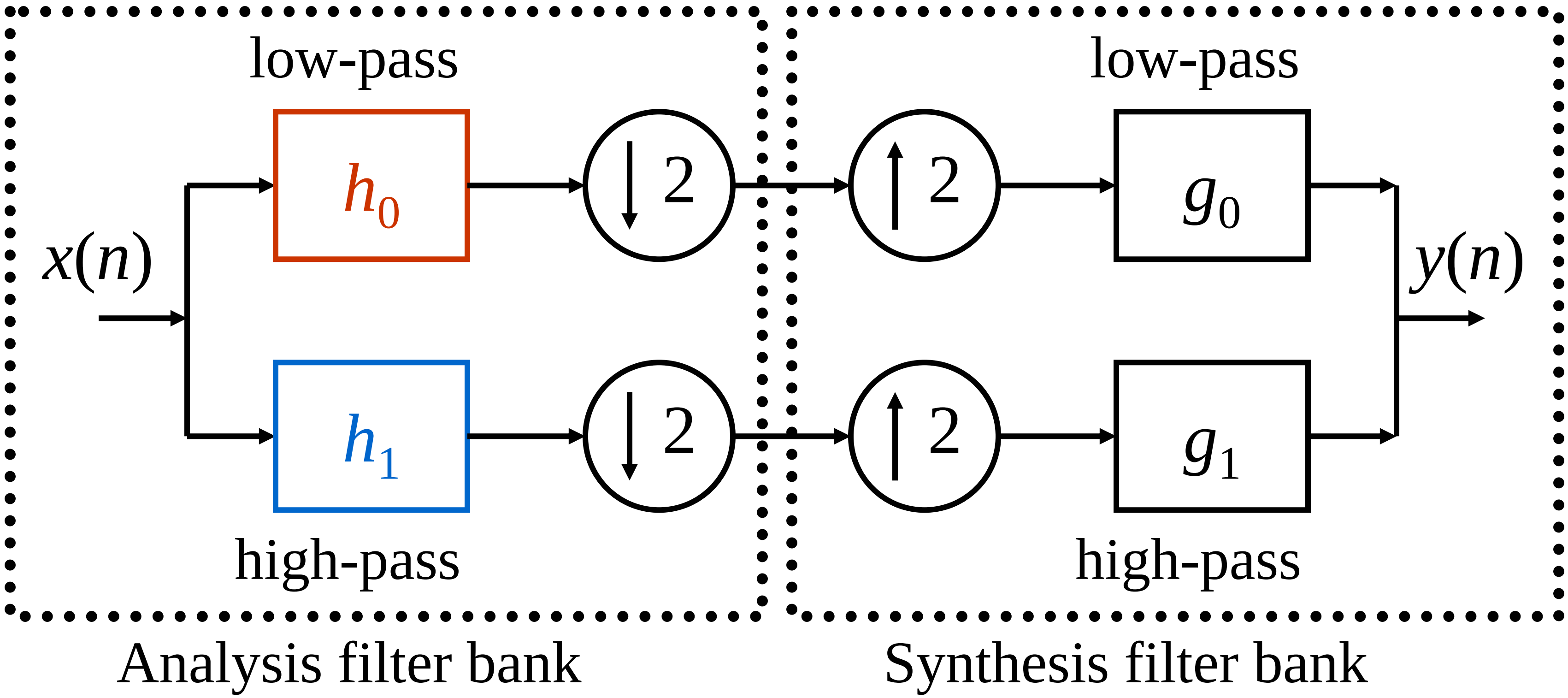
- During the inverse DWT, each subband is interpolated by a factor of two by inserting zeros between samples and then filtering each resulting sequence with the corresponding low-pass, g_0, or high-pass, g_1, **synthesis filters**.
- The filtered sequences are added together to form an approximation to the original signal.

… **0** 100 **0** 112.5 **0** 212.5 **0** 200…

… 0 **0** 0 **0** 50 **0** 0 **0**…

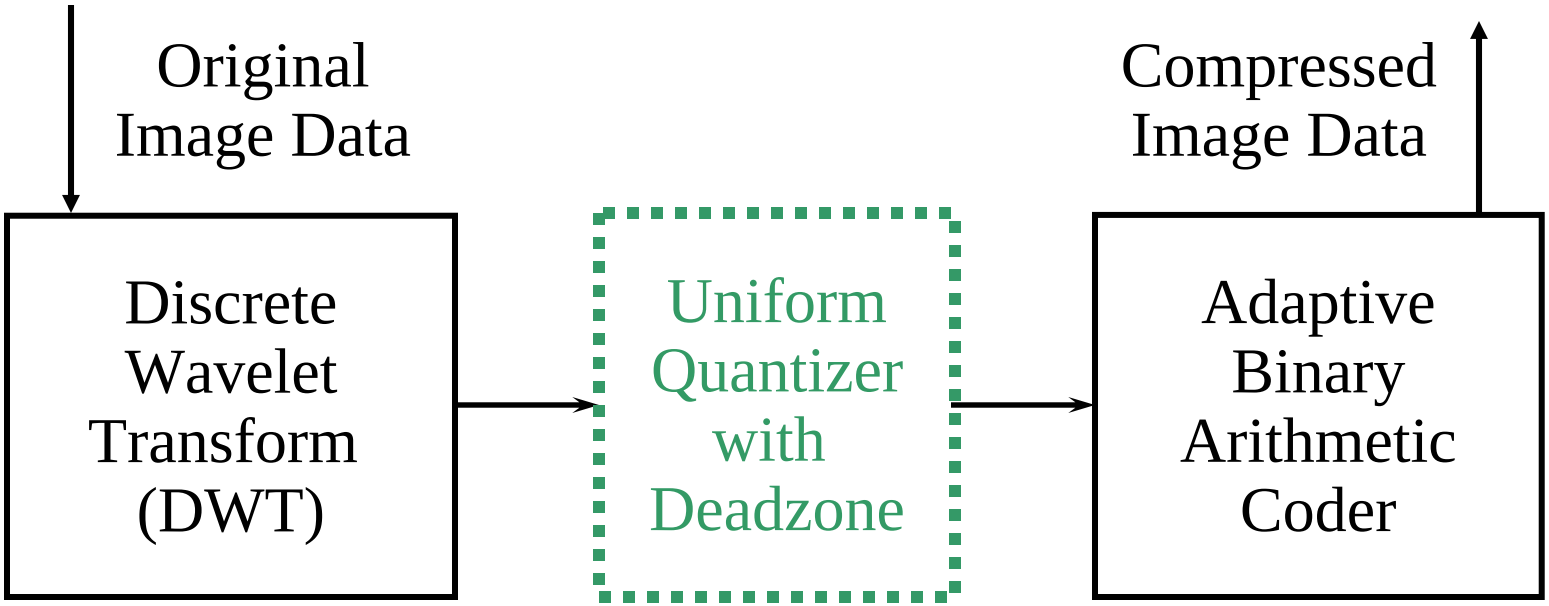
… 100 100 100 100 200 200 200 200...

The 1-D Two-Band DWT



Ideally, it is desired to choose the analysis filter banks (h_0 and h_1), and the synthesis filter banks (g_0 and g_1), in such a way so as to make the overall distortion zero, i.e., $x(n) = y(n)$. This is called the **perfect reconstruction** property.

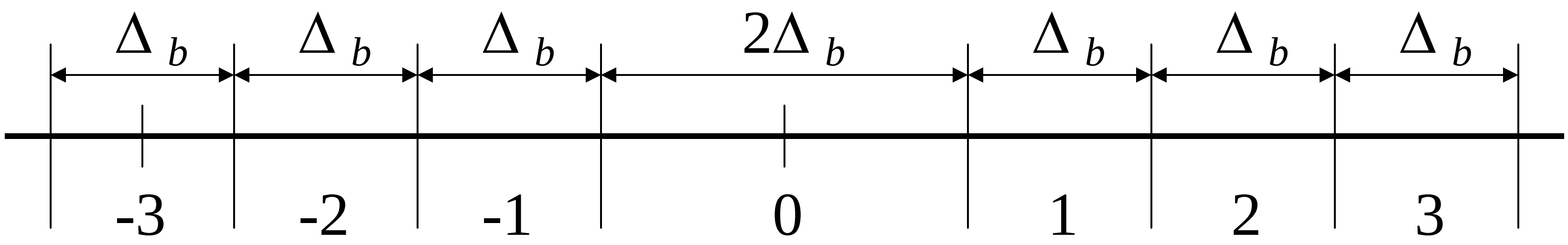
JPEG2000 Part I Encoder



Quantization step size can vary from one subband to another according to visual models (similar to JPEG Q-table specification).

Quantization in Part I

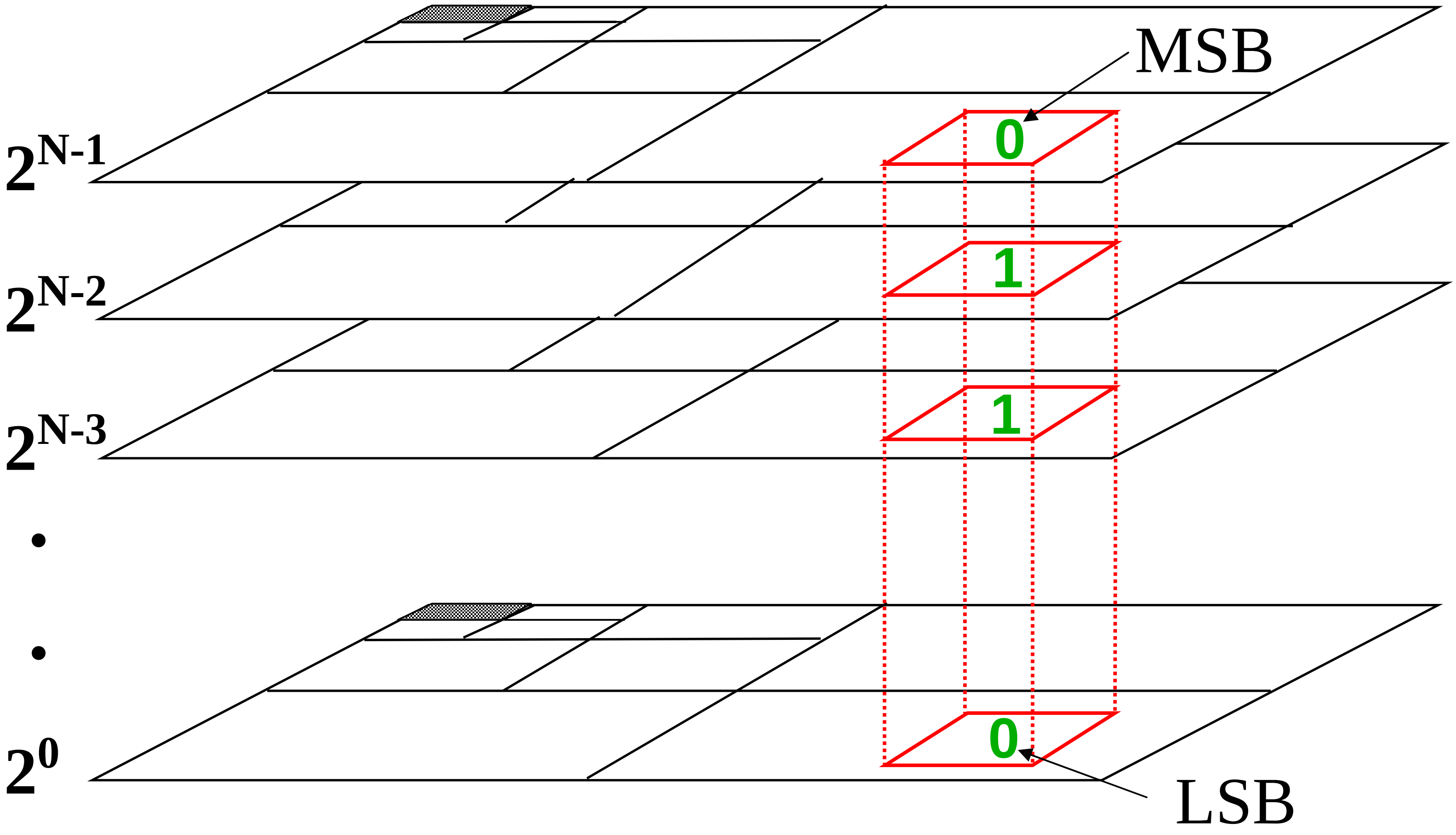
- Uniform quantization with deadzone is used to quantize all the wavelet coefficients.
- For each subband *b*, a basic quantizer step size Δ_b is selected by the user and is used to quantize all the coefficients in that subband.
- The choice of the quantizer step size for each subband can be based on visual models and is likened to the q-table specification in the JPEG DCT.



Embedded Quantization in Part I

- Unlike JPEG Baseline, where the resulting quantizer index q is encoded as a single symbol, in JPEG2000 it is encoded one bit at a time, starting from the MSB and proceeding to the LSB.
- During this progressive encoding, the quantized wavelet coefficient is called **insignificant** if the quantizer index q is still zero. Once the first nonzero bit is encoded, the coefficient becomes **significant** and its sign is encoded.
- If the p least significant bits of the quantizer index still remain to be encoded, the reconstructed sample at that stage is identical to the one obtained by using a UTQ with deadzone with a step size of $\Delta_b 2^p$.

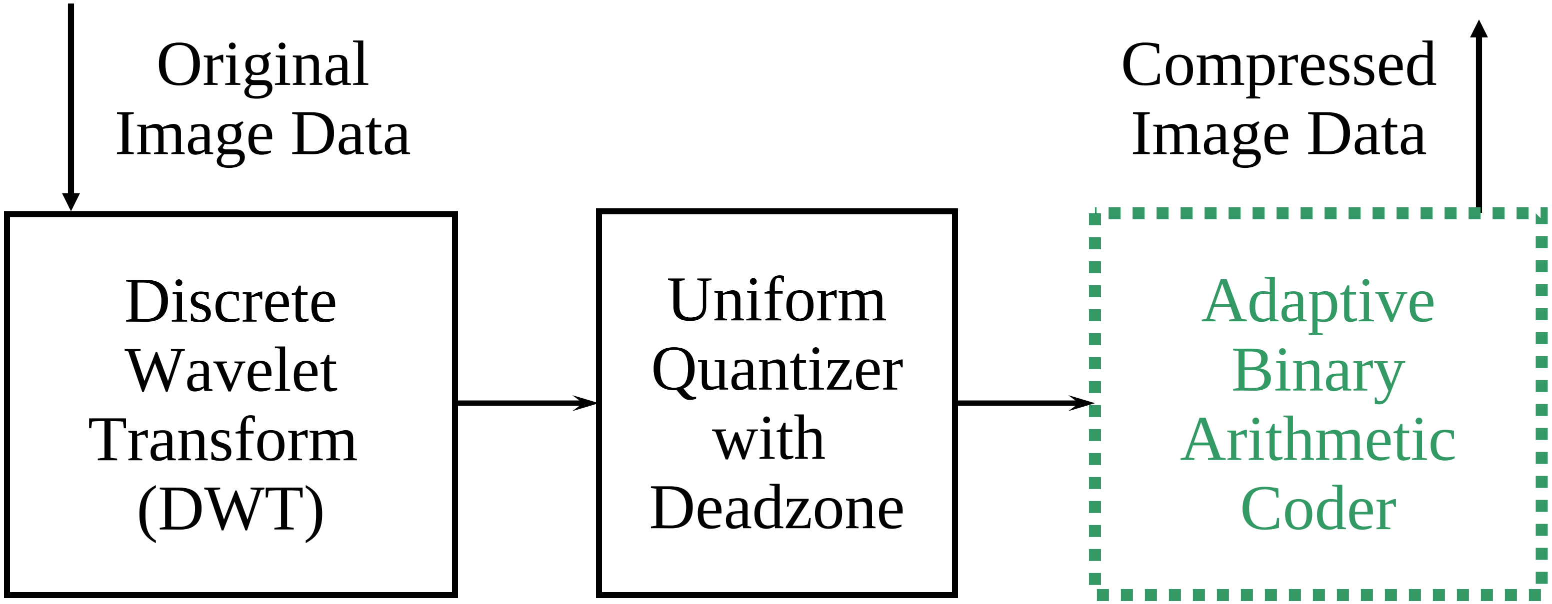
Embedded Quantization by Bit-Plane Coding



Embedded Quantization Example

- Wavelet coefficient = 83
  - Quantizer step size = 3
  - Quantizer index = $\lfloor 83/3 \rfloor$ = 27
  - Quantizer index in binary: 00011011
  - Decoded index after 6 BP's: 000110 = 6
  - Step size with 2 BP's remaining = 12
  - Quantizer index with step size of 12 = 6
  - Dequantized value = (6 + 0.5) x 12 = 78

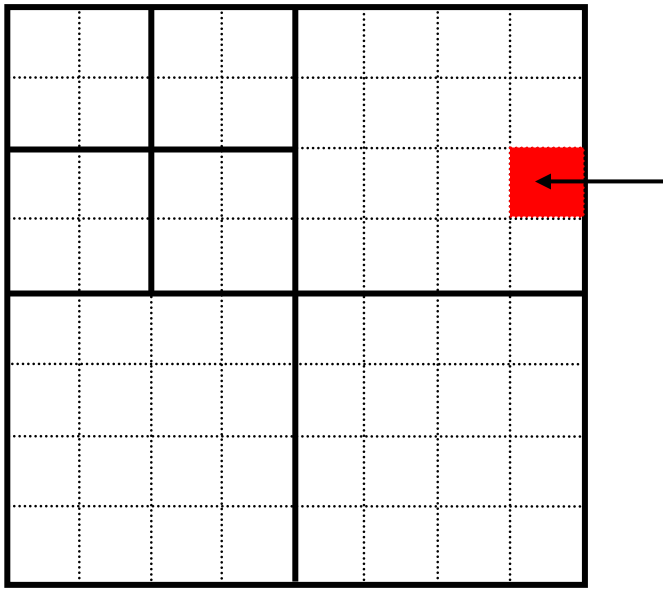
JPEG2000 Part I Encoder



Context-based adaptive binary arithmetic coding is used in JPEG2000 to efficiently compress each individual bit plane.

JPEG2000 Entropy Coder

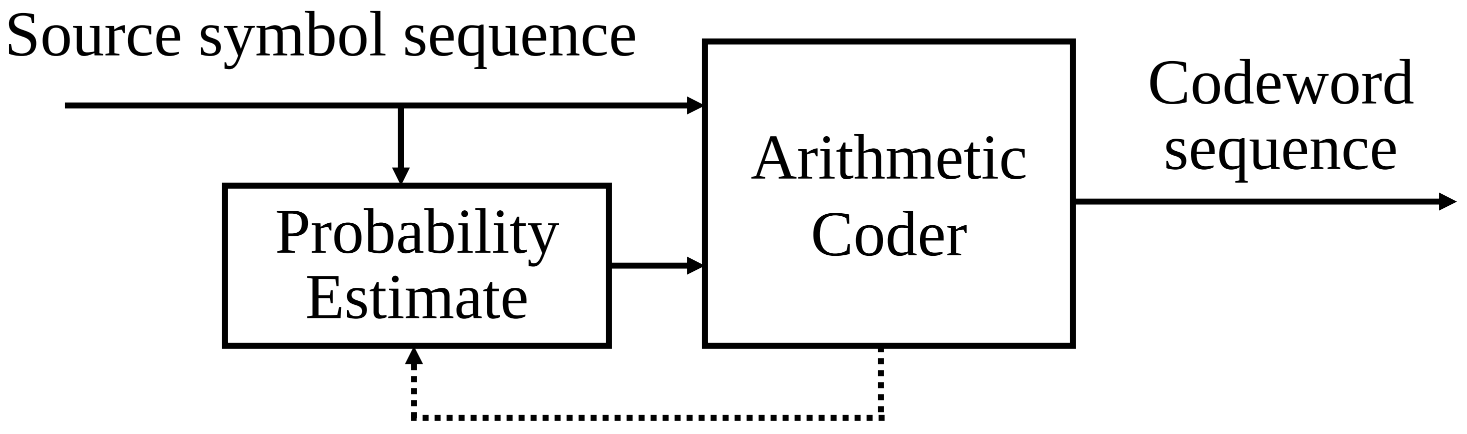
- Each bit plane is further broken down into blocks (e.g., 64 x 64). The blocks are coded independently (i.e., the bit stream for each block can be decoded independent of other data) using three coding passes. The coding progresses from the most significant bit-plane to the least significant bit-plane.



A coding block of a bit plane of a subband

Arithmetic Coding (AC)

- An arithmetic coder accepts at its input the symbols in a source sequence along with their corresponding probability estimates, and produces at its output a code stream with a length equal to the combined ideal codelengths of the input symbols.
- Some implementations of arithmetic coding adaptively update the symbol probability estimate in each context as the symbols get encoded.
- Practical implementations of AC, such as the JBIG/JPEG **QM-Coder** or **MQ-Coder**

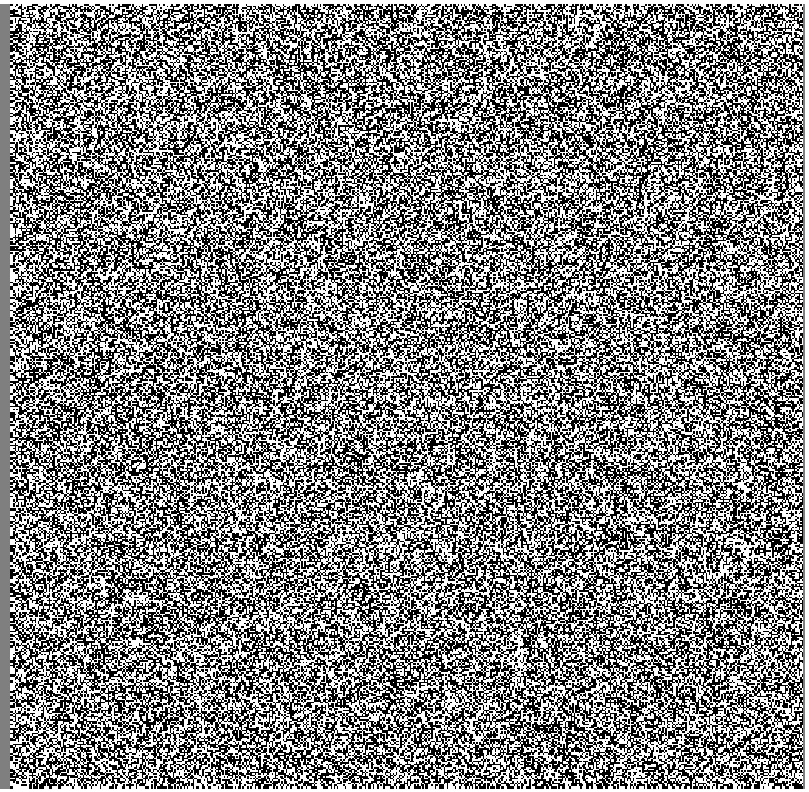


JPEG2000 Entropy Coder

- The binary value of a sample in a block of a bit plane of a subband is coded as a binary symbol with the JBIG2 **MQ-Coder** that is a context-based adaptive arithmetic coder.
- Each bit-plane of each block of a subband is encoded in **three sub bit plane passes** instead of a single pass. The bitstream can be truncated at the end of each pass. This allows for:
  - Optimal embedding, so that the information that results in the most reduction in distortion for the least increase in file size is encoded first.
  - A larger number of bit-stream truncation points to achieve finer SNR scalability.

Example: Lena MSB and LSB Planes

Most significant bit plane

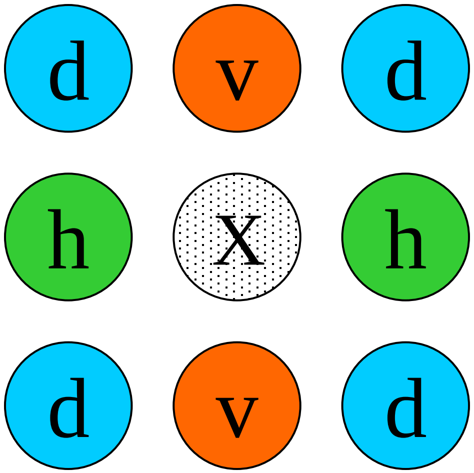


Least significant bit plane

Significance Propagation Pass

- The first pass in a new bit plane is called the **significance propagation pass**. A symbol is encoded if it is insignificant but at least one of its eight-connected neighbors is significant as determined from the previous bit plane and the current bit plane based on coded information up to that point. These locations have the highest probability of becoming significant).

- The probability of the binary value at a given location of a bit-plane of a block of a subband is modeled by a context formed from the significance values of its neighbors.



Refinement and Clean-up Passes

- **Refinement** (REF): Next, the significant coefficients are refined by their bit representation in the current bit-plane.
- **Clean-up**: Finally, all the remaining coefficients in the bit-plane are encoded. (*Note*: the first pass of the MSB bit-plane of a subband is always a clean-up pass).
- The coding for the first and third passes are identical, except for the run coding that is employed in the third pass.
- The maximum number of contexts used in any pass is no more than nine, thus allowing for extremely rapid probability adaptation that decreases the cost of independently coded segments.

Sig. Prop. = 0

Refine = 0

Cleanup = 21

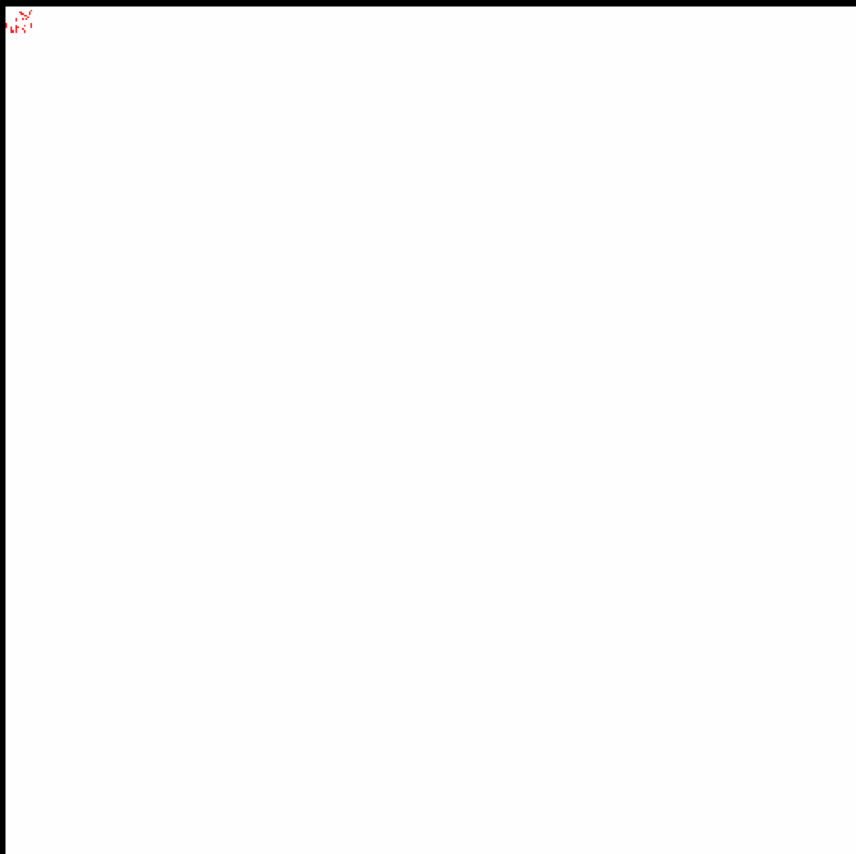
Total Bytes 21

Bit plane **1**

Compression ratio = 12483 : 1

RMSE = 39.69 PSNR = 16.16 db

% refined = 0 % insig. = 99.99



Sig. Prop.	=	18
Refine	=	0
Cleanup	=	24
Total Bytes		42

Bit plane **2**

Compression ratio = 4161 : 1

RMSE = 29.11 PSNR = 18.85 db

% refined = 0.01 % insig. = 99.95

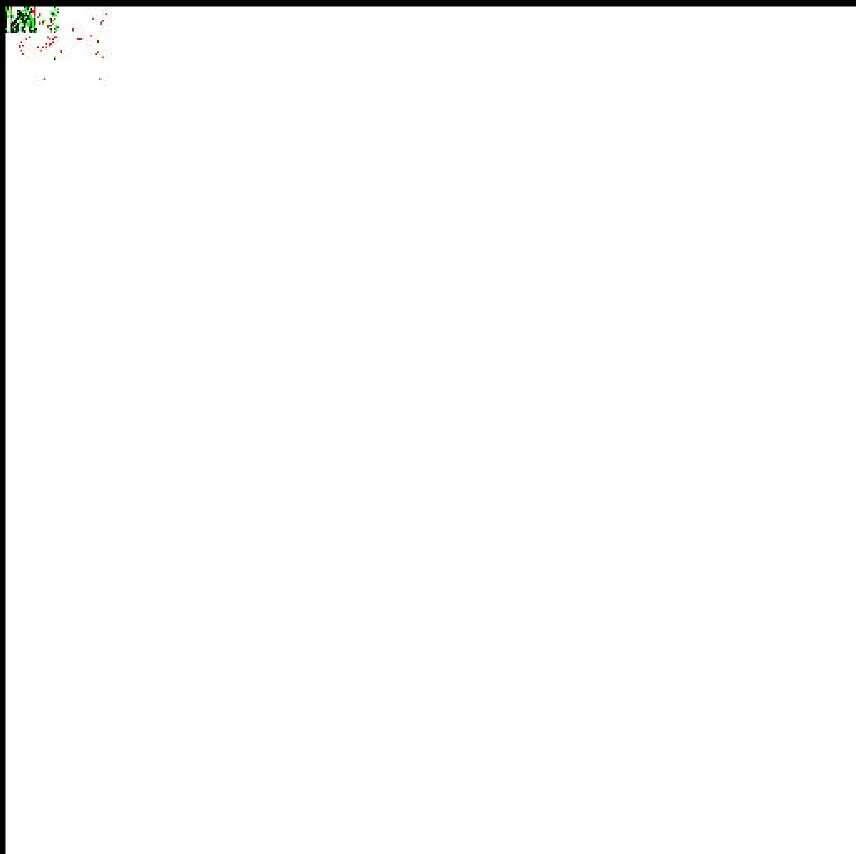
Sig. Prop.	=	38
Refine	=	13
Cleanup	=	57
Total Bytes		108

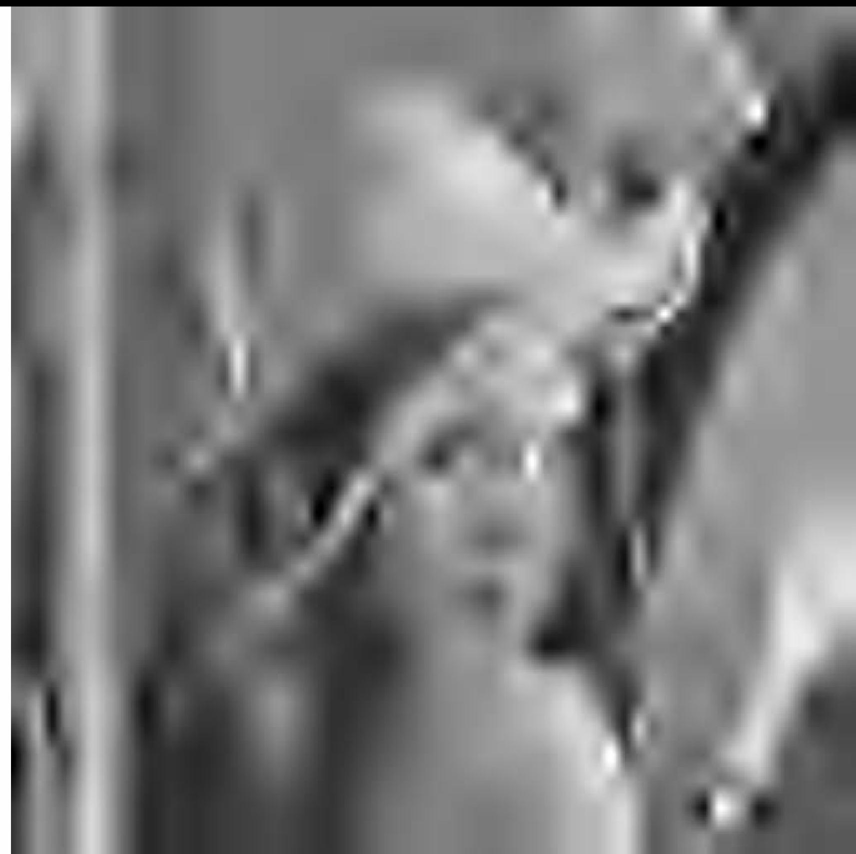
Bit plane **3**

Compression ratio = 1533 : 1

RMSE = 21.59 PSNR = 21.45 db

% refined = 0.05 % insig. = 99.89





Sig. Prop. = 78

Refine = 37

Cleanup = 156

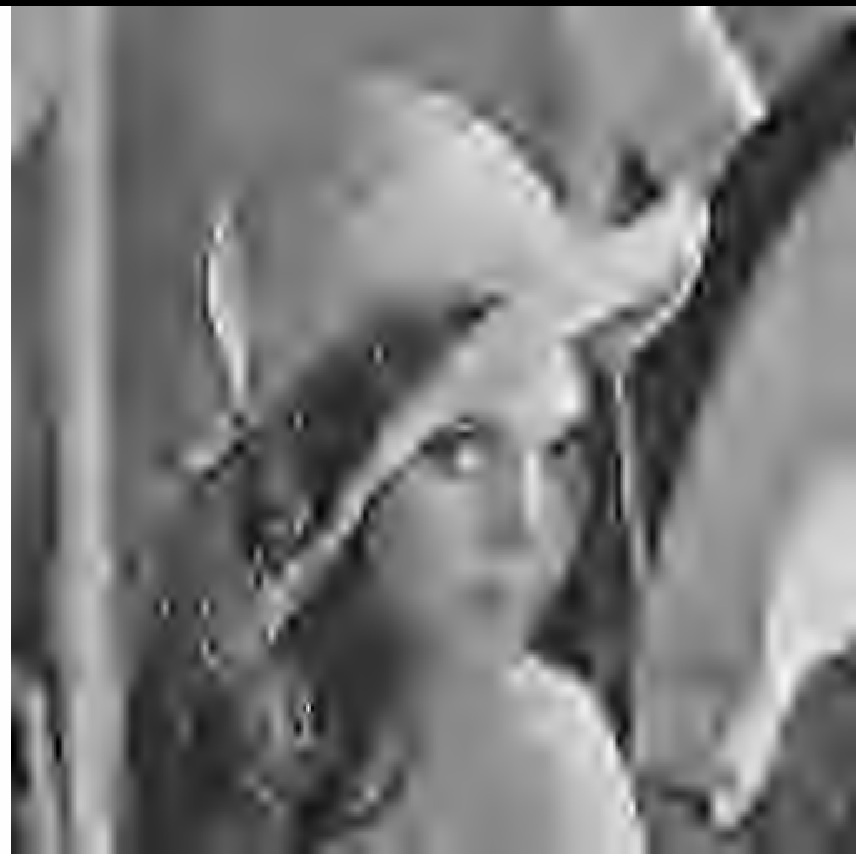
Total Bytes 271

Bit plane **4**

Compression ratio = 593 : 1

RMSE = 16.58 PSNR = 23.74 db

% refined = 0.11 % insig. = 99.77



Sig. Prop.	=	224
Refine	=	73
Cleanup	=	383
Total Bytes		680

Bit plane 5

Compression ratio = 233 : 1

RMSE = 12.11 PSNR = 26.47 db

% refined = 0.23 % insig. = 99.43

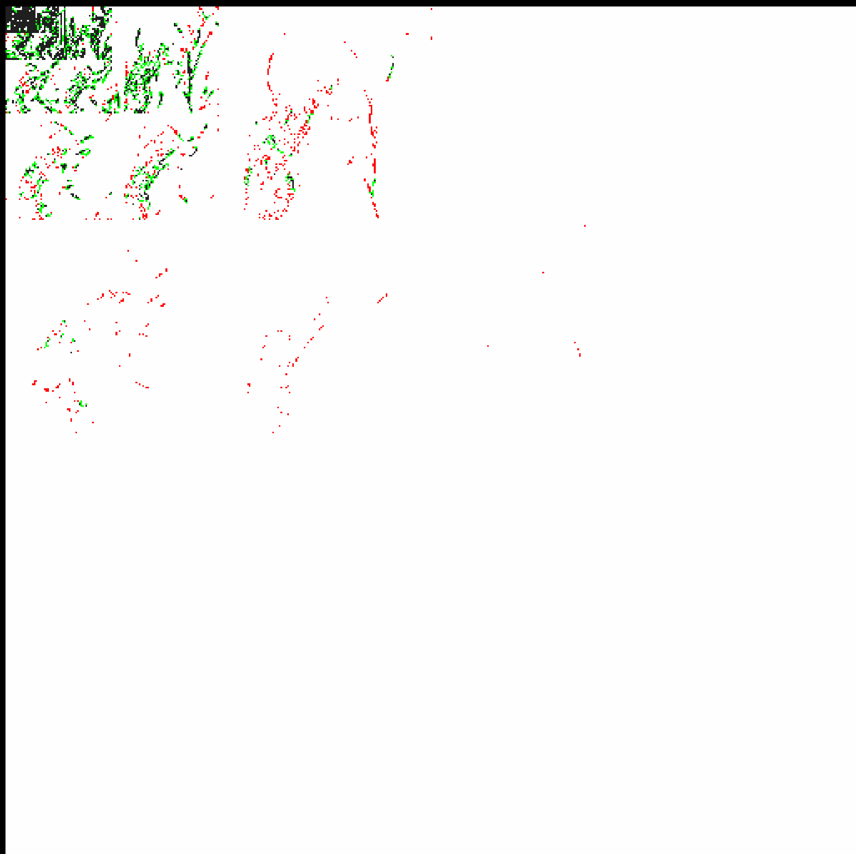
Sig. Prop.	=	551
Refine	=	180
Cleanup	=	748
Total Bytes		1479

Bit plane **6**

Compression ratio = 101 : 1

RMSE = 8.65 PSNR = 29.39 db

% refined = 0.58 % insig. = 98.68



Sig. Prop. = 1243

Refine = 418

Cleanup = 1349

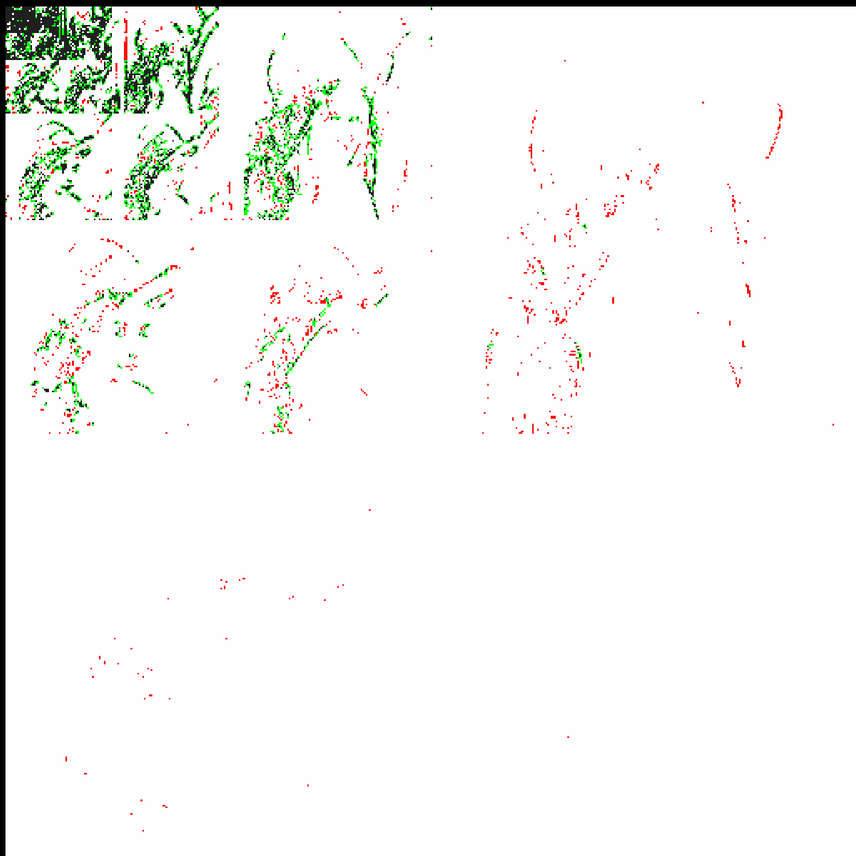
Total Bytes 3010

Bit plane **7**

Compression ratio = 47 : 1

RMSE = 6.02 PSNR = 32.54 db

% refined = 1.32 % insig. = 97.09



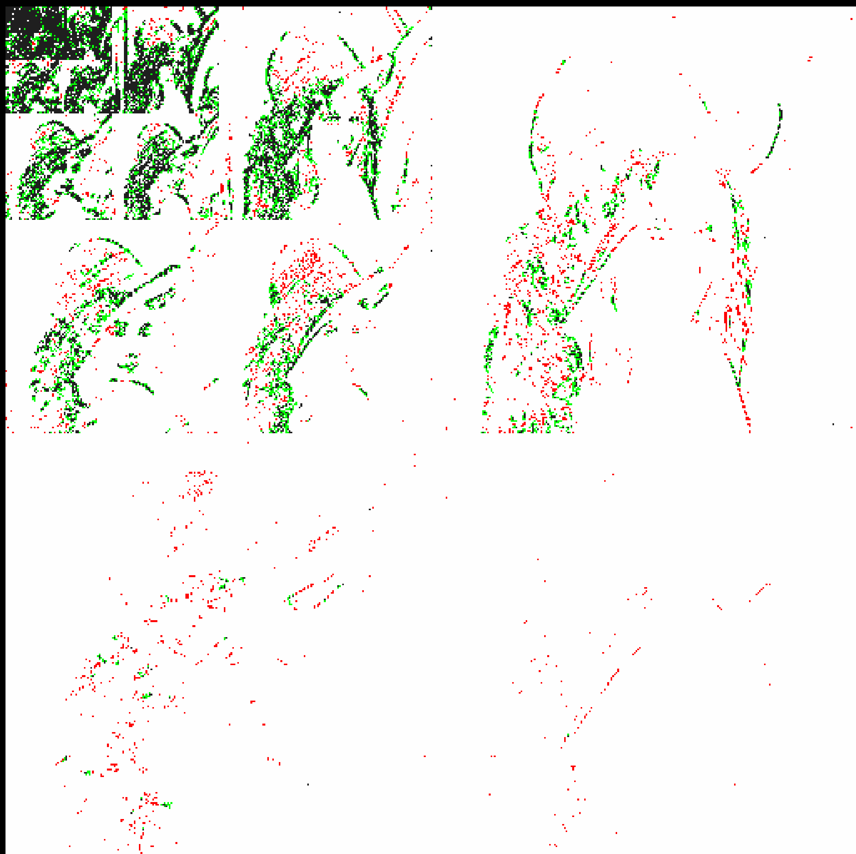
Sig. Prop.	=	2315
Refine	=	932
Cleanup	=	2570
Total Bytes		5817

Bit plane **8**

Compression ratio = 23 : 1

RMSE = 4.18 PSNR = 35.70 db

% refined = 2.91 % insig. = 93.99



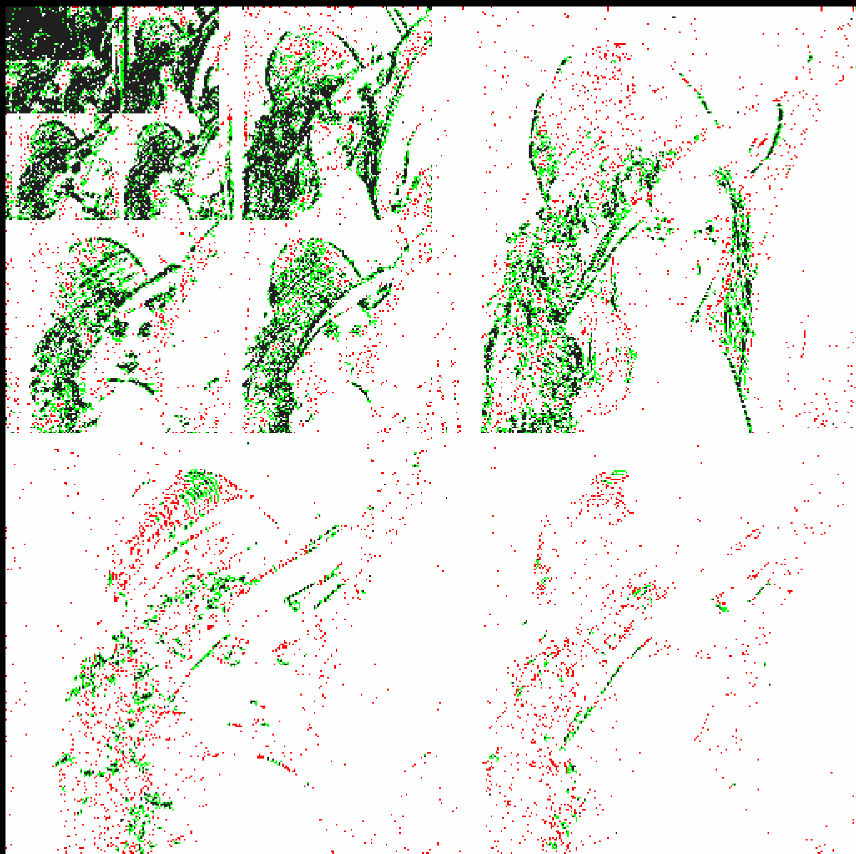
Sig. Prop.	=	4593
Refine	=	1925
Cleanup	=	5465
Total Bytes		11983

Bit plane 9

Compression ratio = 11.2 : 1

RMSE = 2.90 PSNR = 38.87 db

% refined = 6.01 % insig. = 87.66



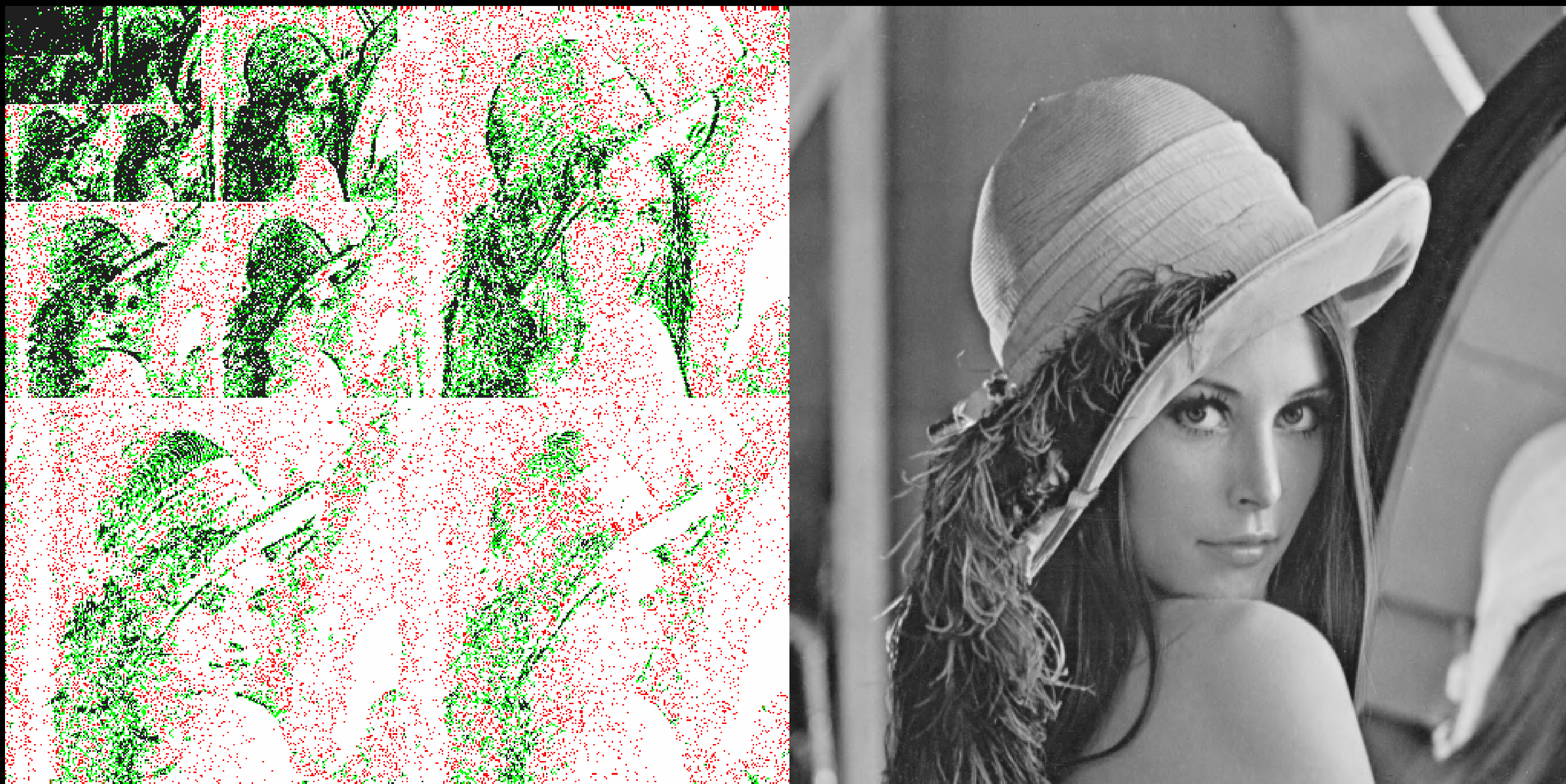
Sig. Prop.	=	10720
Refine	=	3917
Cleanup	=	12779
Total Bytes		27416

Bit plane **10**

Compression ratio = 5.16 : 1

RMSE = 1.78 PSNR = 43.12 db

% refined = 12.34 % insig. = 82.00



Sig. Prop. = 25421
Refine = 8808
Cleanup = 5438

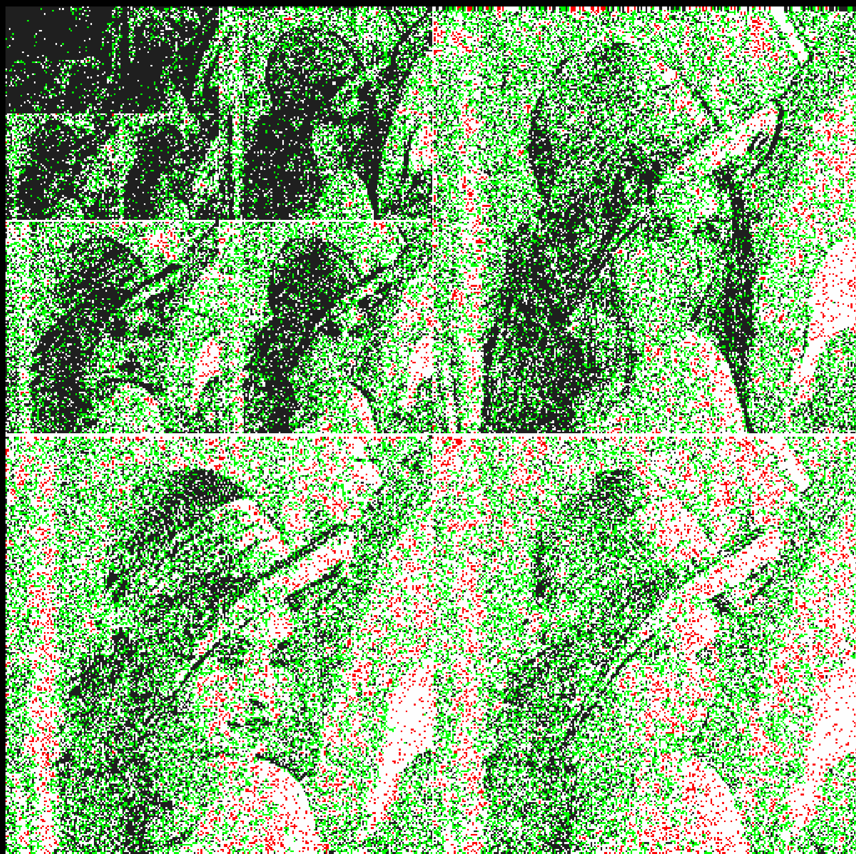
Total Bytes 39667

Bit plane **11**

Compression ratio = 2.90 : 1

RMSE = 0.90 PSNR = 49.00 db

% refined = 28.12 % insig. = 46.80



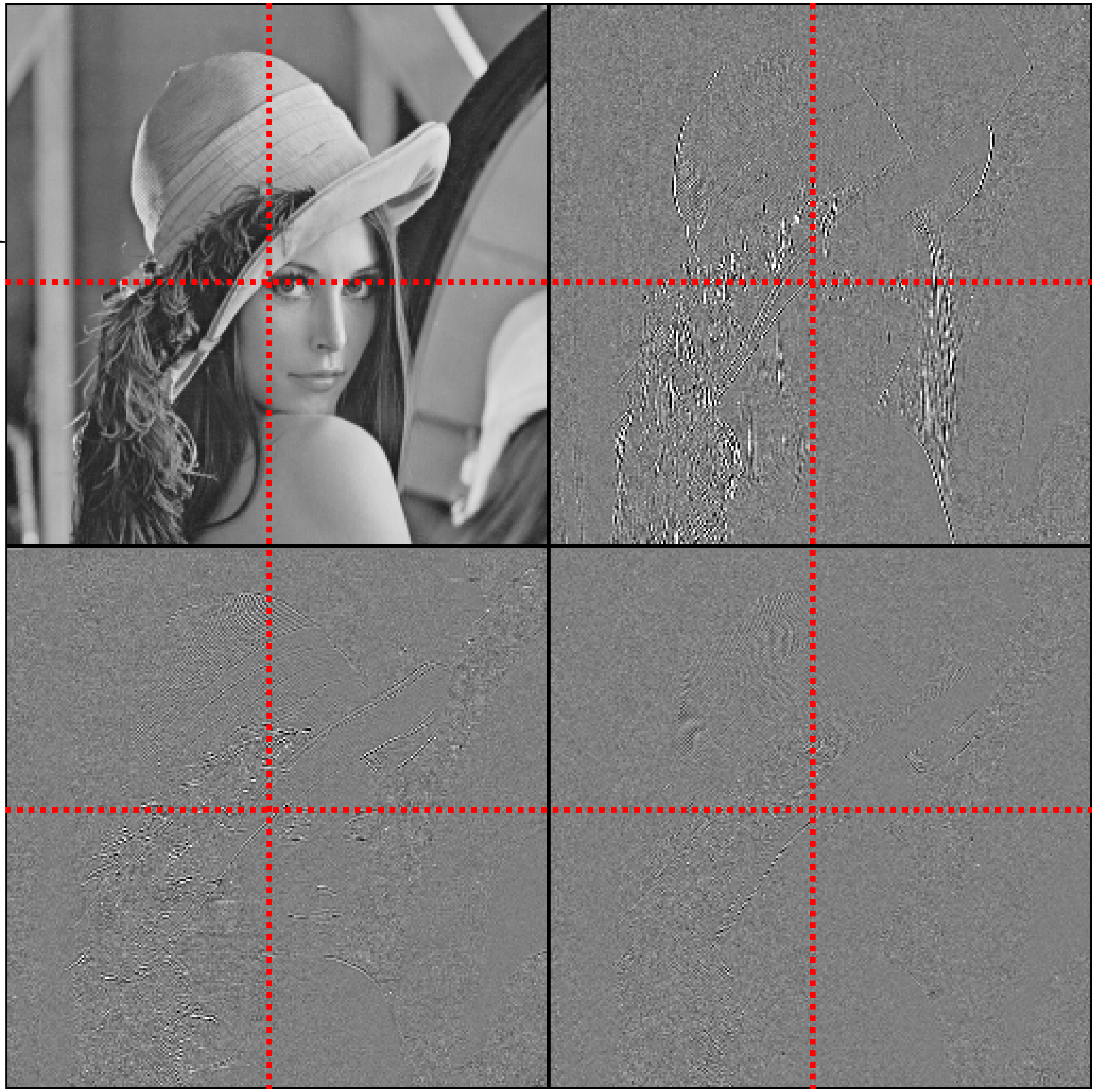
Benefits of the JPEG2000 Entropy Coder

- Bit plane representation of data allows for embedded bit stream
- Region of interest coding (ROI) coding can be performed by prioritizing the coding of the ROI bit plane information
- Arithmetic coding allows for efficient compression of sparse binary data
- Context modeling allows for efficient compression of binary correlated data
- Packetized information allows for improved error resilience

Example of Bit Plane Reordering

Subband 1

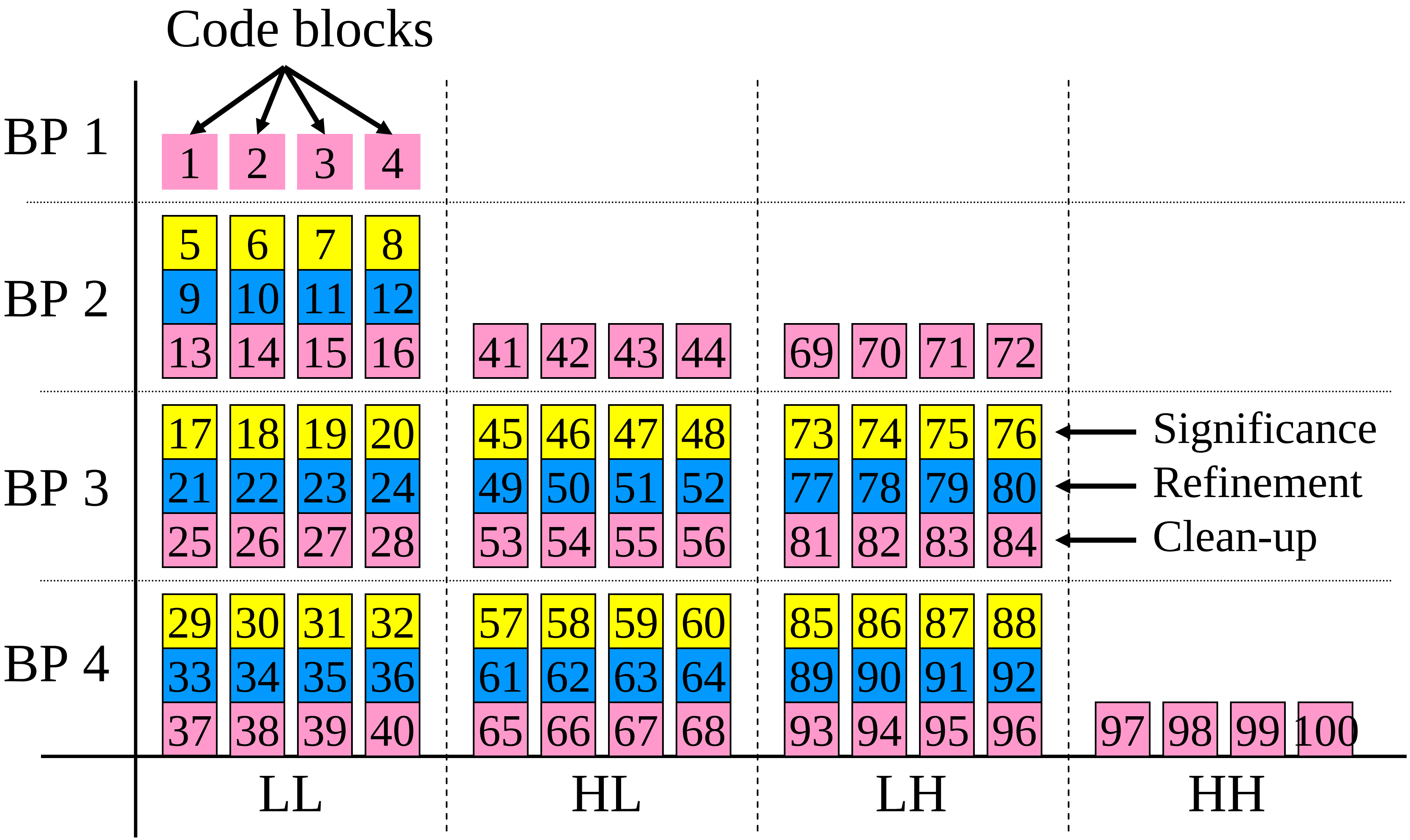
Subband 2



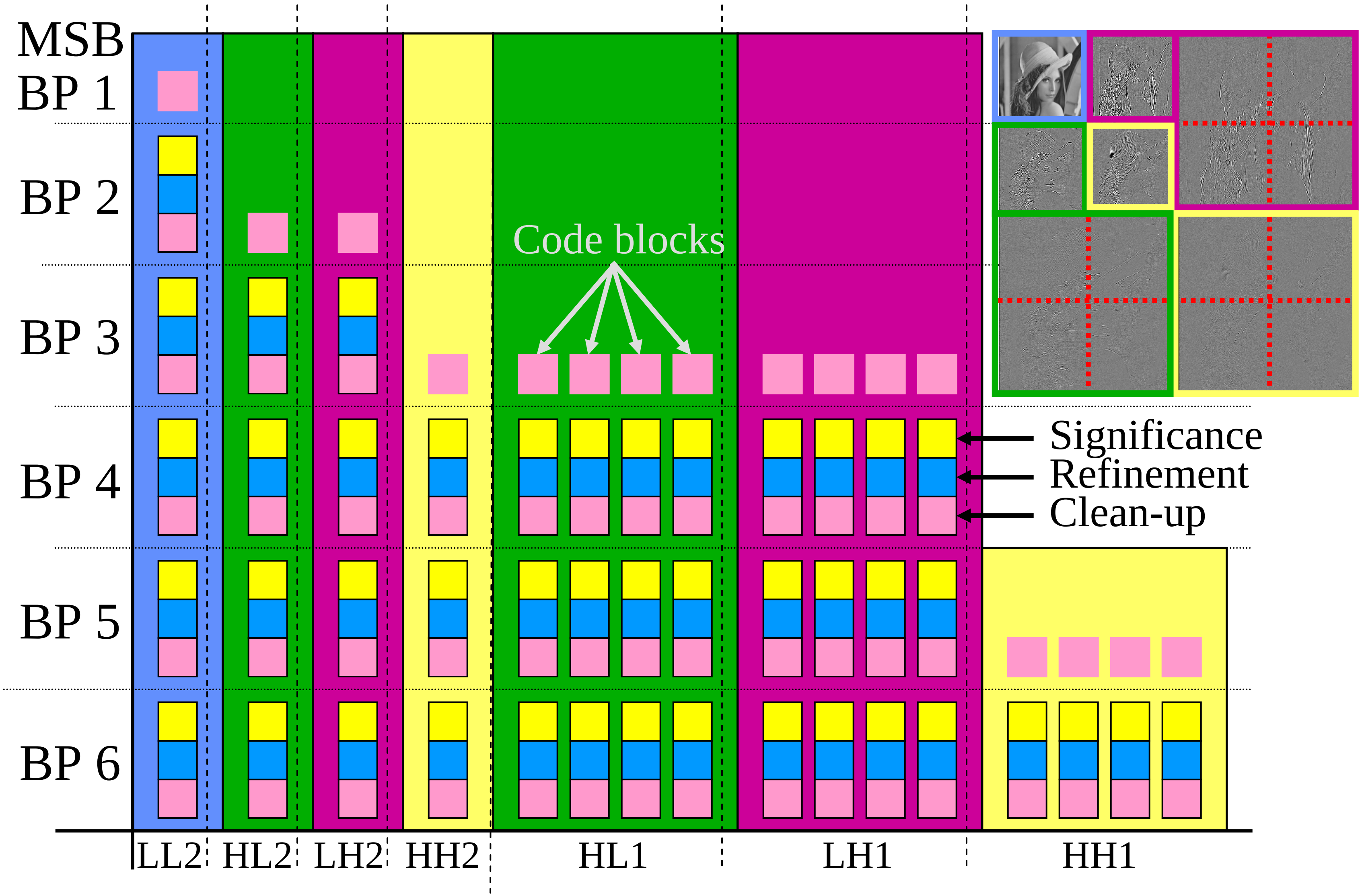
Subband 3

Subband 4

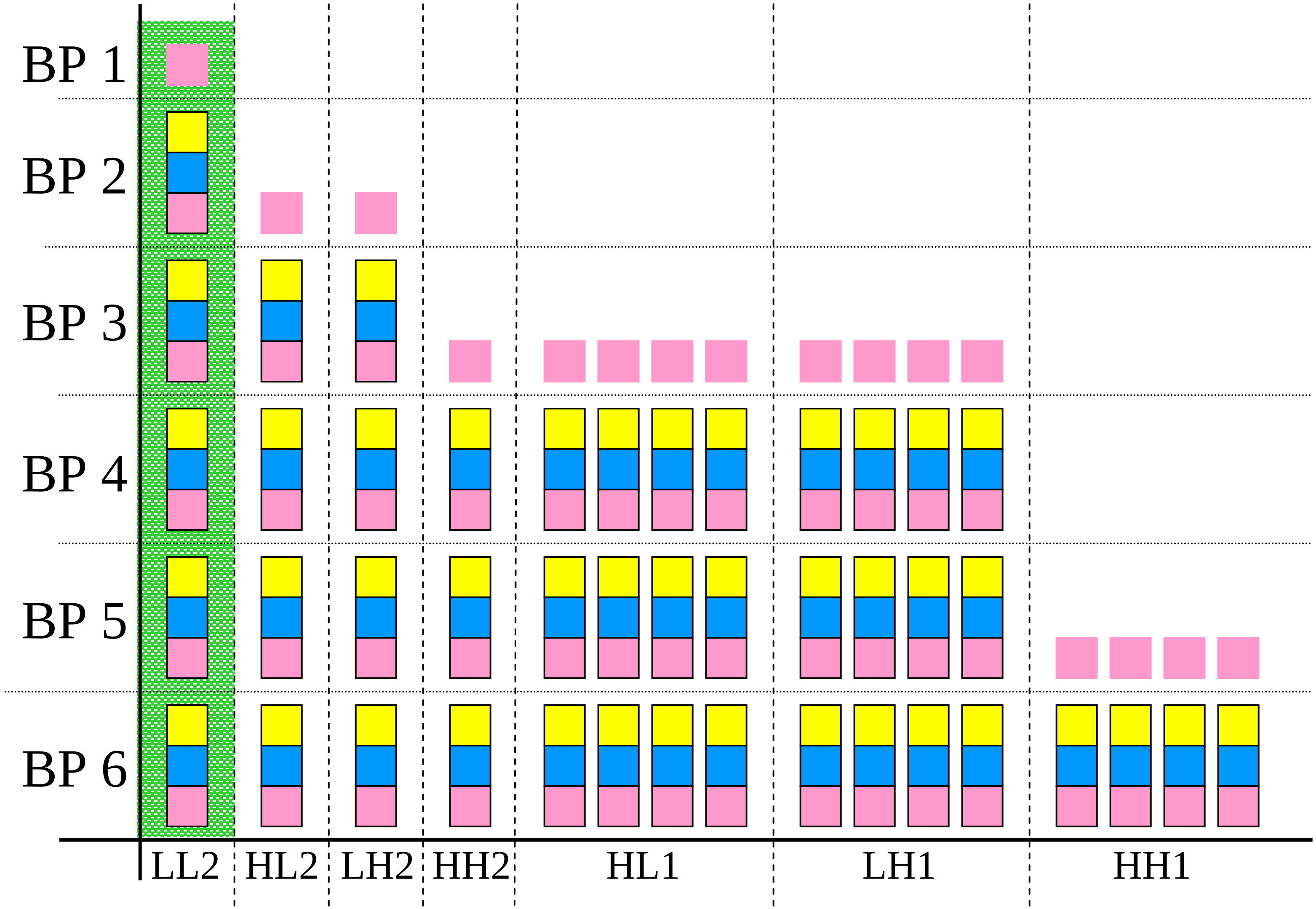
Example of Bit-Plane Reordering



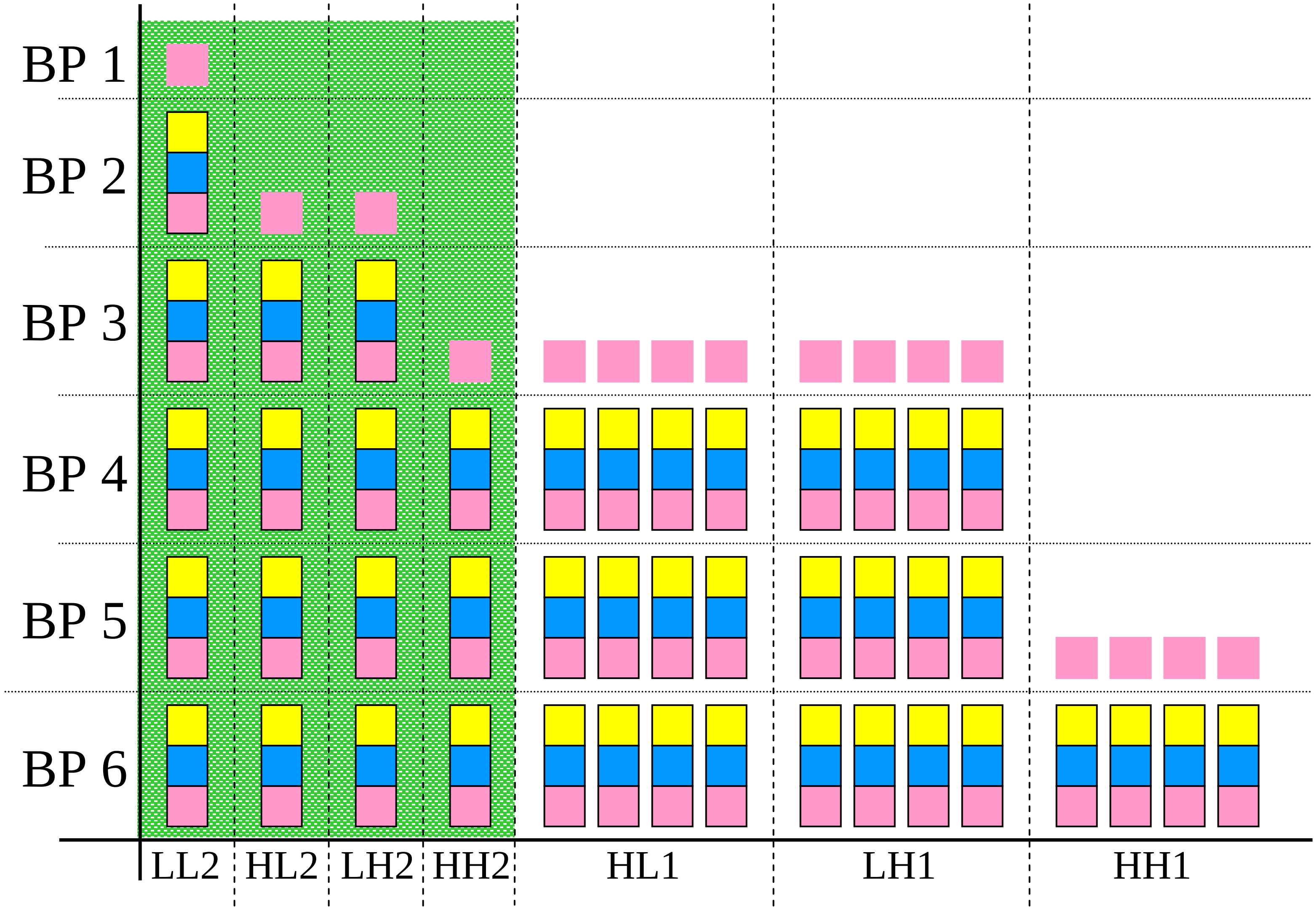
Example of Bit-Plane Data Ordering



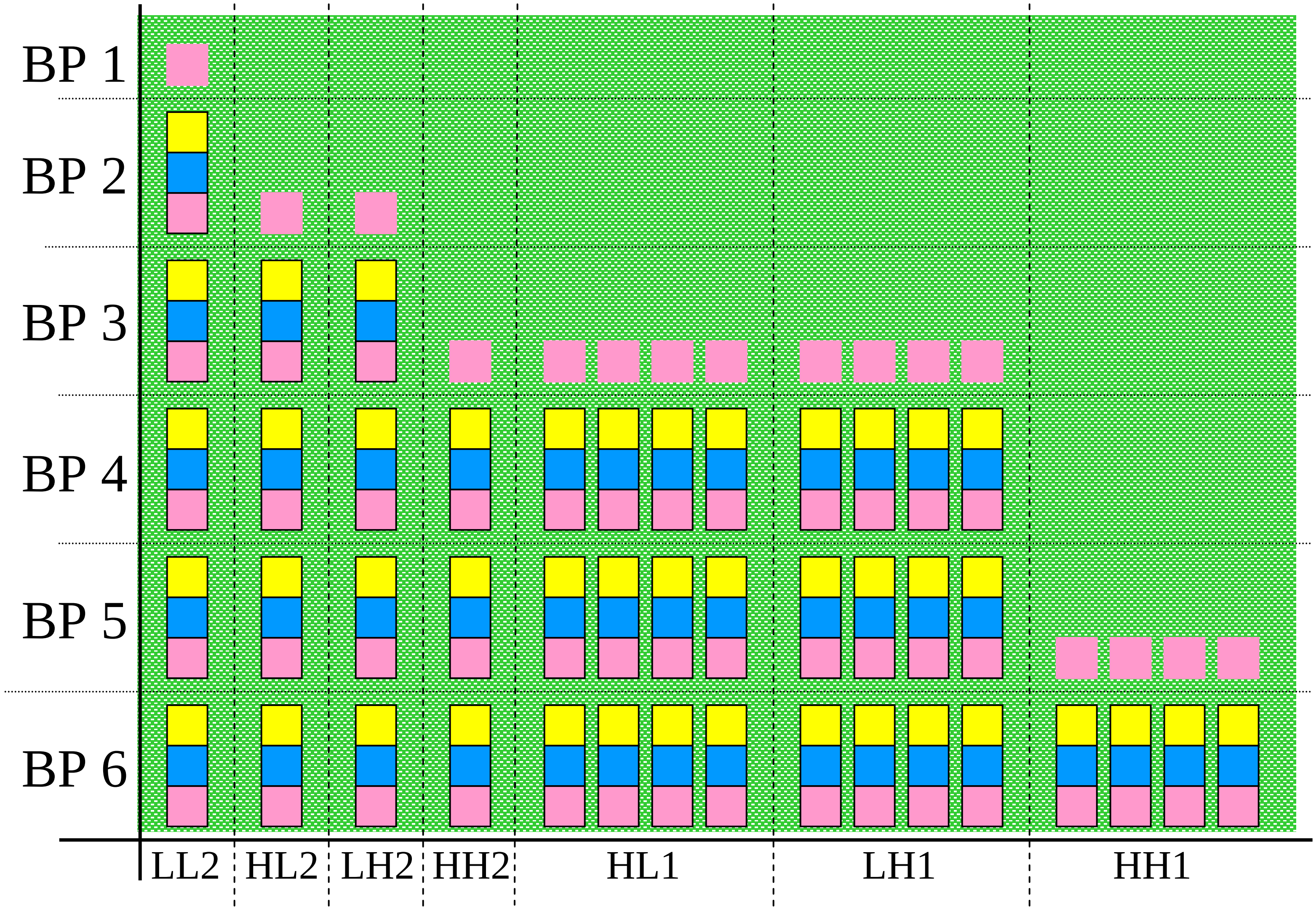
Lowest Resolution, Highest Quality



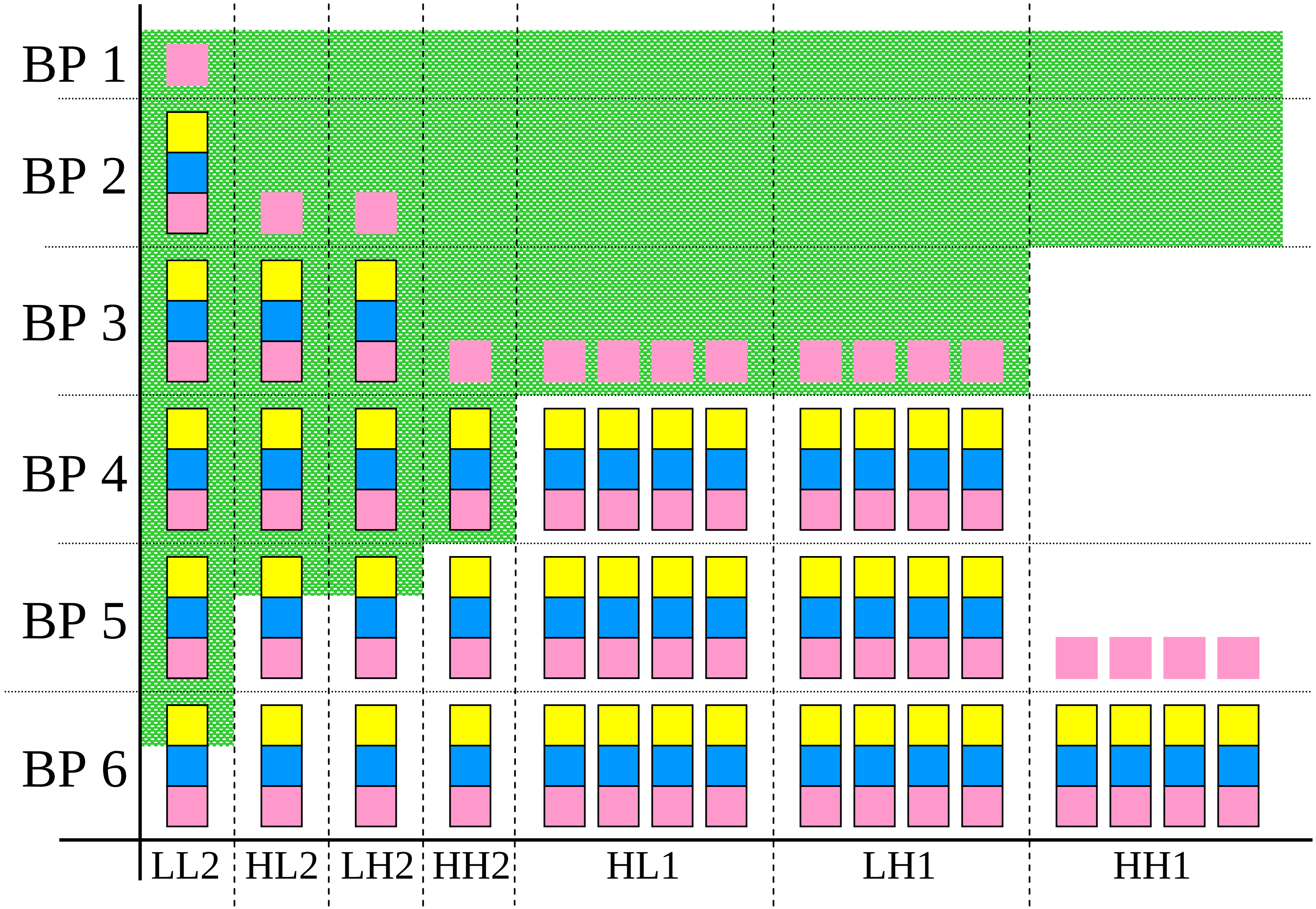
Medium Resolution, Highest Quality



Highest Resolution, Highest Quality



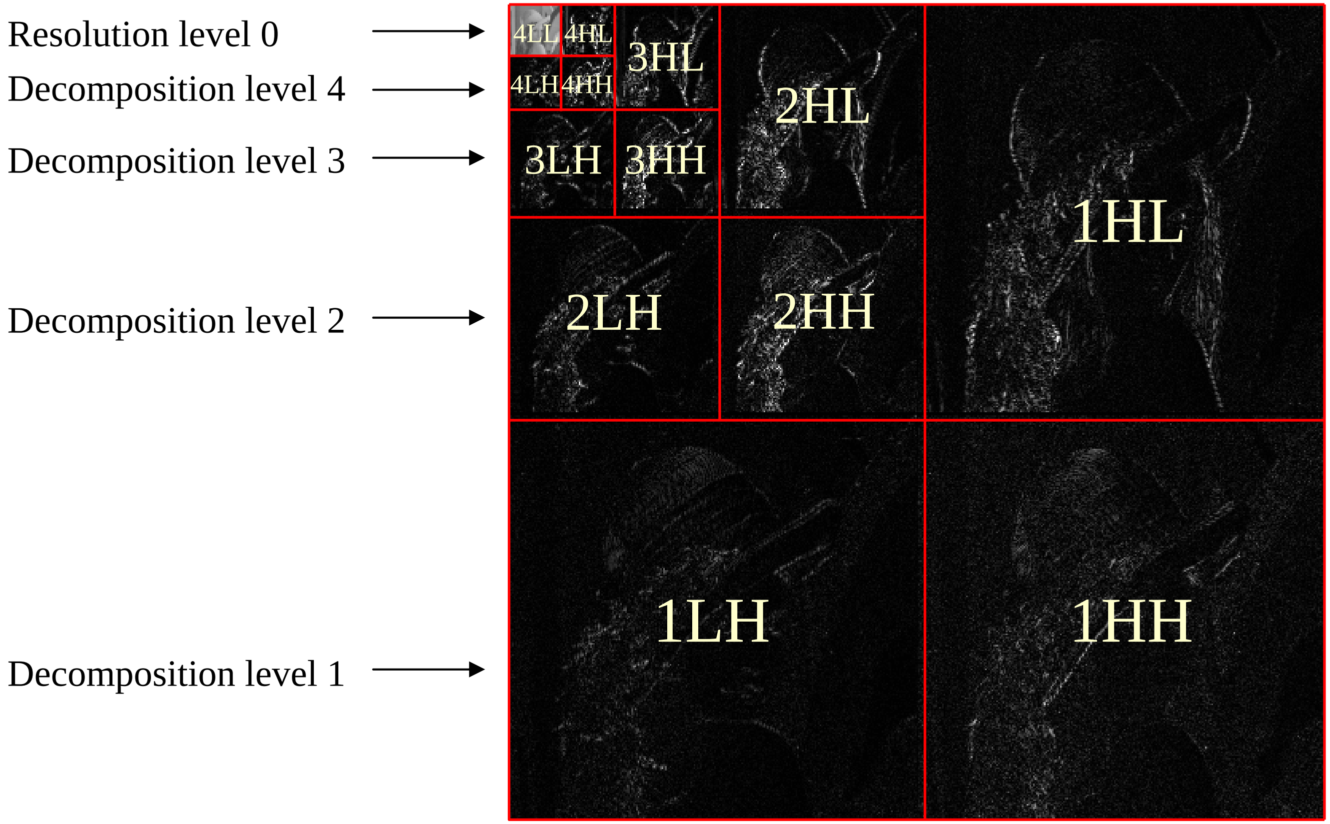
Highest Resolution, Target Visual Quality



JPEG2000 Bitstream Syntax

- **Precinct**: Each subband is divided into non-overlapping rectangles of equal size (except maybe at the boundaries where the size can be different) called precincts. Precincts provide some level of spatial locality in the bit stream and their boundaries are aligned with code blocks. Their size can vary for each tile, (color) component, and resolution.
- **Packet**: Consists of the compressed bit streams associated with a certain number of sub-bit planes from each codeblock in a precinct. Packets serve as one quality increment for one resolution level at one spatial location.
- **Layer**: is a collection of packets, one from each precinct at each resolution. It can be interpreted as one quality increment for the entire image at full resolution.

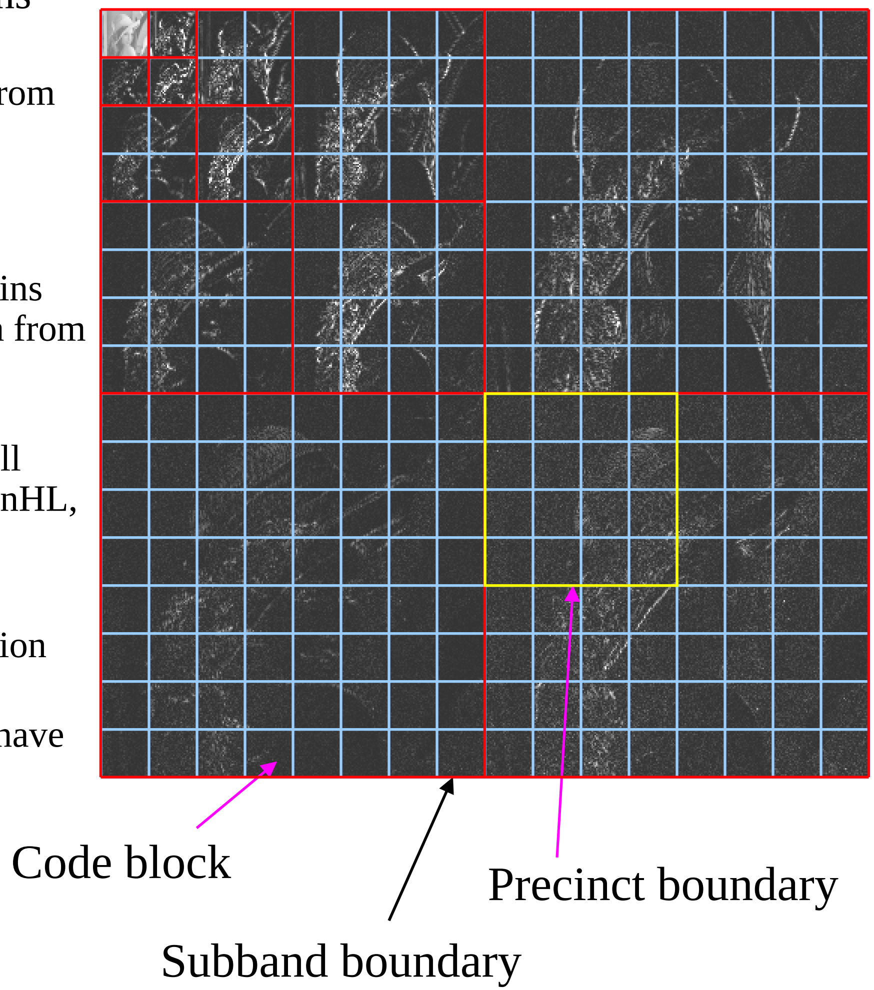
Resolution & Decomposition Levels



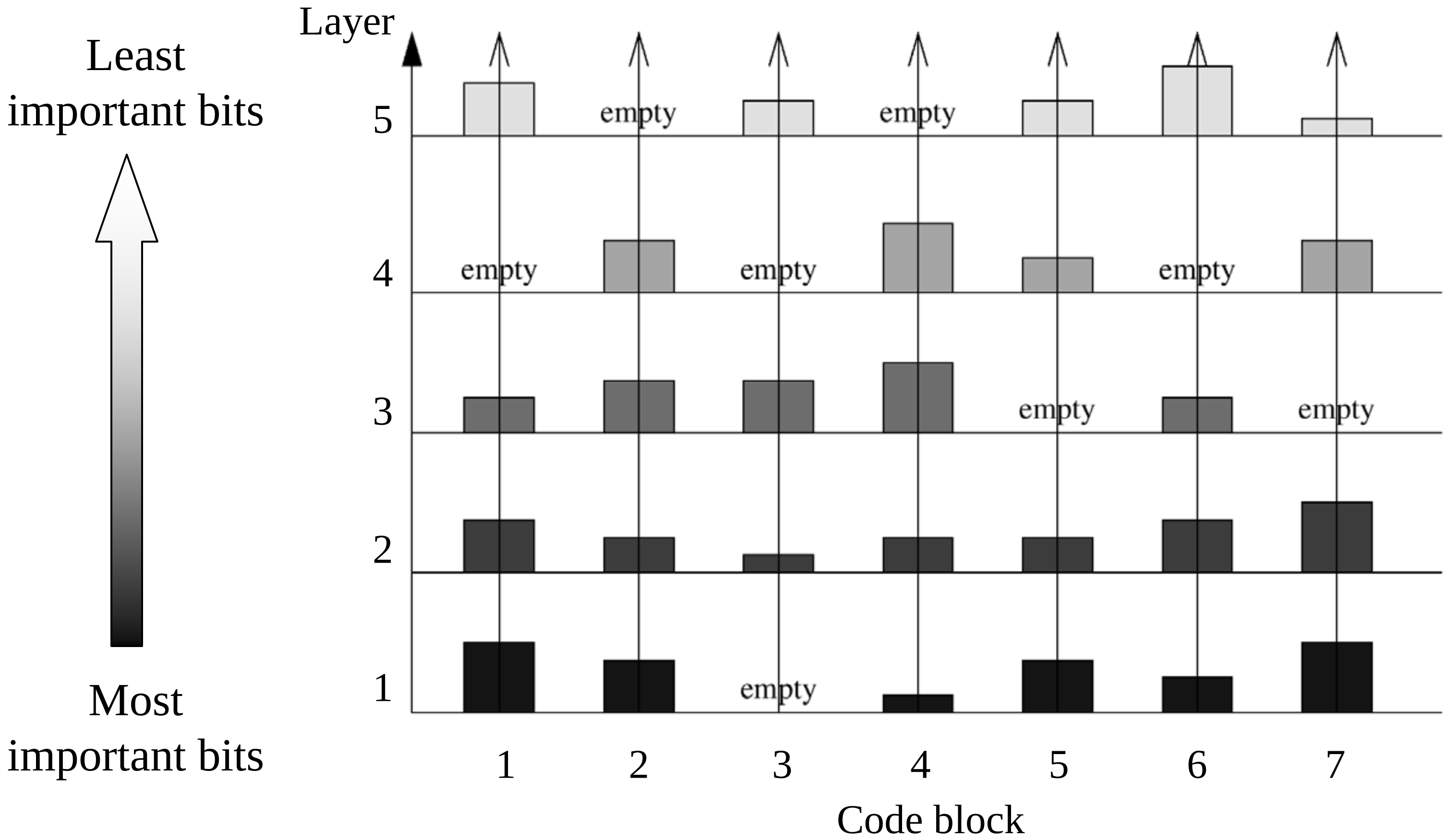
- Decomposition level
  - A set of HL, LH, and HH subbands. For the last level of decomposition (N_L), the LL subband is included. Decomposition levels run from 1 to N_L.
- Resolution level
  - Related to decomposition angle by $n = N_L - r$, where n is a decomposition level and r is a resolution level. Reconstruction levels run from N_L to 0.

Code Blocks, Packets & Precincts

- Code blocks divide subbands into regions that may be extracted independently
  - A rectangular grouping of coefficients from the same subband
  - Code blocks do not “scale” with levels
- Packets
  - Part of the compressed bitstream. Contains packet header & compressed image data from one layer of one precinct of one decomposition level
  - Contains compressed image data from all code blocks of a given resolution level (nHL, nLH, and nHH subbands)
- Precinct
  - Rectangular region within a decomposition level used to limit the size of packets
  - Subbands with more than one precinct, have multiple packets in a given layer.



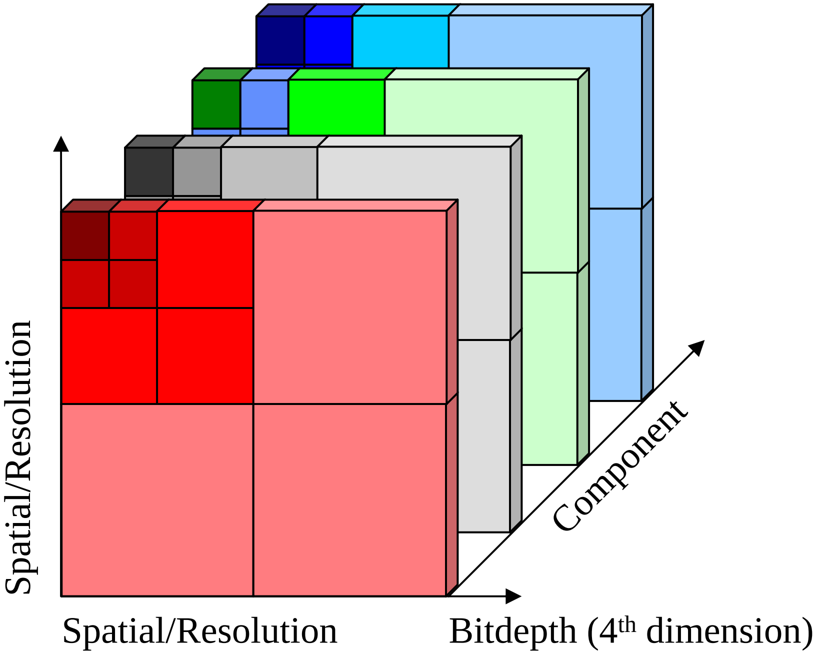
Layer Formation



- Layer
  - An arbitrary collection of compressed image data from coding passes of one or more code blocks. Typically a layer represents an improvement in image quality.
  - Packet headers describe the contribution of each code block in each layer

JPEG 2000 Progression Types

- After wavelet processing, we have a four dimensional cube of data
  - Spatial/Resolution (two)
  - Component
  - Bitdepth
- JPEG 2000 allows progression along four dimensions
  - Layer (L)
  - Resolution (R)
  - Component (C)
  - Precinct or position (P)
- These are roughly equivalent as follows
  - Resolution & Precinct ⇔ Spatial/Resolution
  - Component ⇔ Component
  - Layer ⇔ Bitdepth



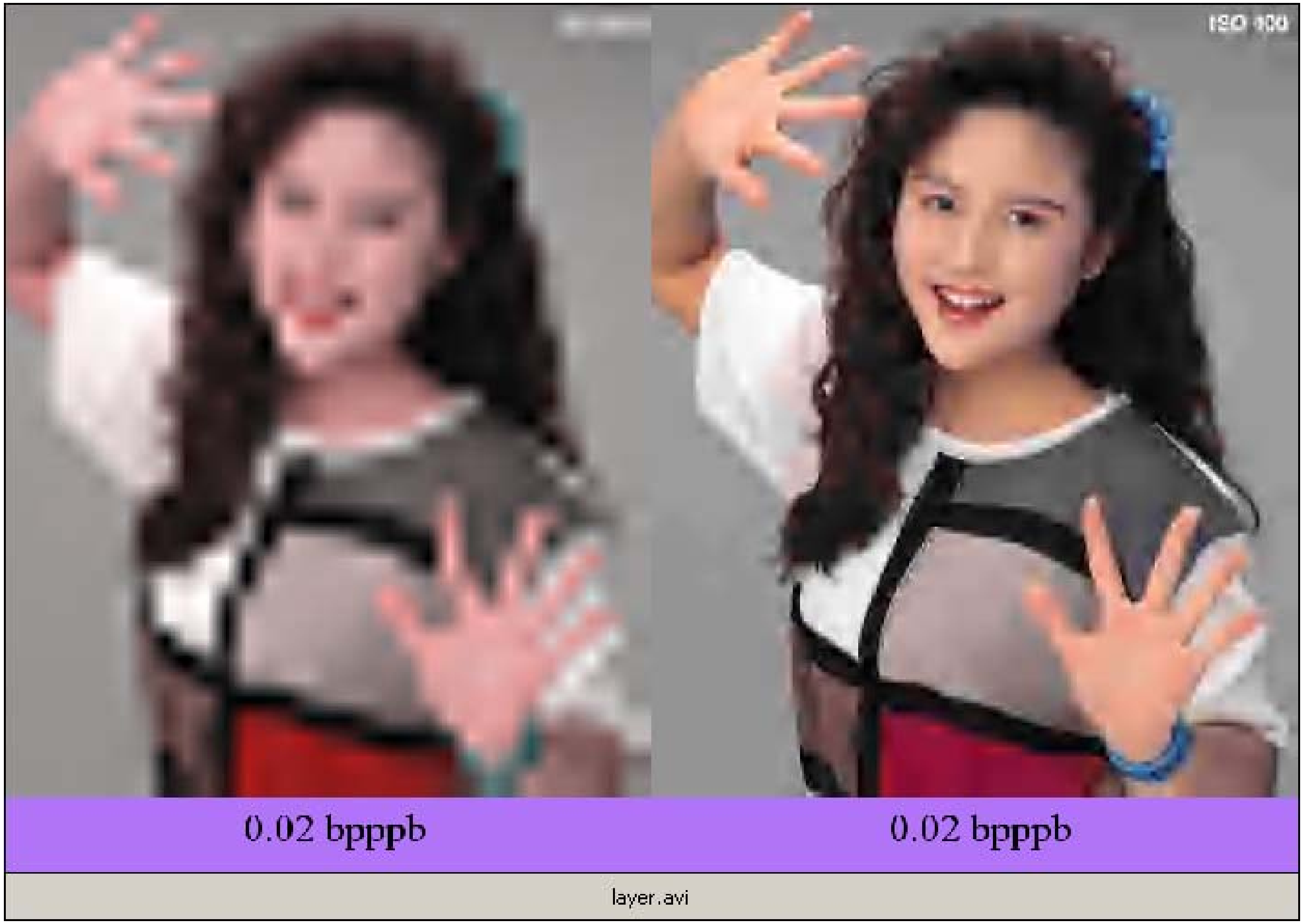
Wavelet processed components

JPEG 2000 Progression Types

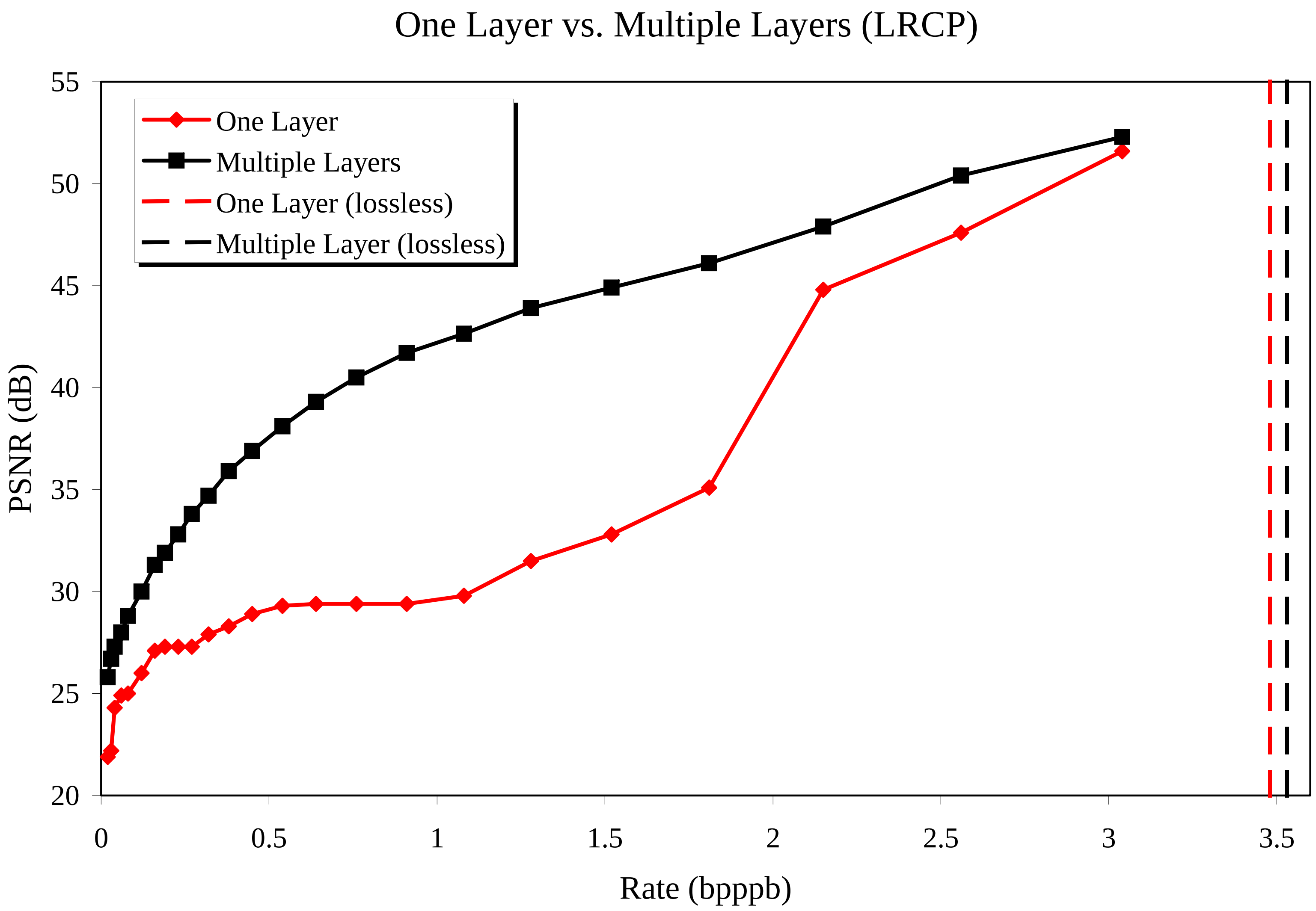
- There are five allowed progression types
  - LRCP, RLCP, RPCL, PCRL, CPRL
  - View progression as four nested loops, read left to right
- LRCP
  - Progression by SNR. Best full size image (SNR)
  - Loop over all layers
    - Loop over all resolution levels
      - Loop over all components
        - Loop over all precinct (positions)
- RLCP
  - Progression by resolution. Improving image quality for a fixed resolution
- Note that "C" (component) is in an inner loop in all progressions except CPRL

Effects of Layering

Single Layer Multiple Layers



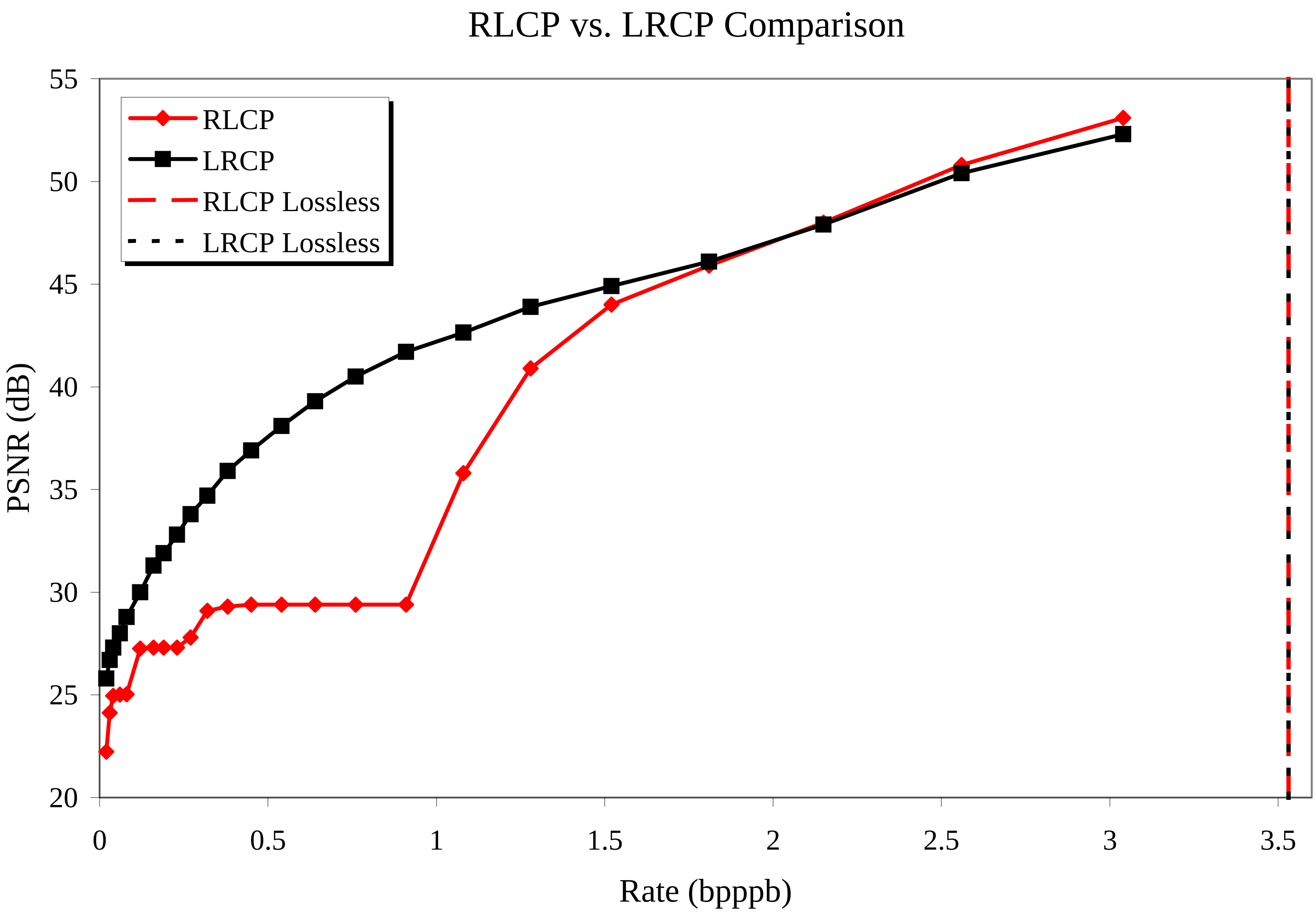
Effects of Layering



Effects of Progression

RLCP Progression LRCP Progression

Effects of Progression



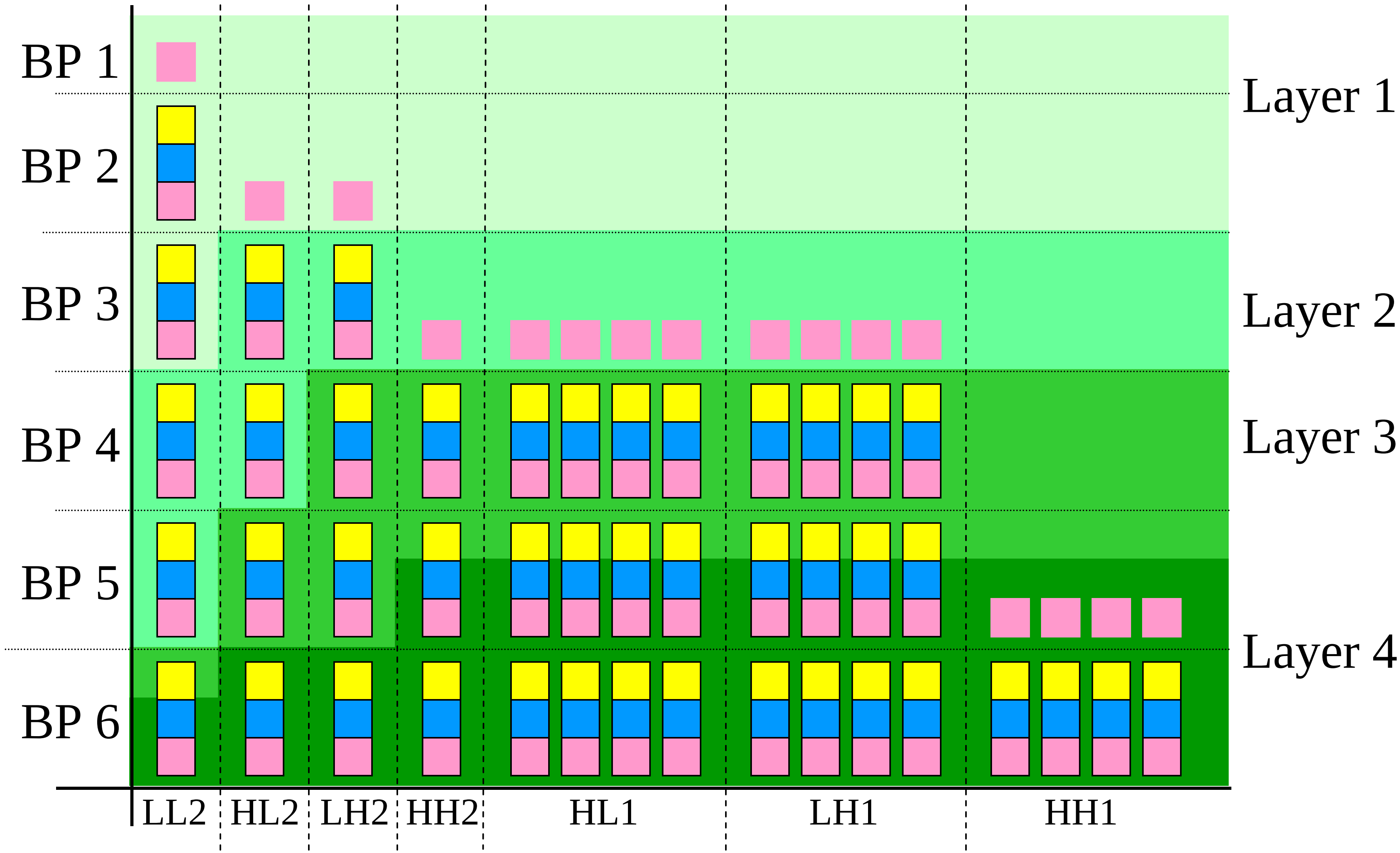
Tiles

- Tiles are independently coded sub images. Nothing crosses tile boundaries
  - Wavelet
  - Entropy coding
  - Layers
  - Progressions
- Tiles may be broken into tile parts. Tile parts from different tiles can be interspersed in a codestream
  - Only mechanism available to achieve “tile progression”
- In general, need to parse data out of tiles to achieve a different image quality
  - If all tiles are compressed at 2.0 bpp and you want 1.0 bpp, then need to go into each tile and get the 1.0 bpp

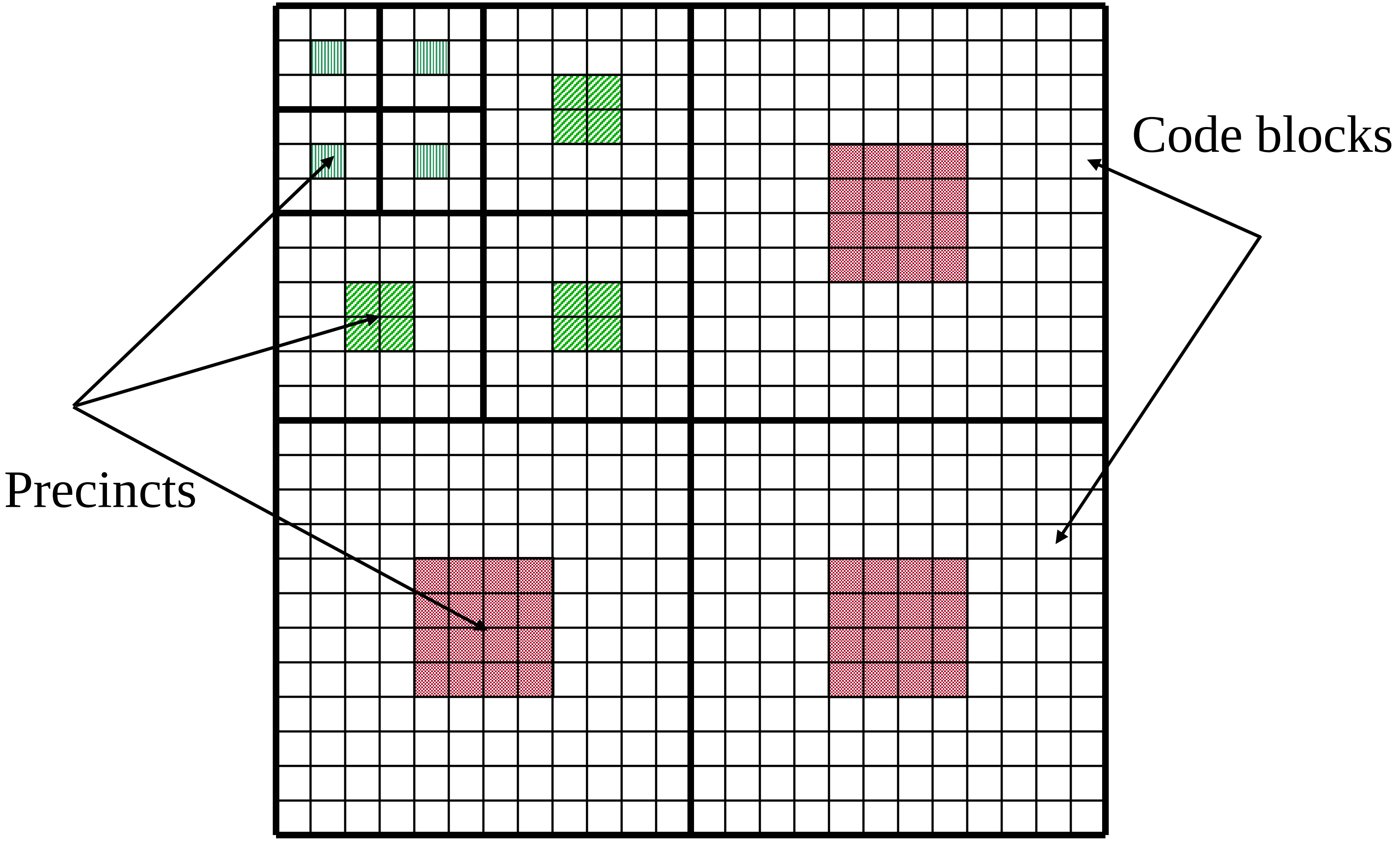
Tiled Image Parsing

No Parsing Performed | Parsing Performed

Bitstream Ordering with Four Quality Layers



Example: Precincts and Codeblocks

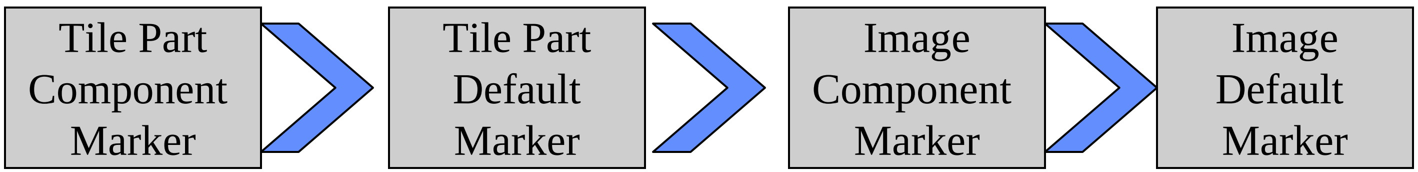


JPEG 2000 Codestream Structure

- Delimiting markers
  - These are markers that are used in the start of most major sections of codestream and the very end.
  - Start of codestream (SOC), start of tile part (SOT), start of data (SOD), and end of codestream (EOC)
- Fixed information marker segments
  - This marker includes information that is required to properly decode the image.
  - Size Marker (SIZ) which includes reference grid size, tile size, resolution/sampling (relative to grid), image and tile offsets (into the grid), number of components, and component precision (data type and bit depth)

JPEG 2000 Codestream Structure

- Functional Marker Segments
  - These marker define the parameters used in the compression of a tile or an image. The order or precedence of the markers are



  - Default markers (can be used in image header or tile header)
    - coding style default (COD), quantization default (QCD), region of interest (RGN), progression order changes (POC)
  - Component Markers
    - coding style component(COC), quantization component (QCC)

JPEG 2000 Codestream Structure

- Functional Markers
  - Coding style default and coding style component include information on coding style, number of decompositions, code-block size, code-block style, wavelet transformation, precinct style
  - Quantization default and quantization component include information on the quantization for the derived, expounded or no quantization
  - Region of interest marker includes the location of the ROI
  - Progression order changes describes the bounds and progression order for any progression other than that in the COD marker in the main image header

JPEG 2000 Codestream Structure

- Pointer Marker Segments
  - Pointer markers are used for quick access to data that is required for decompression of a given location, resolution quality, or component.
  - All of the pointer segments define lengths of segments which allow fast rearranging of data and pointer markers
  - Main Header Markers
    - Tile part lengths (TLM), packet lengths main header (PLM), packed packet headers (PPM) (main header).
  - Tile part header
    - Packet length tile-part header (PLT), packed packet headers (tile part header) (PPT).

JPEG 2000 Codestream Structure

- In Bit stream markers
  - Start of packet (SOP) and end of packet (EOP) markers are used to isolate a given packet in a noisy environment
- Information marker segments
  - Component registration (CRG) marker is used to register components if the components do not have the same sampling to the reference grid
  - The comment marker (COM) is an open style marker that allows for unstructured data

JPEG 2000 Codestream Structure

- The J2K codestream starts with the main header, followed by tile-part header(s), and bitstream(s) and ends with an EOC marker
  - The main header starts with SOC and SIZ markers, then followed (in any order) by COD and QCD markers and possibly QCC, RGN, POC, PPM, TLM, PLM, CRG, COM
  - The tile part header starts with SOT marker and finishes with SOD and can contain (in any order) COD, COC, QCD, QCC, RGN, POC, PPT, PLT, COM
  - The bitstream may contain SOP and EPH markers

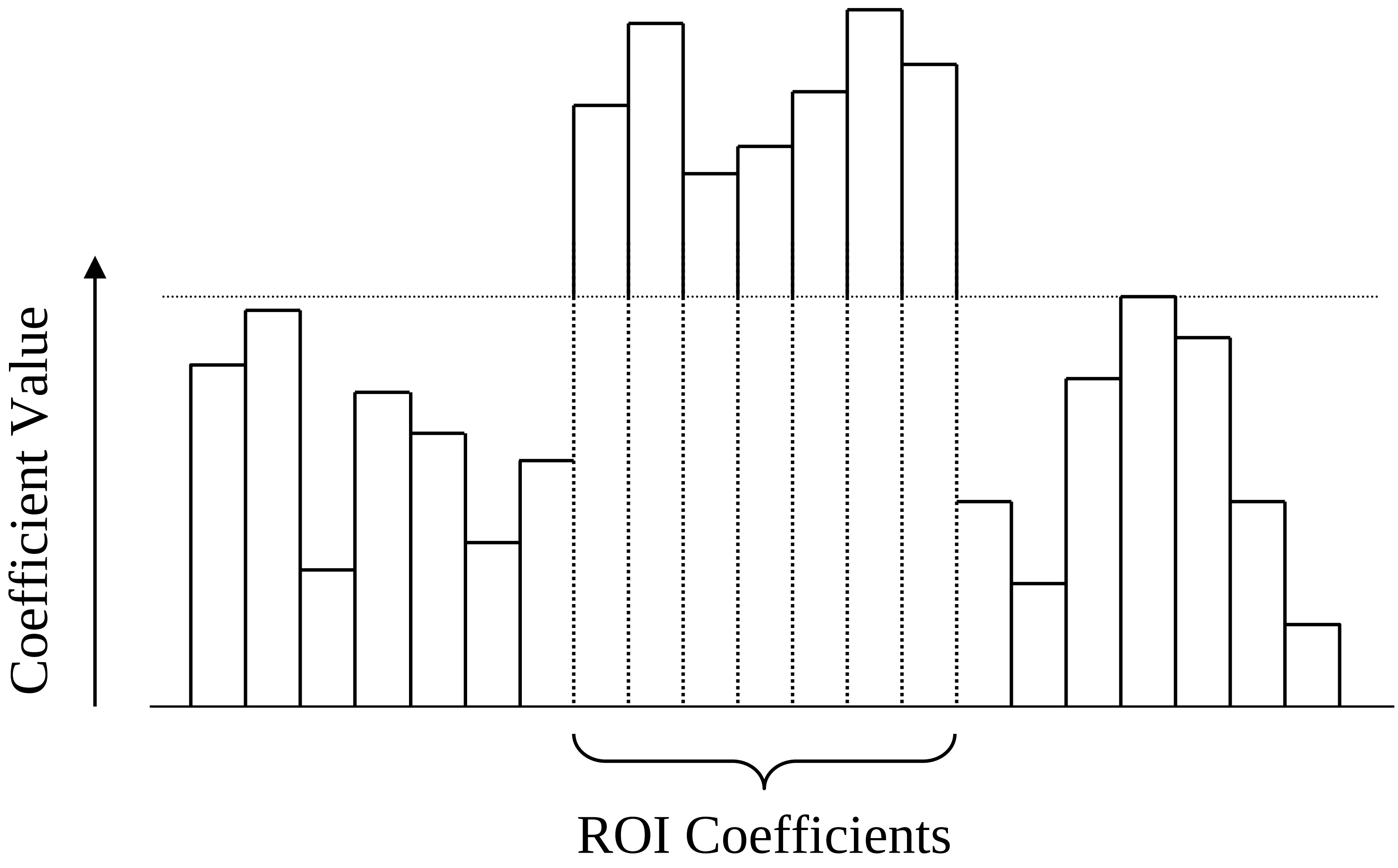
Region of Interest (ROI) Coding

- **Region-of-interest** (ROI) coding allows selected parts of an image to be coded with higher quality compared to the background (BG).
- The ROI encoding is done using an **ROI mask**. This binary mask is generated in the wavelet domain and describes which quantized wavelet coefficients must be encoded with higher quality. It depends on:
  - ROI region specification in the image domain
  - DWT filter

ROI Coding in Part I: Maxshift Method

- In the **Maxshift** method, the wavelet coefficients in the ROI region are scaled up by a fixed number of bits *s*, so that the smallest shifted nonzero ROI coefficient is larger than the largest BG coefficient. The parameter *s* is signaled in the bit stream. As a result, the decoder can discriminate between the ROI and BG coefficients by comparing each decoded value to a threshold. Pros and cons are:
  - It allows for multiple regions of arbitrary shape ROI without the inclusion of the shape information in the bit stream or need for ROI mask generation at the decoder.
  - The user can prioritize the coding of the ROI region over the BG but does not have control over the quality differential between ROI and BG.

Maxshift Method of ROI Coding



Other JPEG2000 Features

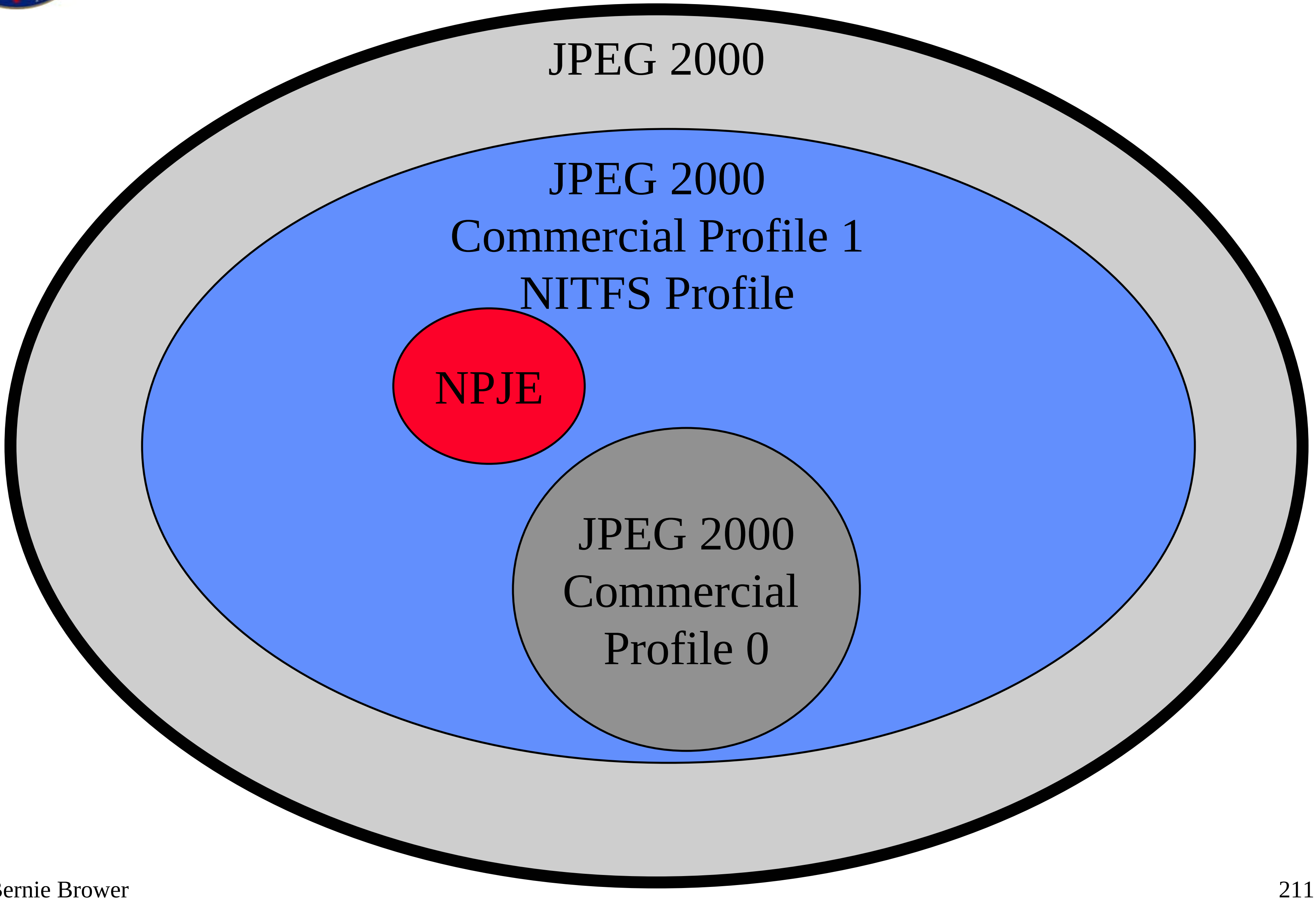
- Tiling of large images provides for independent processing of different image regions.
- The canvas coordinate system allows for efficient recompression of cropped images.
- Rich bit stream syntax provides means for transcoding of the data for streaming, resolution progression, quality progression, or any mixture thereof.
- Rich file format allows for various color spaces and metadata information.
- Compressed domain image manipulation (rotations of 90, 180, 270 degrees, horizontal and vertical flipping)

JPEG2000 Feature Summary

- Improved coding efficiency (up to 30% compared to DCT)
- Multi-resolution representation
- Quality scalability (SNR or visual)
- Target bit rate (constant bit rate applications)
- Lossless to lossy progression
- Improved error resilience
- Tiling
- Rich bit stream syntax (layering, packet partitions, canvas coordinate system, etc.)
- Rich file format

JPEG 2000 Profile

JPEG 2000 Standard and Profiles



Overview of the profile structure

- The Goal of the profile is to define the JPEG 2000 codestream restrictions (limits of the compressed data) that all NITFS 2.1 compliant systems will be required to support.
  - While keeping in line with the commercial profiles to save money with SCOTS.
- The profile will also promote wide interoperability for all NITFS systems
  - National/primary dissemination
  - Tactical/Secondary dissemination
- Make recommendations that try to achieve the greatest functionality for the NSGI architecture
- Give examples of the most common processes that would occur using the JPEG 2000 compressed data.
- Define the interaction between NITFS file format and the JPEG 2000 bitstream headers

Overview of the profile structure

- Introduction and Methodology: Background on JPEG 2000 and the NSGI architecture
  - Gives an understanding of why JPEG 2000 and what it will bring to the NSGI
- Profile: Define all the codestream limitations of JPEG 2000 Part 1.
  - This profile is the same as the ISO JPEG Profile 1 (right now)
  - We expect to add upon this when new hard requirements are defined
    - For example, we may add multiple component compression from Part 2
- Recommendations: Define the recommended values of parameters to promote the most functionality and interoperability within the NSGI
  - Supports the multiple resolutions, quality levels, and tile parsing.

Overview of the profile structure

- Population of NITFS Image Subheaders: How do the values in the NITFS Image Subheader interact with the image header in the JPEG 2000 codestream
- JPEG 2000 Processes Flow: Defines generic flows for multiple processes that would performed by every NITFS JPEG 2000 system
  - Encoding process flow
  - Decoding process flow
  - Parsing data flow
    - Tile parsing, quality layer parsing, resolution parsing
  - Enhancement processes (not defined as of yet)
  - Repackaging procedures
- Several Annexes
  - JPEG 2000 Commercial Profiles
  - JPEG/USIGS (NSGI) background and history

The NITFS 2.1 Supported Profile

- The NITFS 2.1 Profile will be compliant to the ISO JPEG 2000 Profile-1 compliant.
- The compliance class will be dependent on NITFS compliance class
  - We expect to support both JPEG 2000 Cclass 1 and 2
- The limitation of the JPEG 2000 Profile 1
  - Rsiz = 2 (Marker that states that this is profile 1)
  - The image size is limited to less than 2^{31}
  - Tiles are limited to no greater than 1024-by-1024 and must be square (XTsiz = Ytsiz, Xtsiz/min(XRsiz, YRsiz) $\leq$ 1024) or one tile for the entire image
    - If one tile is used, the LL subband should be included that is no bigger than 128 on a side.

The NITFS 2.1 Supported Profile

- The limitation of the JPEG 2000 Profile 1
  - The image and tile origins are required to be less than 2^{31}
  - There is a limit to 37 region of interests (ROI) for each image
  - Code-block sizes are limited to 2^6 (64 maximum to a side)
- Compliance classes are not specified in the profile and will be part of the compliance testing of the JITC
  - The commercial compliance classes are found in JPEG 2000 Part-4
  - The general concept is to guarantee decoding depending on the image size, memory and computational cost to the decoder

	Cclass 0	Cclass 1	Cclass 2	Cclass 3
Image Size limits	128x128	2,048x2,048	16,384x16,384	2^{31}x2^{31}
Number of bands	1	4	256	16,384
Bit Depth	8	12	24	32
NITFS Clevel	None	Clevel 3	Clevel 5	Clevel 6, 7, 9

NSGI Recommendations

Tile and Wavelet Recommendations

- Recommend images tiled at 1024 x 1024 pixels
  - Allow fast access to spatial chips
  - At least five levels of wavelet decomposition (R0 through R5)
    - Enables resolution scalability
    - R5 is 32 x 32 in size
    - R6+ generation is an issue
      - Do more wavelet levels initially
      - Mosaic R5s and wavelet transform
- Recommend one tile part per tile
  - Chipping is more important "tile progression"
- Tile Length Markers (TLM markers) are recommended to speed access to tiles
  - Appears in codestream main header
  - Can be used to derive pointers to start of each tile
- Packet Length Tile markers (PLT markers) are recommended to facilitate parsing of packets
  - Appears in each tile header
  - Facilitates parsing of packets (units of entropy coded wavelet coefficients)

Proposed Layers and Applications

- Layers enable *quality* scalability
- If *numerically* lossless compression is needed
  - Use the 5-3R integer reversible wavelet transform
  - 5-3R can also be used for lossy compression
- For lossy only compression
  - Use the 9-7I irreversible floating point wavelet transform
  - Better lossy performance than the 5-3R

Layer	Bits Per Pixel (bpp)	Application(s)
Layer 19 (5-3R filter)	Lossless	Radiometric
Layer 19 (9-7I filter)	Visual lossless	MC&G
Layer 18	3.5 bpp	MC&G
Layer 17	2.3 bpp	MC&G
Layer 16	2.0 bpp	MC&G
Layer 15	1.7 bpp	MC&G Visual exploitation
Layer 14	1.5 bpp	Visual exploitation
Layer 13	1.3 bpp	Visual exploitation
Layer 12	1.2 bpp	Visual exploitation
Layer 11	1.1 bpp	Visual exploitation
Layer 10	1.0 bpp	Visual exploitation and Tactical users
Layer 9	0.9 bpp	Tactical users
Layer 8	0.8 bpp	Tactical users
Layer 7	0.7 bpp	Tactical users
Layer 6	0.6 bpp	Tactical users
Layer 5	0.5 bpp	Tactical users
Layer 4	0.25 bpp	BW constrained users
Layer 3	0.125 bpp	BW constrained users
Layer 2	0.0625 bpp	BW constrained users
Layer 1	0.03125 bpp	BW constrained users

Progression Recommendation

- Recommend "LRCP"
  - Assume want best R0 quality as a function of rate
  - Other progressions will occur in commercial imagery!
    - Libraries must be able to read such files
    - Libraries should be able to change "transcode" from one progression to another
- JPEG 2000 allows for the progression order to change within a file (POC marker)
  - Useful for interactive streaming sessions
    - Server streams data to user
    - User may tell server to concentrate on an area or resolution
  - If you don't want full resolution or have different resolutions within a mosaic

Other Recommendations

- Code blocks
  - 64 x 64 in size recommended
    - Increase arithmetic coder efficiency
    - In commercial imagery may see 32 x 32 and possibly 1024 x 4
- Precincts (tiling within wavelet subbands)
  - Not recommending their use since we use tiles
    - Standard allows you to use both in one codestream
  - Will definitely see this in commercial images
- Reference Grid
  - Image offsets (XOsiz, YOsiz) set to (0, 0)
  - Tile offsets (XTOsiz, YTOsiz) set to (0, 0)
  - 1024 x 1024 tile size will allow us to chip and maintain these values
  - Imagery with other tile sizes or tile/image offsets may require manipulation of these values when chipped

TRE Motivation

- Place information about JPEG 2000 codestream in NITF image subheader
  - Search on this information
  - Required capabilities to decode
  - Significance of layers in codestream
- Retain information about original image in parsed or chipped images
- This TRE will accommodate NITF JPEG 2000 codestreams

Tag Details

Field	Name/description	Size bytes and format	Req. or Con.	Value Range
CETAG	Unique Extension Type Identifier Unique TRE identifier.	6, BCS-A	R	J2KLRA
CEL	Length of User-Defined Data Length in bytes of data contained in subsequent TRE fields. (TRE length is 11 plus the value given in the CEL field)	5, BCS-N	R	Variable Calculated for each specific TRE.
ORIG	Original compressed data Indicates if the image is in the same original JPEG 2000 compression or it has been parsed to a new JPEG 2000 compression. The conditional fields (NLEVELS_I, NLAYERS_I, NBANDS_I) are only present if ORIG = 1.	1, BCS-N	R	0 - Original NPJE 1 – Parsed NPJE 2 – Original EPJE* 3 – Parsed EPJE* 8 – Original other 9 – Parsed other

Tag Details

Original compressed image information (the first JPEG 2000 Compression)				
Number of Wavelet levels in original image Indicates the number of wavelet decompositions levels performed in the original image.	2, BCS-N	R	00 - 32	
Number of bands in original image Indicates the number of bands in original image.	5, BCS-N	R	00000 - 16384	
Number of Layers in original image Indicates the number of layers in original image.	3, BCS-N	R	000 - 999	

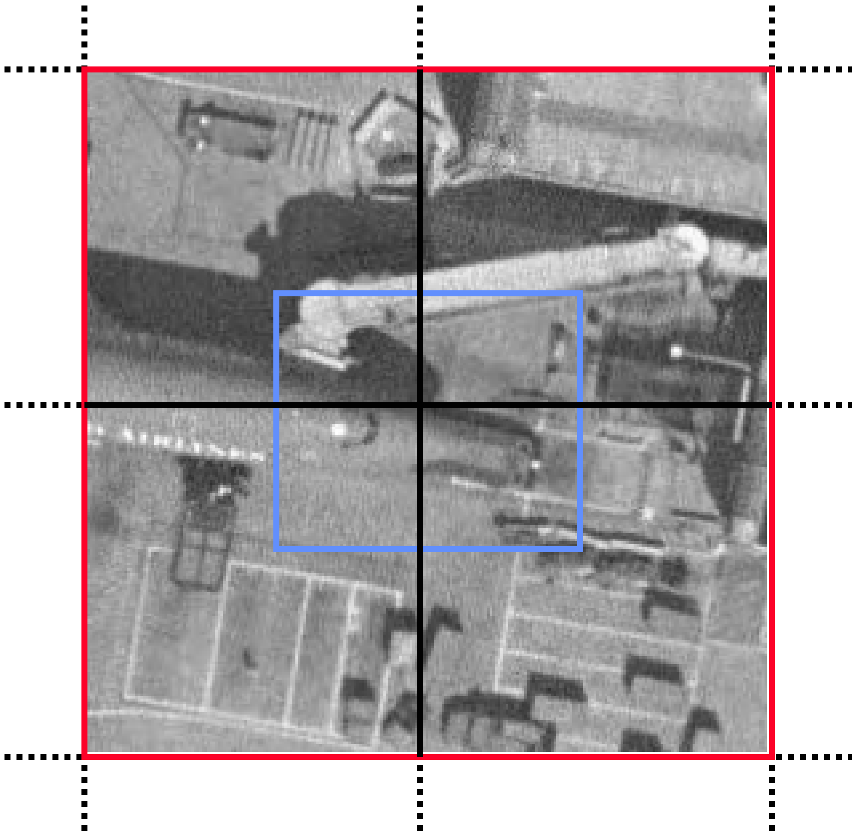
Tag Details

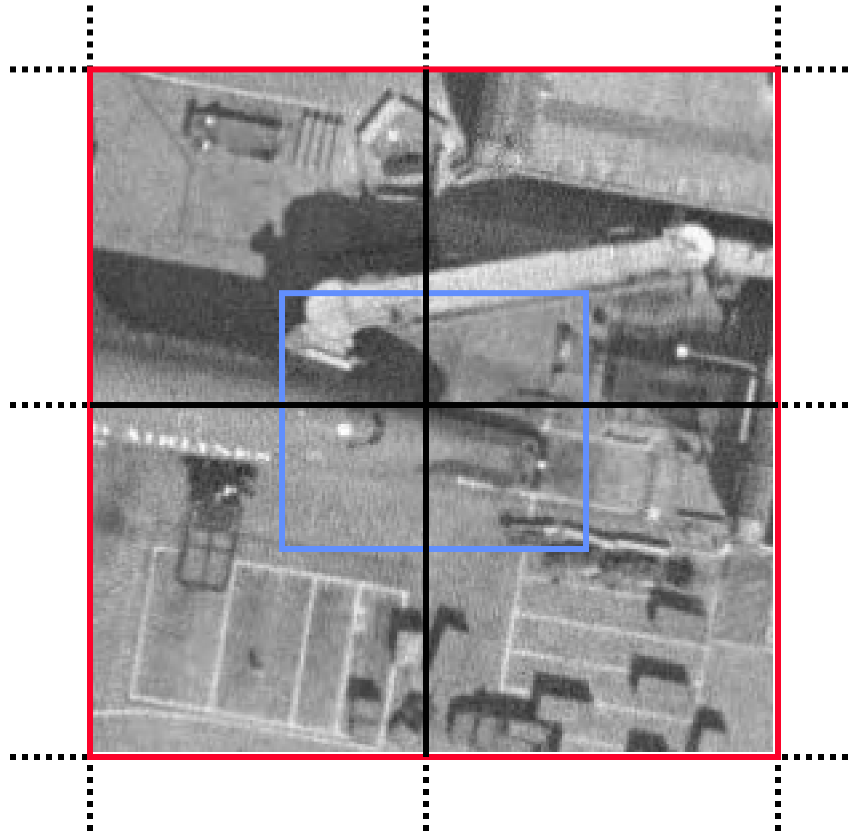
Layer information (This is the start of a repeating section for n = 0 to NLAYERS_O – 1)				
$LAYER\_ID_n$	Layer ID Number Indicates the number of layer being described. Layers are numbered from 0 to NLAYERS_O –1. 0 is the layer with the lowest bitrate.	3, BCS-N	R	000 - 999
$BITRATE_n$	Bitrate Indicates the accumulated bitrate target associated with this and associated lower layers. This is defined in bits per pixel per band. It may happen that the bitrate was not achieved due to data characteristics. Note for JPEG 2000 numerically lossless quality, the bitrate for the final layer is an expected value based on past performance. If there is not a target bit rate, report the achieved bit rate.	9, BCS-A	R	Value 00.000000 – 37.000000

Tag Details

(This is the end of a repeating section.)				
Conditional fields if the data has been parsed; when ORIG = 1, 3, and 9				
NLEVELS_I	Number of Wavelet levels in this image Indicates the number of wavelet decompositions levels included in this image as defined in COD.	2, BCS-N	C	00 – 32
NBANDS_I	Number of bands in this image Indicates the number of bands in this image as defined in SIZ.	5, BCS-N	C	00000 - 16384
NLAYERS_I	Number of Layers in this image Indicates the number of layers in this image as defined in COD.	3, BCS-N	C	000 - 999

RO Display





Current RRDS, 1.3 bpp
Display Requires
4 1024x1024 Tiles at 1.3 bpp

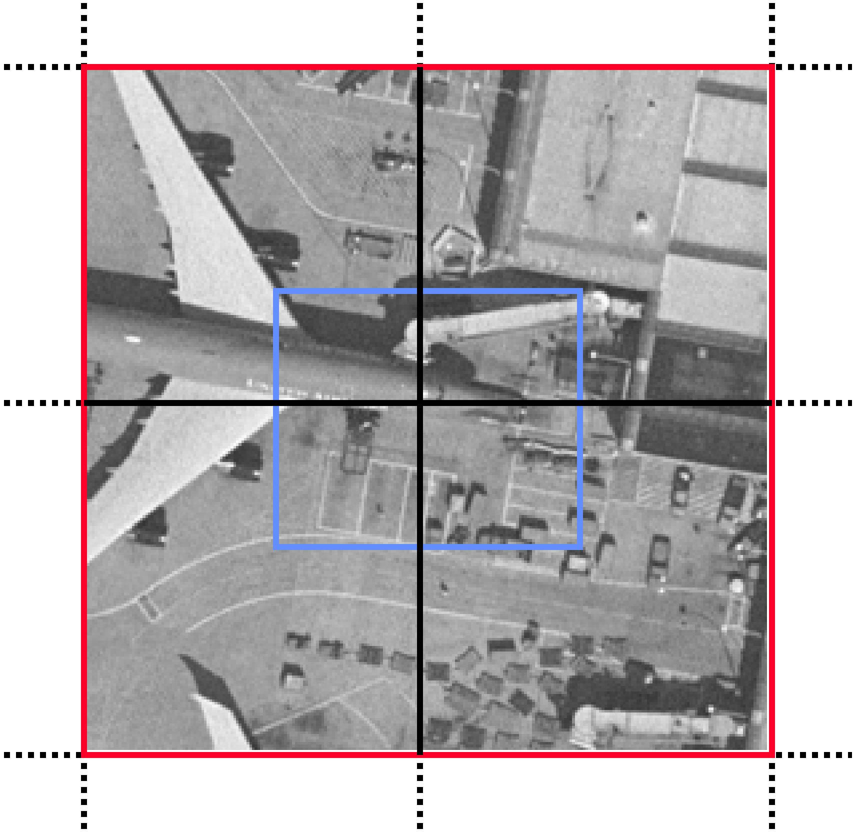
Image Tiles Data

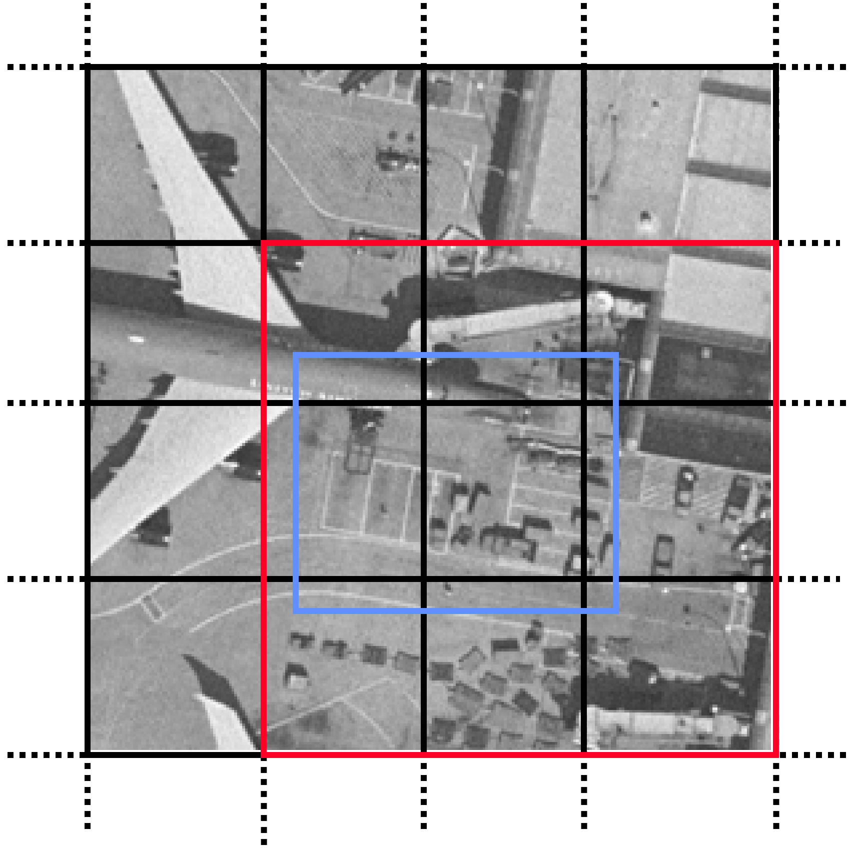
Tiles Sent to display

Display Area (1024-by-1024)

JPEG 2000
Display Requires
4 Tiles at 1.0 bpp

R1 Display



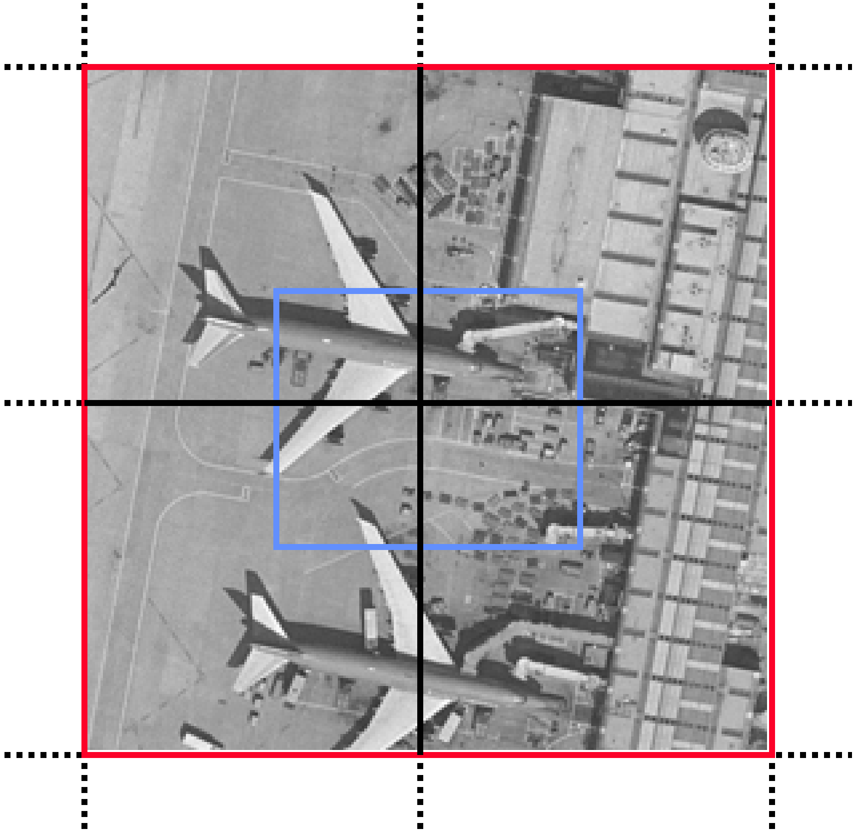


Current RRDS, 1.3 bpp
Display Requires
4 1024x1024 Tiles at 1.3 bpp

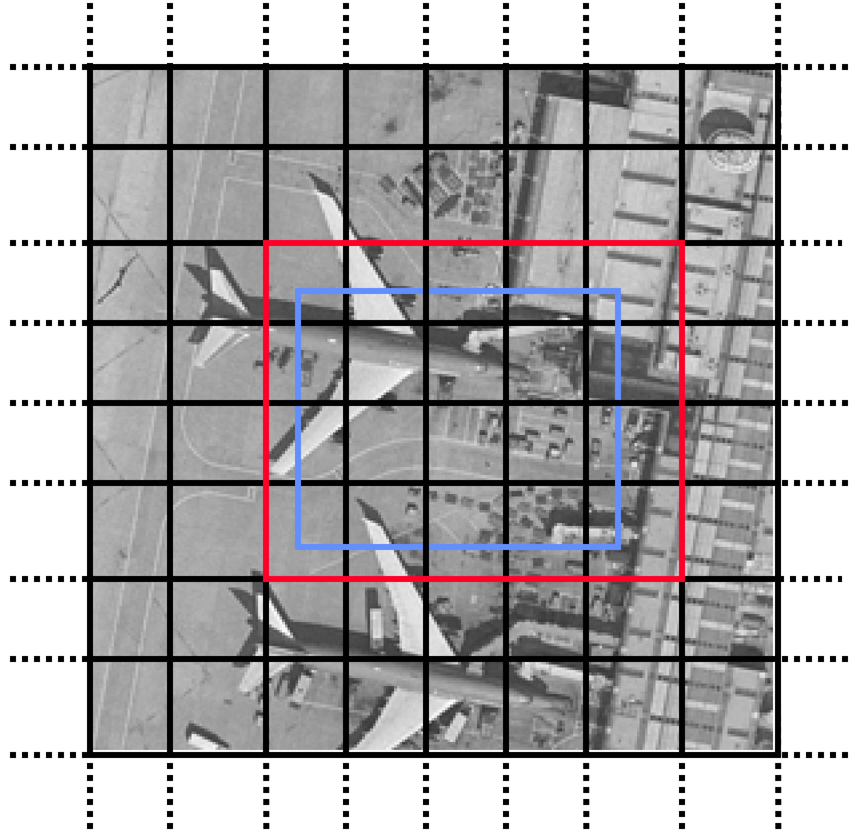
JPEG 2000
Display Requires
9 512x512 Tiles at 1.9 bpp

- Image Tiles Data (black)
- Tiles Sent to display (red)
- Display Area (1024-by-1024) (blue)

R2 Display



Current RRDS, 1.3 bpp
Display Requires
4 1024x1024 Tiles at 1.3 bpp



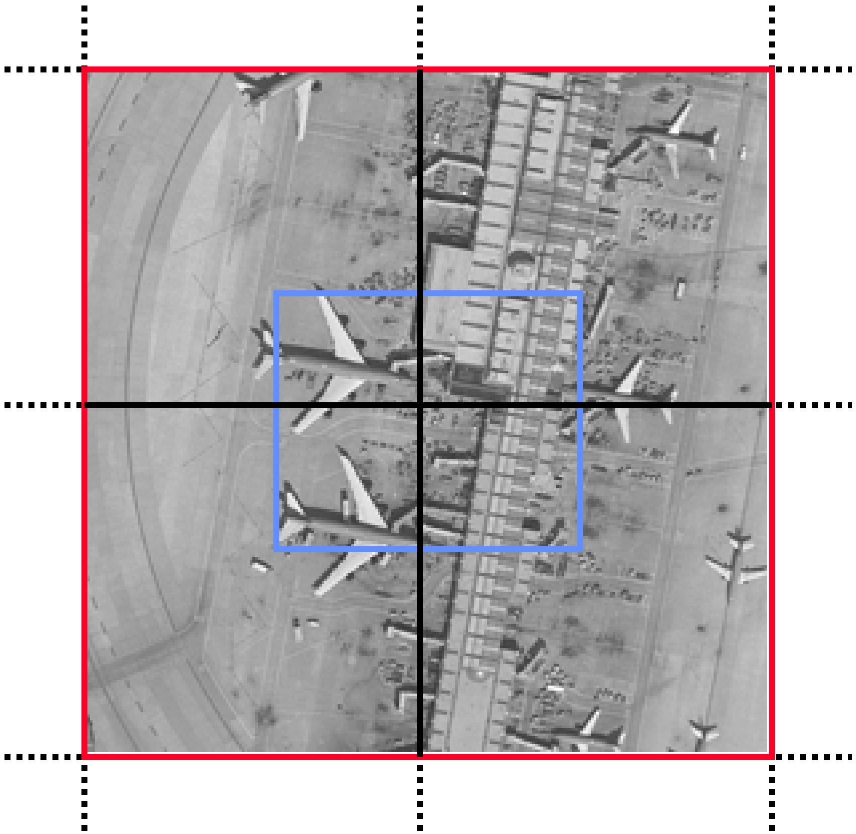
JPEG 2000
Display Requires
25 256x256 Tiles at 2.7 bpp

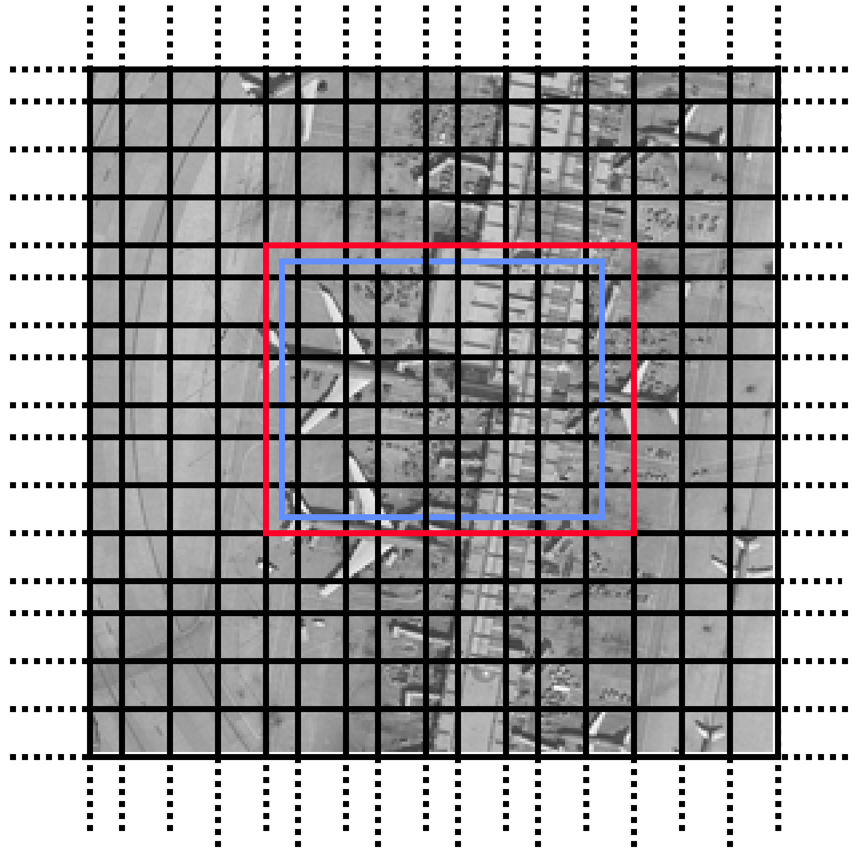
Image Tiles Data

Tiles Sent to display

Display Area (1024-by-1024)

R3 Display





Current RRDS, 1.3 bpp
Display Requires
4 1024x1024 Tiles at 1.3 bpp

Image Tiles Data
Tiles Sent to display
Display Area (1024-by-1024)

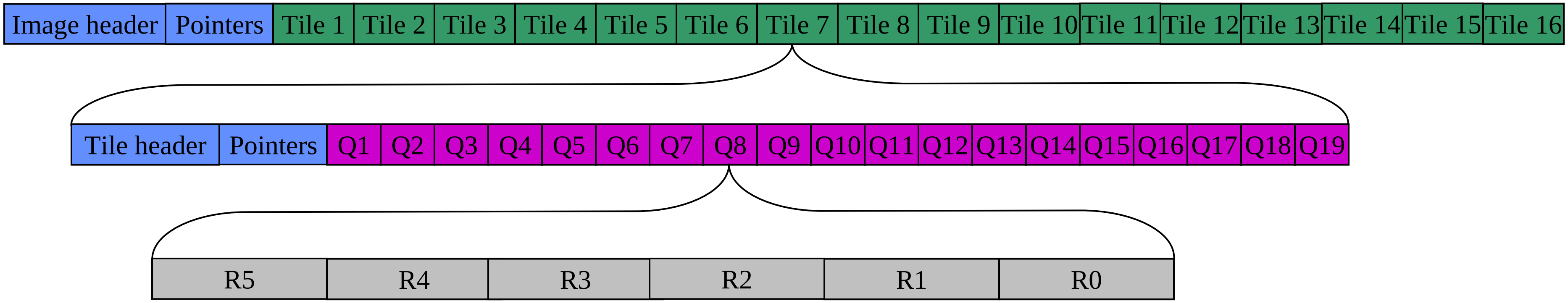
JPEG 2000
Display Requires
81 128x128 Tiles at 3.3 bpp

What is EPJE?

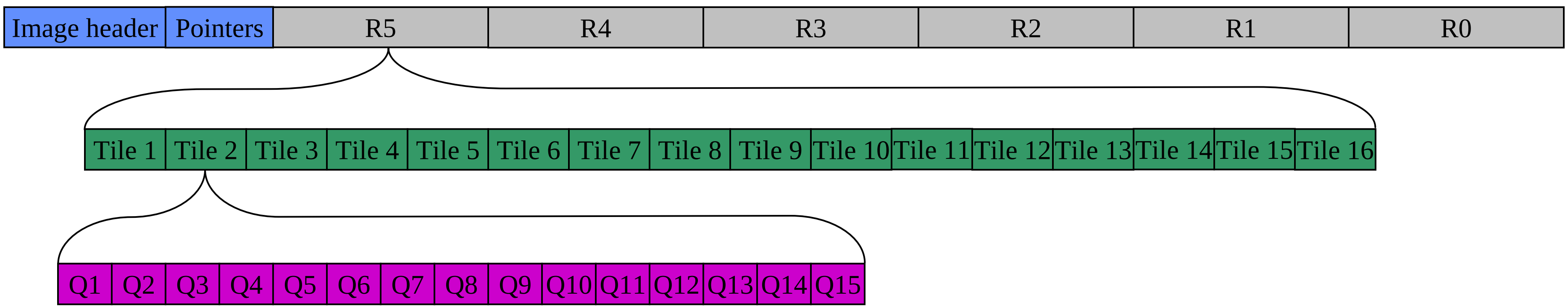
- Exploitation Preferred JPEG 2000 Encoding
- Improved access time for lower resolutions in large images
  - Common improvement is for images larger than 20,000 by 20,000
- Data can be rearranged from NPJE encoded data
  - Identical decoded data
- There is a difference is in progression order
  - NPJE is LRCP progression order
  - EPJE is in RLCP progression order
- There is a difference in tile parts
  - NPJE is one tile part
  - EPJE has multiple tile part

NPJE versus EPJE

NPJE



EPJE



Compression Overview

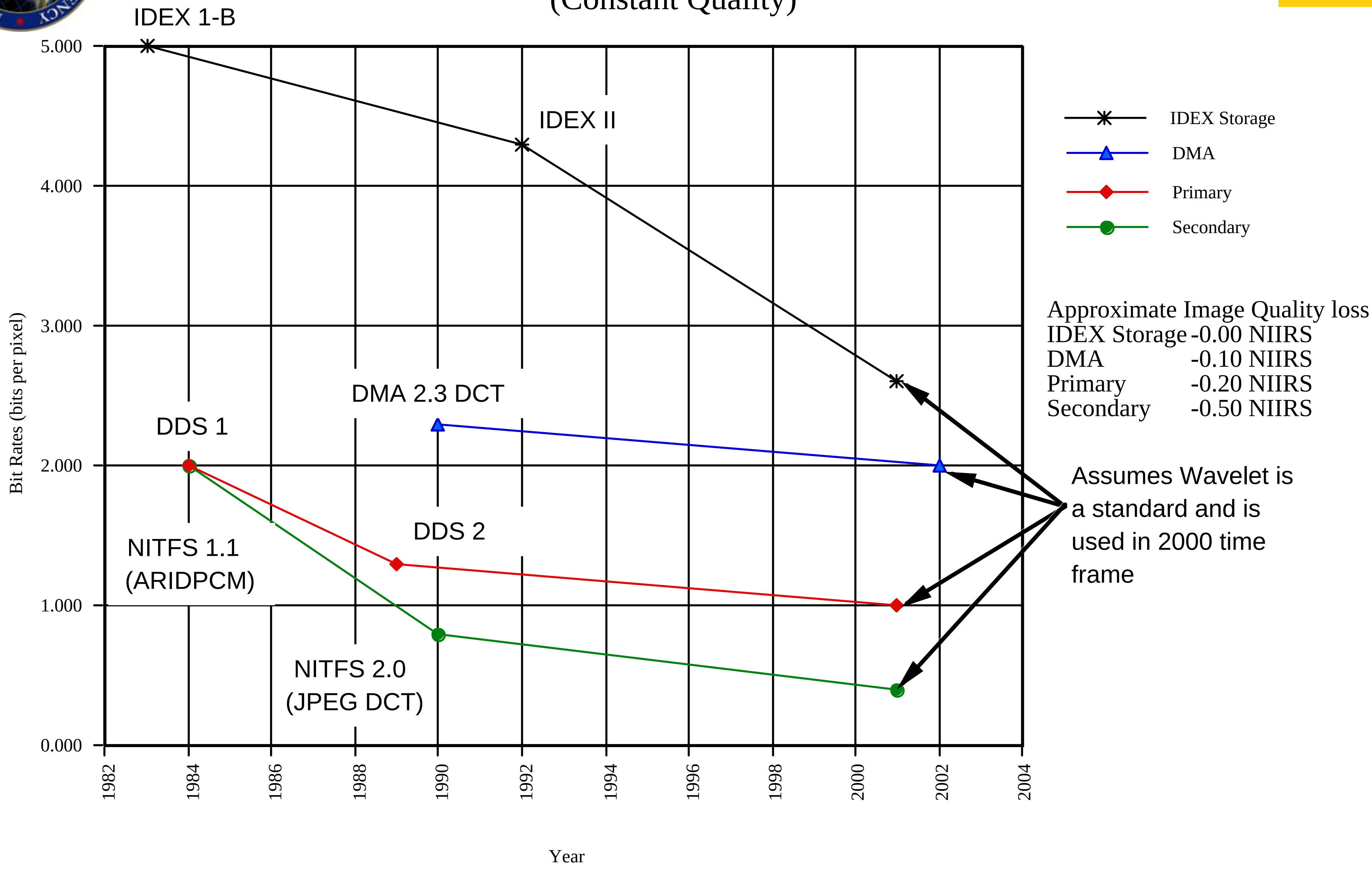
Compression Algorithm	Transform Technique	Quantization	Encoding	Comments
4.3 DPCM	Linear prediction from neighboring pixels	Table look-up	Variable length Huffman encoding	• Low complexity • High quality • Low compression ratio
NITFS JPEG DCT	8-by-8 block Discrete Cosine Transform (DCT)	Human Visual System (HVS) Response quantization in DCT space	Variable length Huffman encoding	• Can be rate controlled • 8-by-8 transform used for speed
1.3 / 2.3 DCT	32-by-32 block DCT	HVS quantization	Variable length Huffman encoding	• Rate controlled to either 1.3 bpp or 2.3 bpp
Vector Quantization	No transform performed	Vector code book matching	Code book numbers	• Low channel error susceptibility
NIMA Method 4	Down-sampled followed by JPEG DCT	JPEG DCT	JPEG DCT	• Achieves very low bit rate at reasonable quality
JPEG 2000	Wavelet-based sub-band transform	Scalar Quantization with Dead-Zone	Bit-Plane Arithmetic encoder	• Highest quality of any of the algorithms • Most functional

Compression Overview

Algorithm	Advantage	Disadvantage
4.3 DPCM	• Low complexity (low power/size/weight) • Visually lossless quality • Low memory requirements • Government standard • Rate controlled	• Low compression ratios compared to frequency-based transform techniques.
1.3 DCT	• High quality • Military standard • Rate controlled	• High complexity (32-by-32 transform, rate-control) • Blocking artifacts
2.3 DCT	• Near lossless quality • Government standard • Rate controlled	• High complexity (32-by-32 transform, rate-control) • High bit rate (2.3 bpp)
NITFS JPEG DCT	• International/commercial standard • Low cost implementation (COTS) • Low complexity (8-by-8 transform)	• Blocking artifacts • Lower quality than 1.3 DCT and wavelets
NITFS VQ	• Low complexity for decompression • Low susceptibility to channel error • High quality on DMA maps • Military standard	• High complexity for compression • Relatively poor quality on images
NIMA Method 4	• Interoperable with NITFS JPEG • High quality at low bit rates • Military standard	• Does not perform well at higher bit rates • Not flexible
Wavelets	• Better quality to compression ratio than any other compression algorithm • Significantly more functionality • Commercial Standard	• Large memory requirements • Computational Complexity • Significant start-up cost

Compression Rate Improvement Over Time

(Constant Quality)



Related Issues (File Formats)

- TFRD
  - Used for primary dissemination
  - End users exploit imagery to develop image products
    - Requirements included SDE information
  - Does not include graphics or overlays
- NITFS
  - Used for secondary dissemination of exploited data
  - End users only use imagery and do not exploit imagery
  - Require graphics and overlays
- The two are trying to extend their file format and compression to incorporate the others needs.
  - NIMA now has control over all of the systems (primary/secondary)

Acronym List

- Image Processing
  - DRA = Dynamic Range Adjustment
  - TTC = Tone Transfer Curve
  - RRDS = Reduced Resolution Data Sets
  - MSE = Mean Square Error
  - NIIRS = National Image Interpretability Rating Scale
  - RMSE = Root MSE
  - Color space transformations
    - Red Green Blue (RGB) Hue Intensity Transform (YCrCb, YIQ, YUV)
  - HVS = Human Visual System
- General
  - COTS = Commercial off the self
  - GSD = Ground Sampled Distance
  - DPI = Dots Per Inch
  - IA = Image Analyst

Acronym List

- Compression
  - DPCM = Differential Pulse Code Modulation
  - DCT = Discrete Cosine Transform
  - JPEG = Joint Photographic Experts Group
  - MPEG = Motion Picture Experts Group
  - VQ = Vector Quantization
- Systems/ users/ formats
  - USIGS = United States Image and Geospatial System
  - DDS = Defense Dissemination System
  - NITFS = National Imagery Transmission Format Standards
  - NIMA = Nation Imagery and Mapping Agency
  - TFRD = Tape Format Requirements Document
  - RE/RL = Receive Entity
  - DE = Distribution Entity
  - DoD = Department of Defense