NIMA
Systems Engineering
Services

Kodak

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

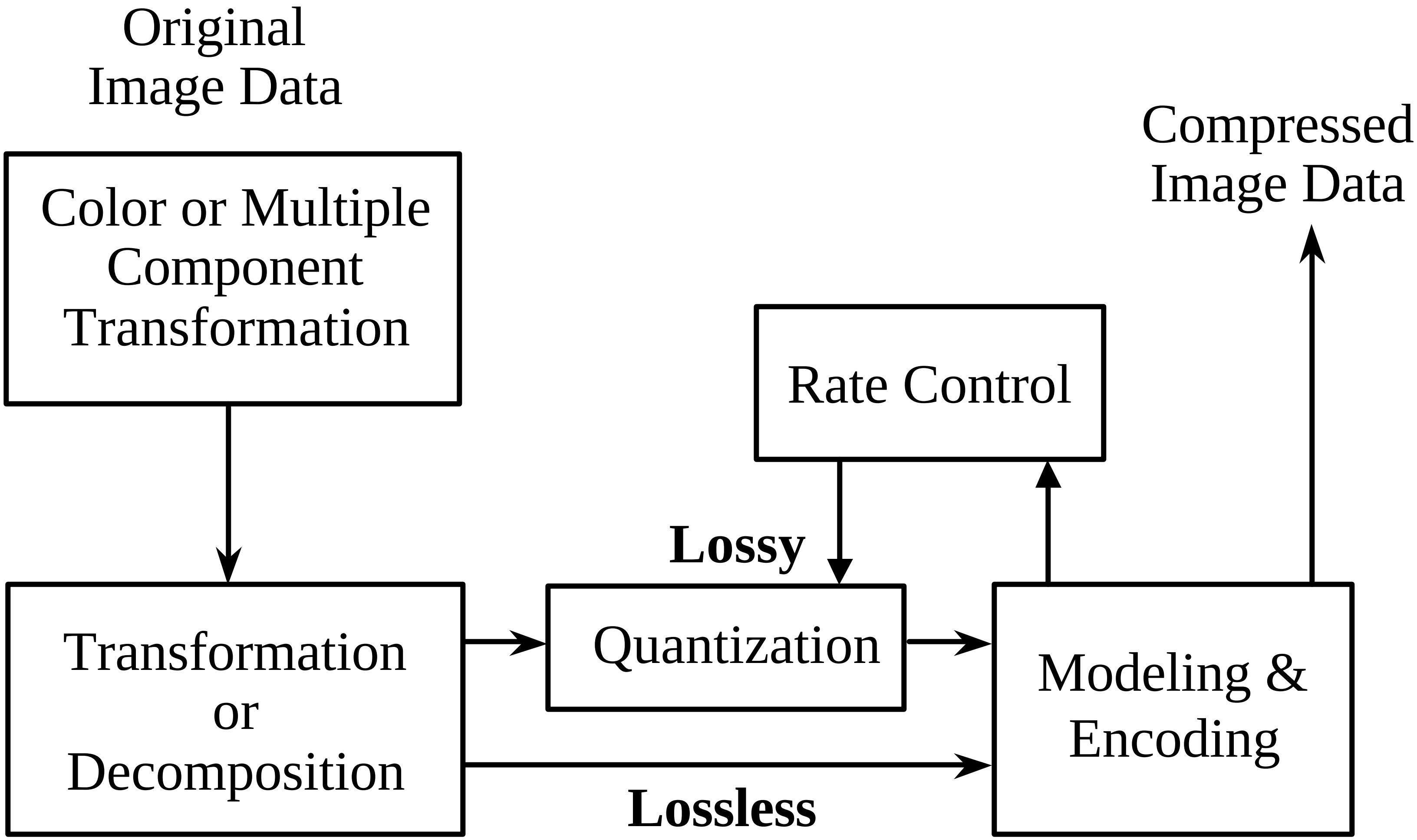
Lossless (Reversible) Compression

- 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

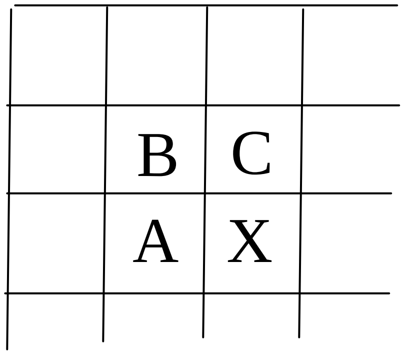
Decomposition or Transformation

- 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

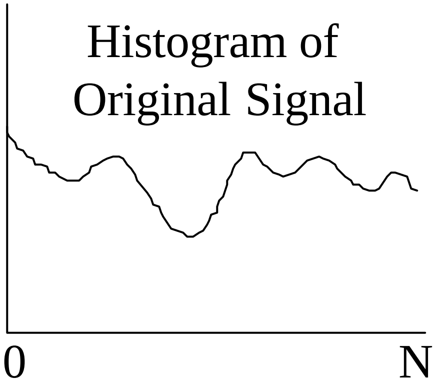
- Lossless JPEG and 4.3 DPCM are based on differential pulse code modulation (DPCM).
  - In DPCM, a combination of previously encoded pixels (A, B, C) is used as a prediction (χ) for the current pixel (X).
  - The difference between the actual value and the prediction (χ – X) is encoded using Huffman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

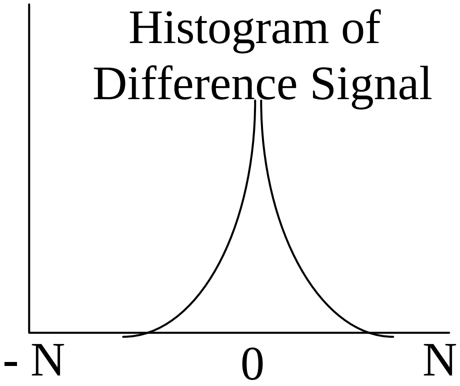


χ = A

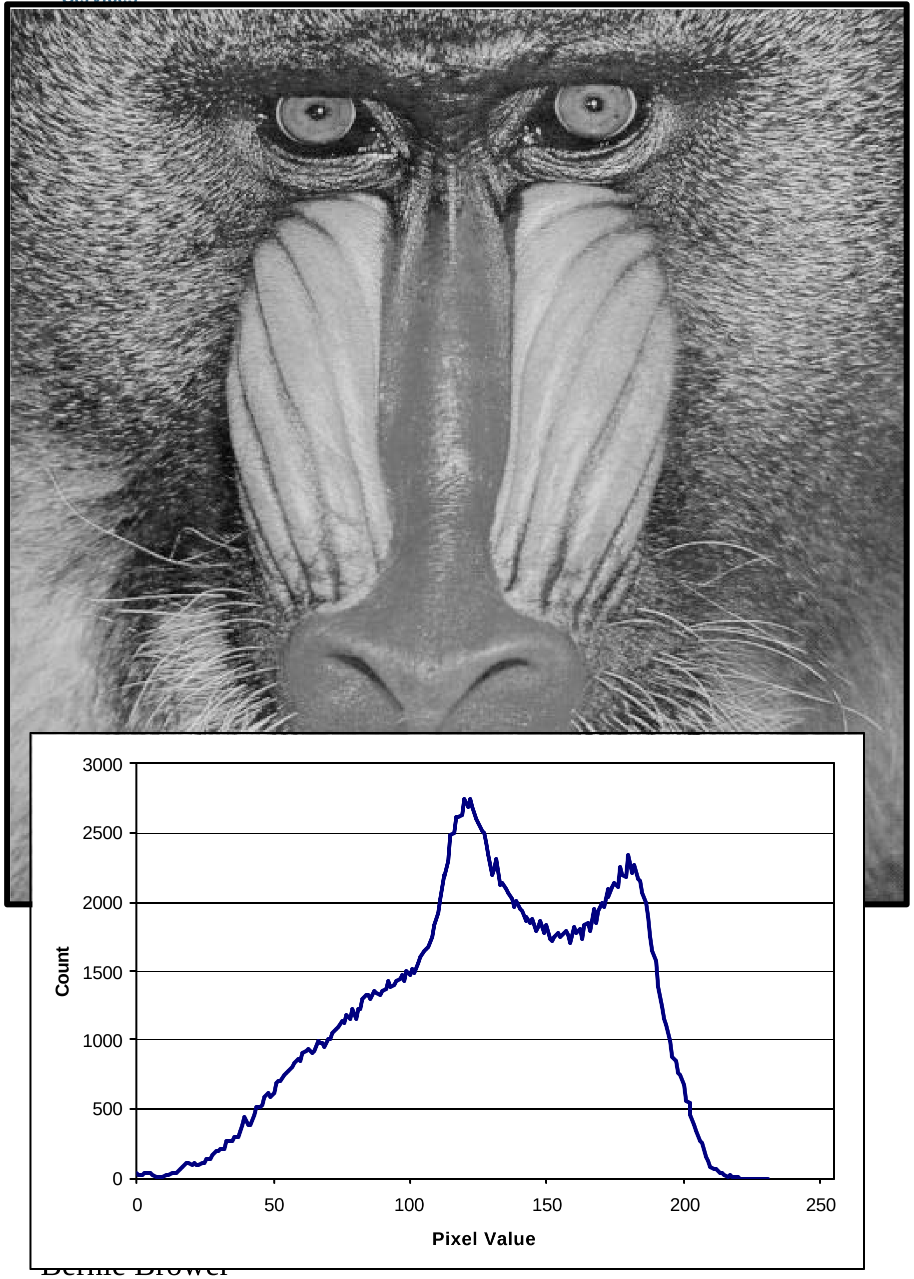
χ = (A +C)/2

χ = (A + C - B)

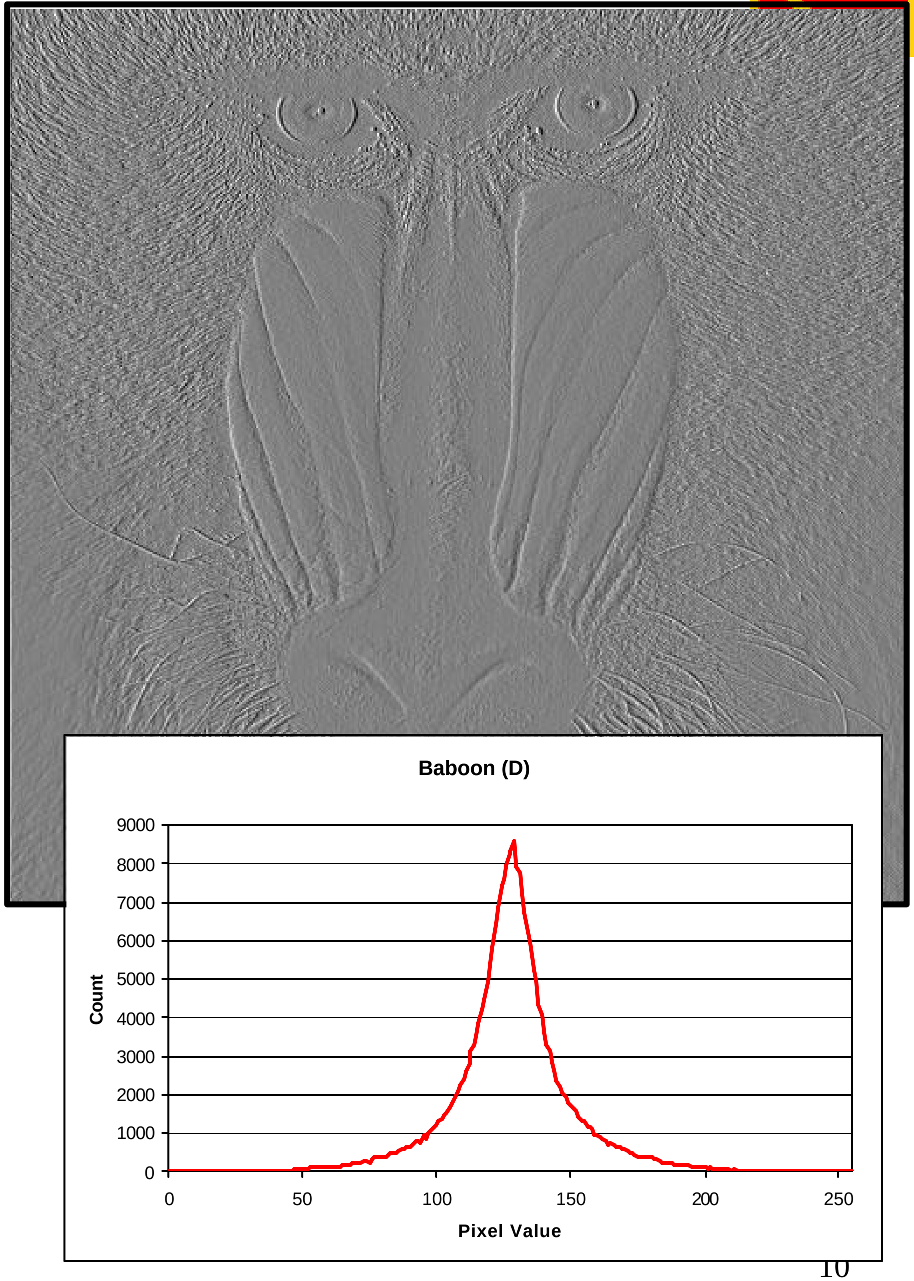




Original



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

Systems Engineering Services

2-Dimensional 8 x 8 DCT Basis Functions

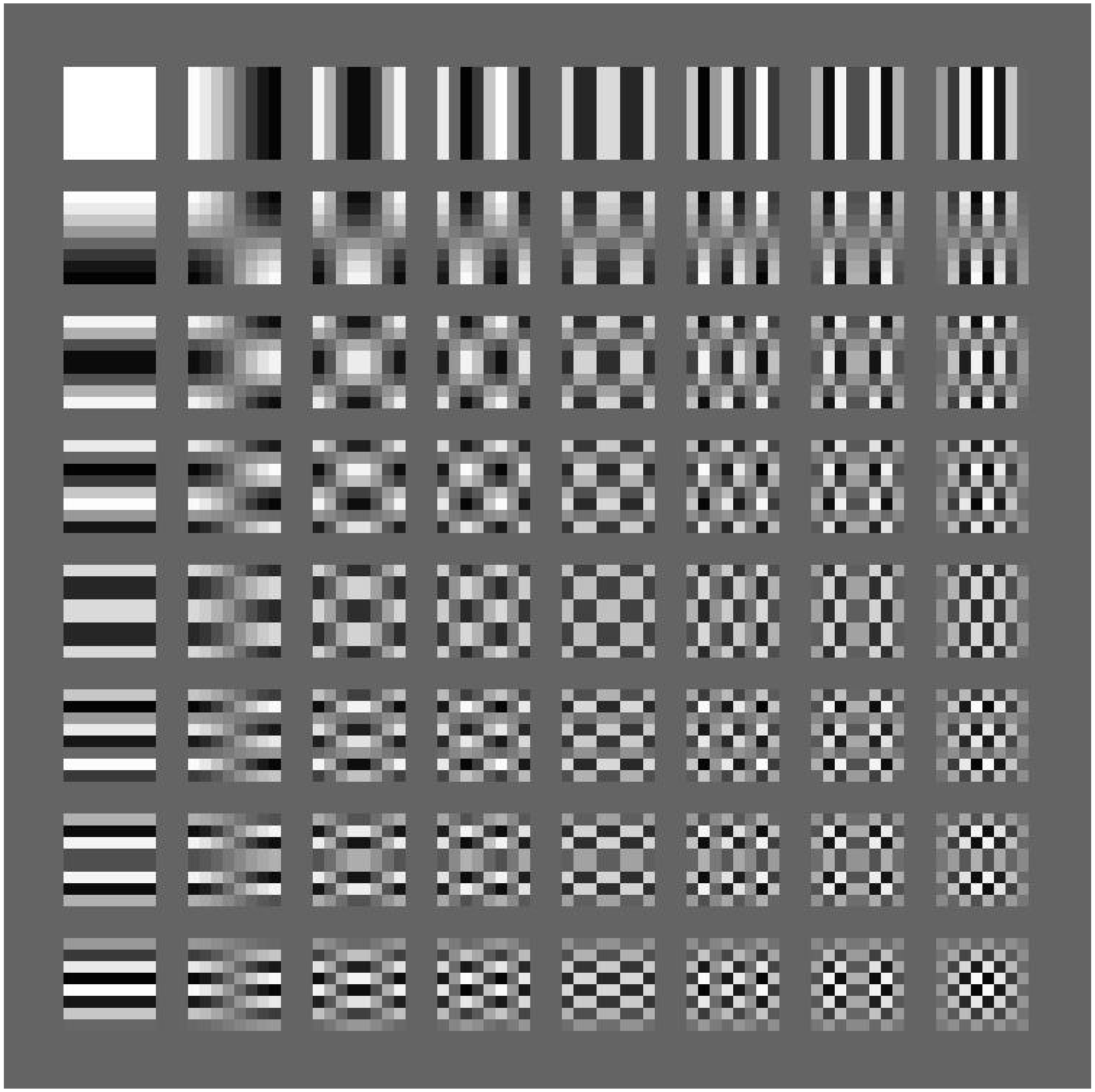
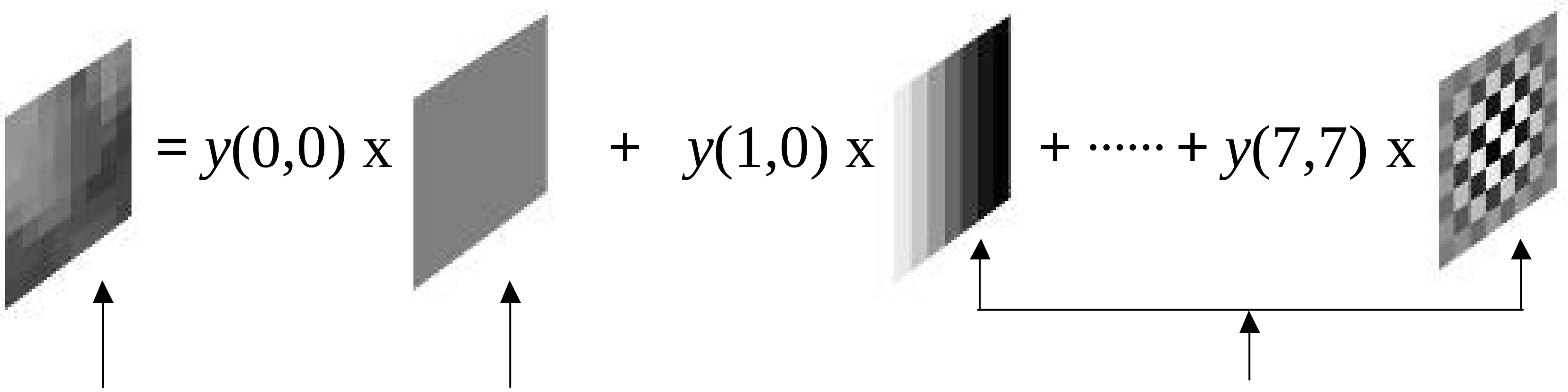


Image Representation with DCT

- 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.



$= y(0,0) \times + y(1,0) \times + \cdots\cdots + y(7,7) \times$

Original
image block

DC (flat) basis
function

AC basis
functions

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

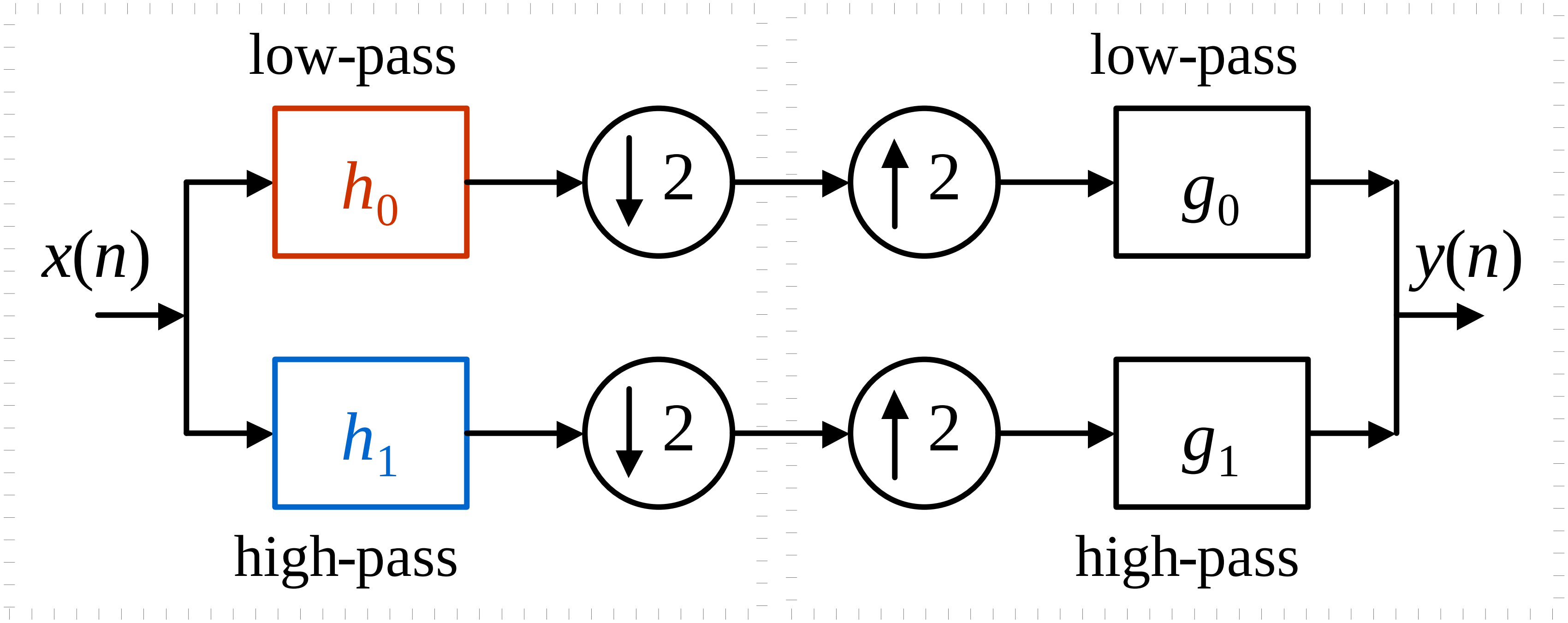
$y(u,v) =$

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



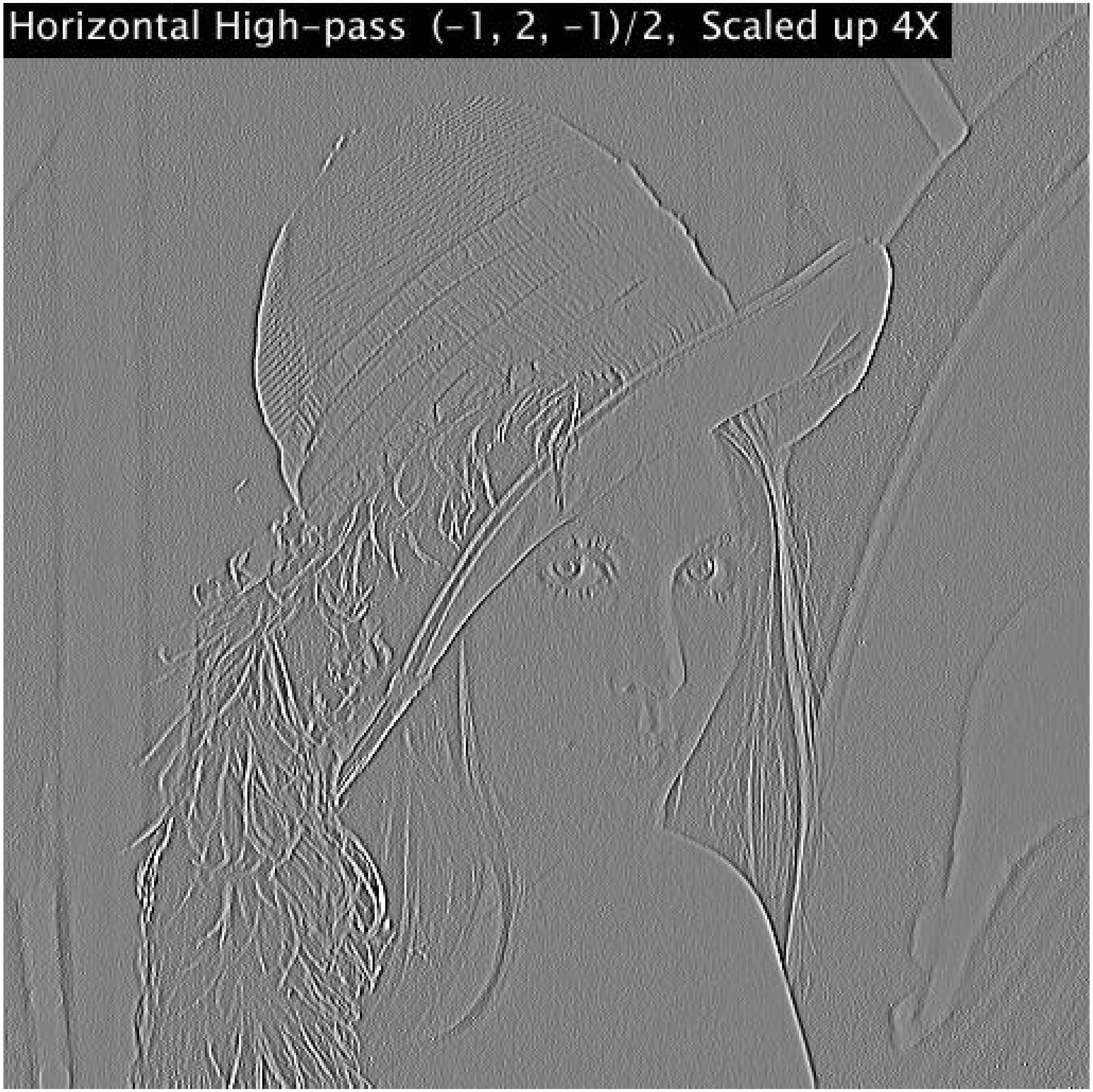
Analysis filter bank

Synthesis filter bank

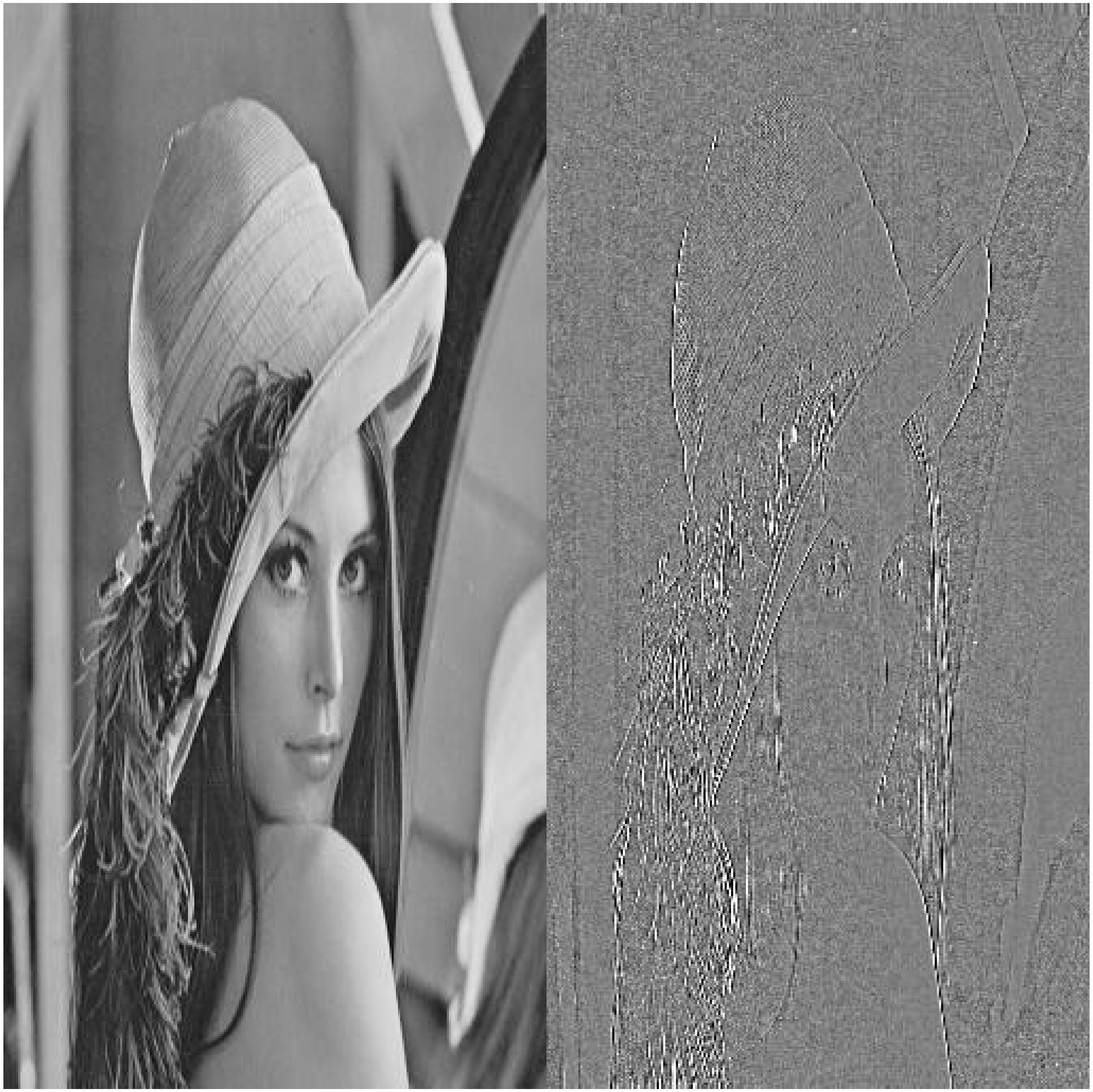
Ideally, it is desired to choose the analysis filter banks (h_0 and h_1), and the synthesis filter banks (g_0 and g_1), in such a way so as to make the overall distortion zero, i.e., $x(n) = y(n)$. This is called the **perfect reconstruction** property.

Systems Engineering Services

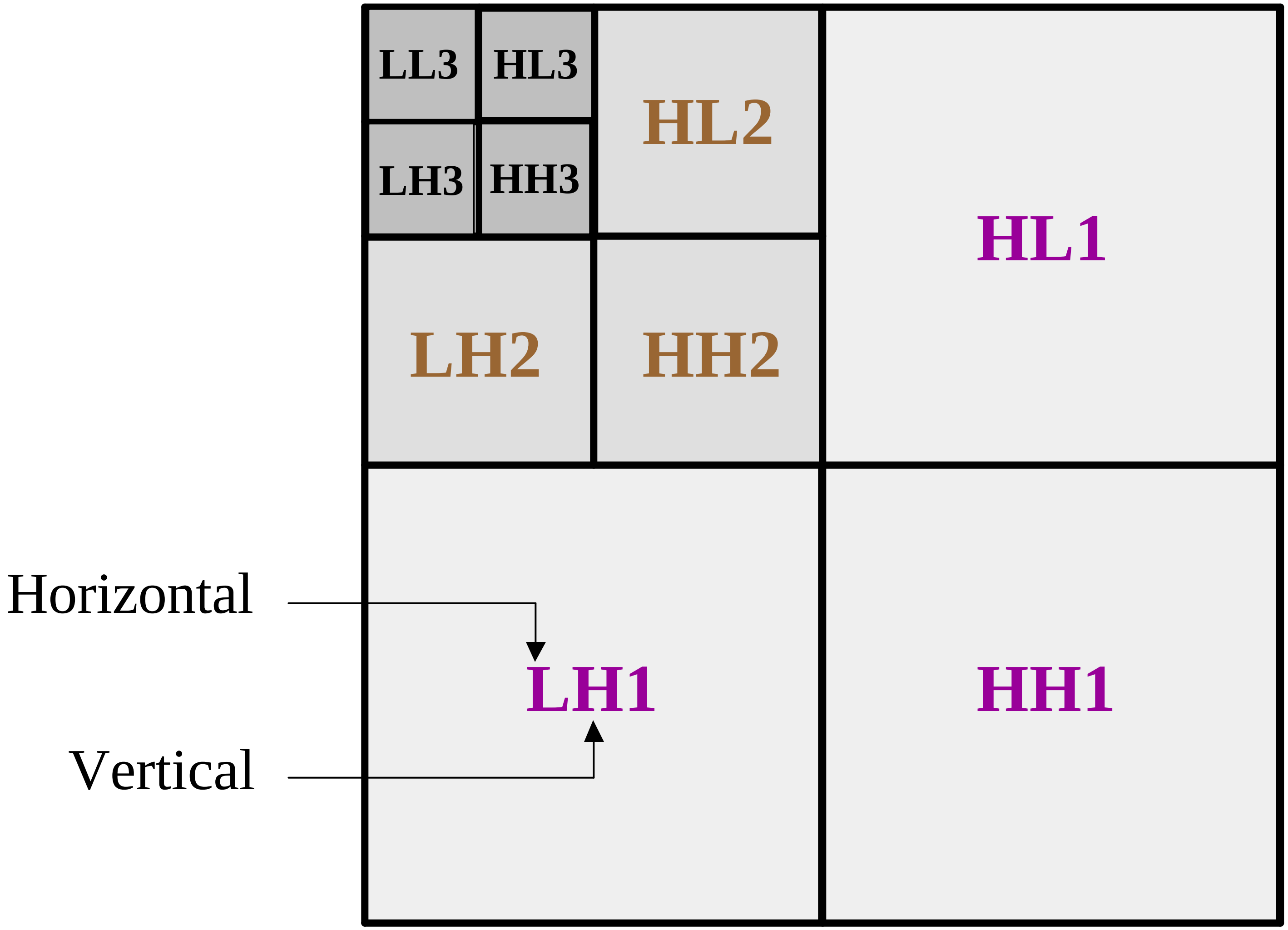
Horizontal Low-pass (-1, 2, 6, 2, -1)/8



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



2-D Wavelet Decomposition



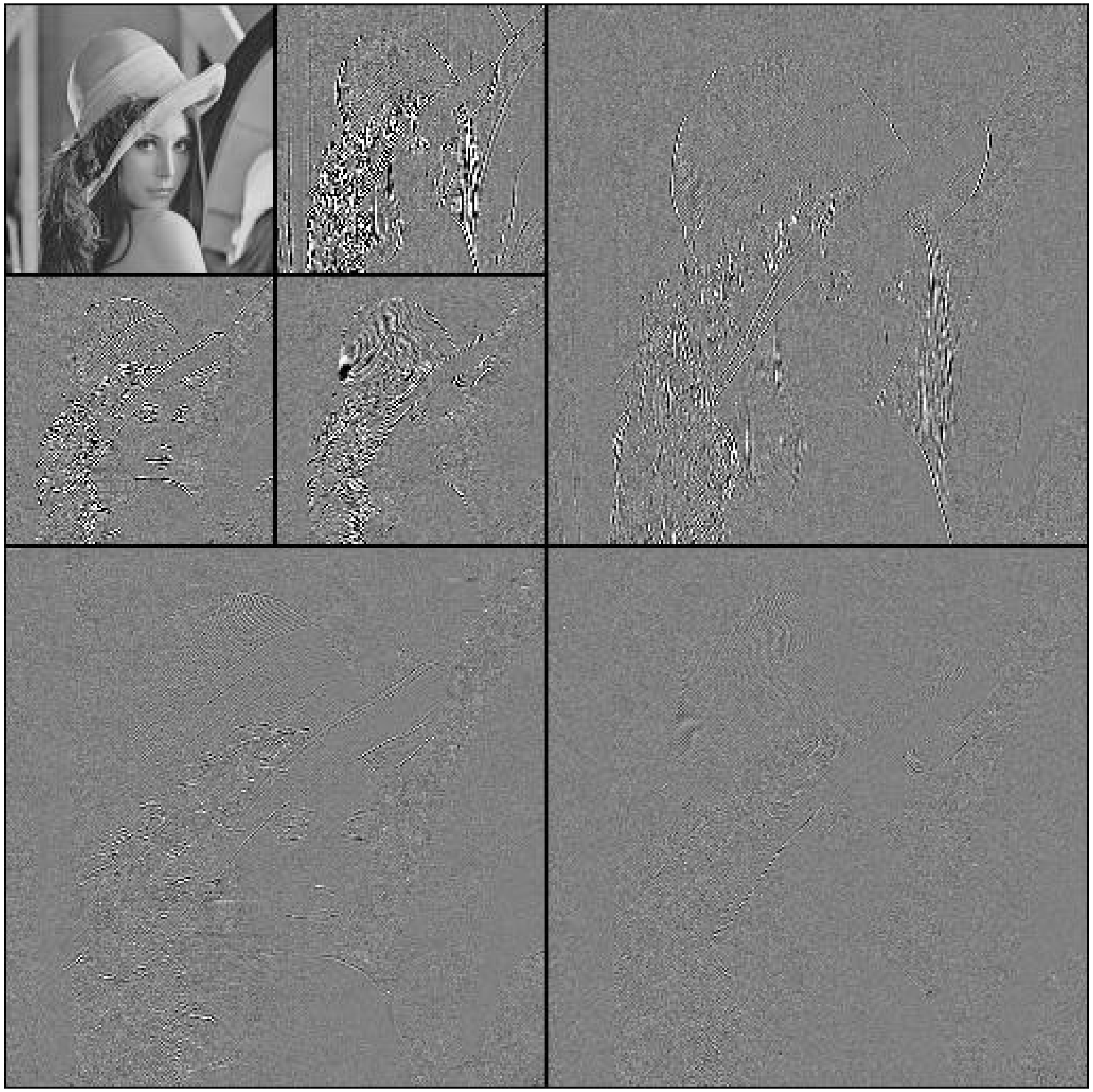
Original Lena Image

Systems Engineering
Services

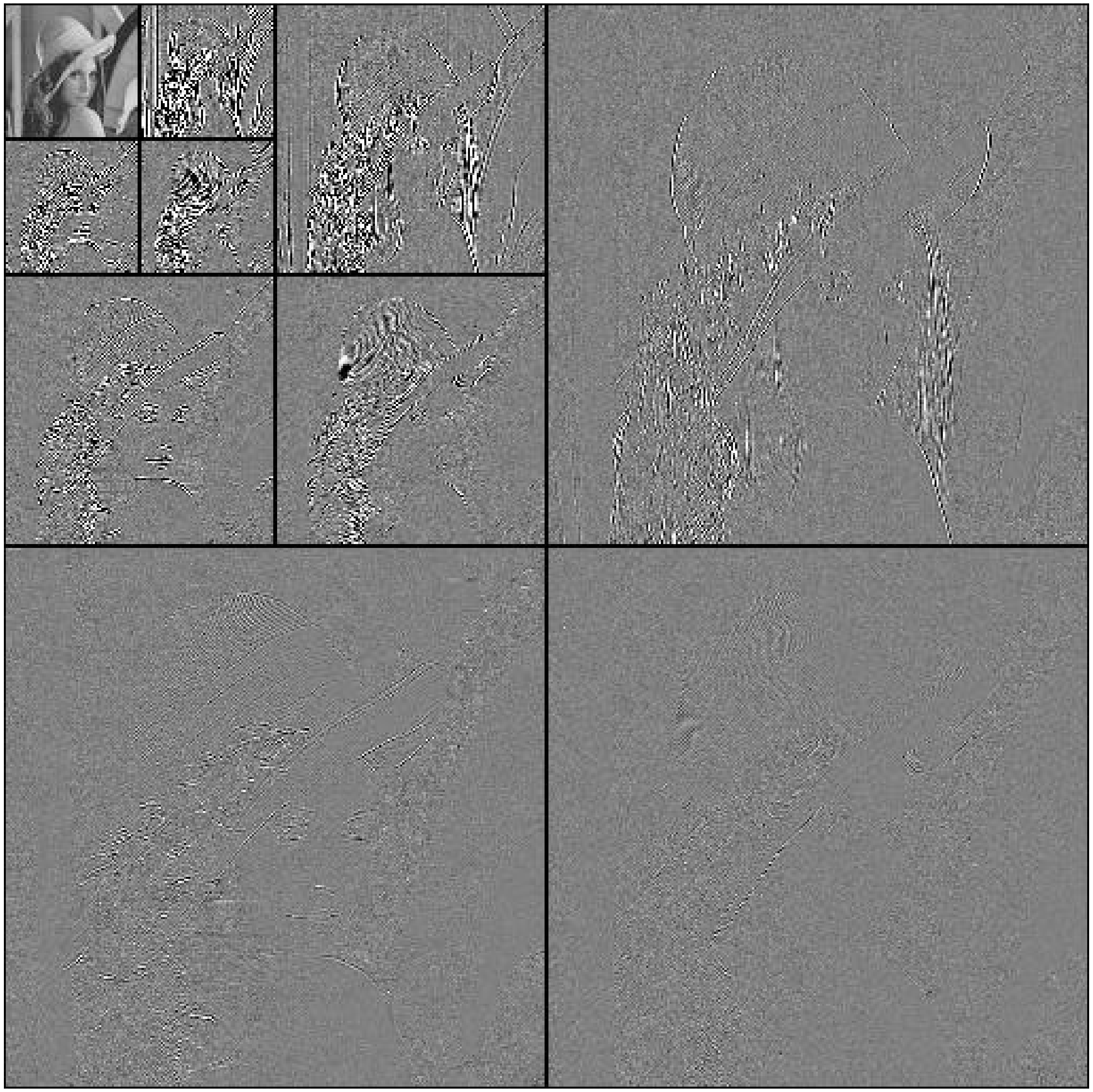
1-Level, 2-D Wavelet Decomposition of Lena

Systems Engineering Services

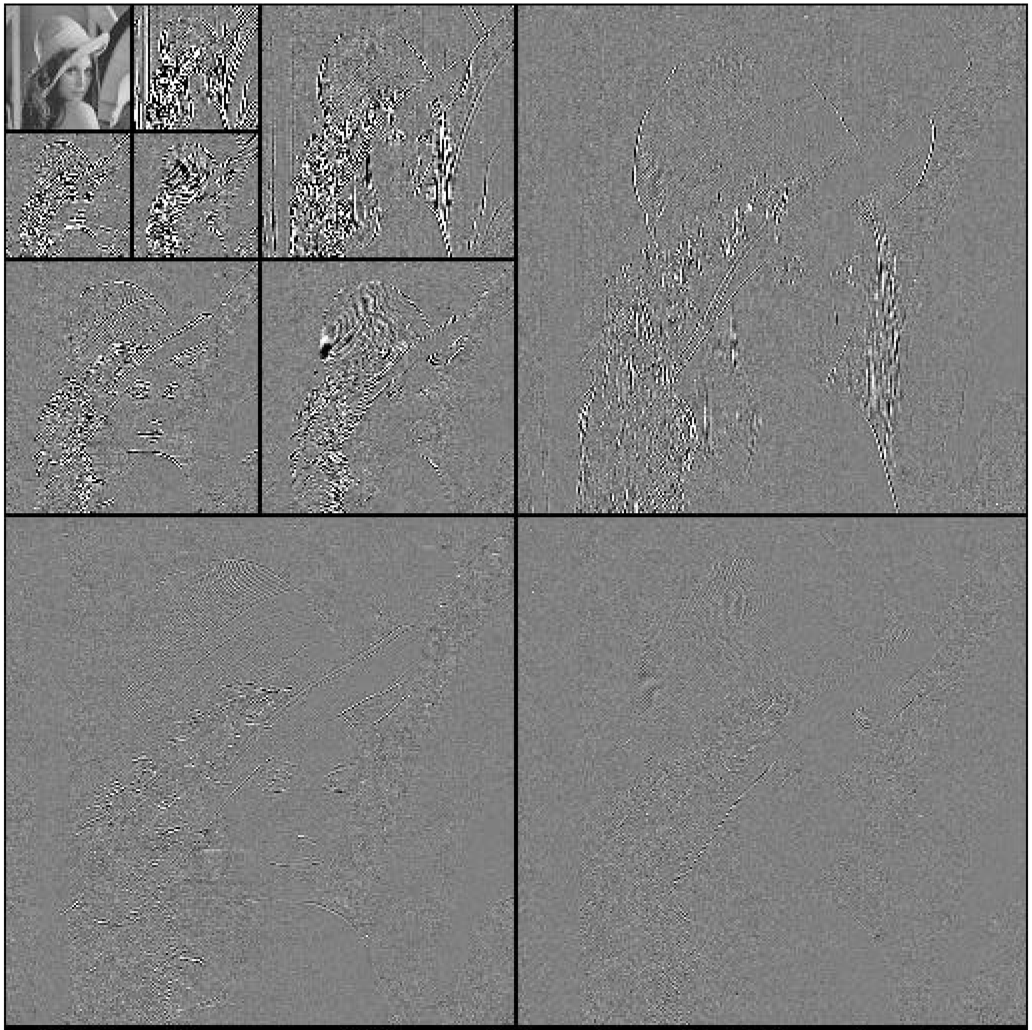
2-Level, 2-D Wavelet Decomposition of Lena



3-Level, 2-D Wavelet Decomposition of Lena



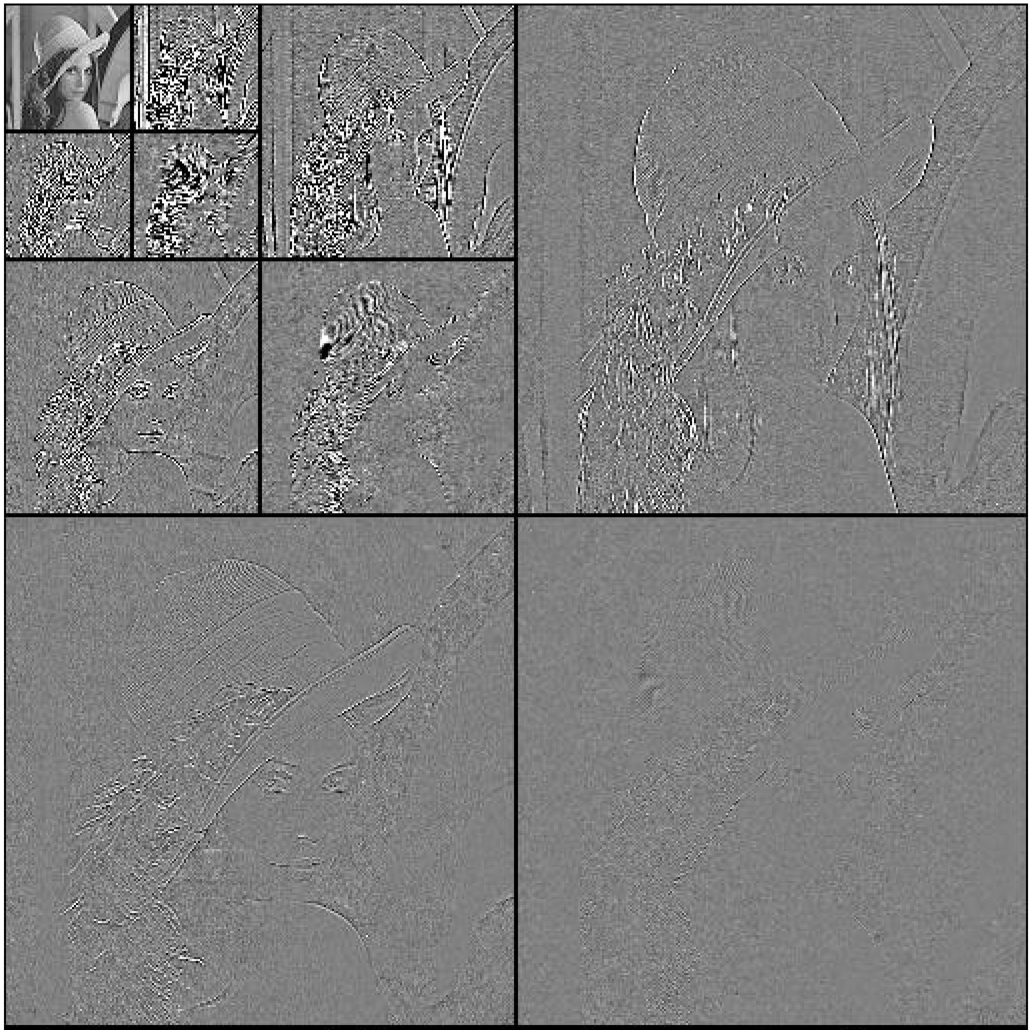
3-Level, 2-D DWT with (9,7) Filter



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

Systems Engineering
Services

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



(5,3) orthonormal filter, AC scale = 4, DC scale =1/8

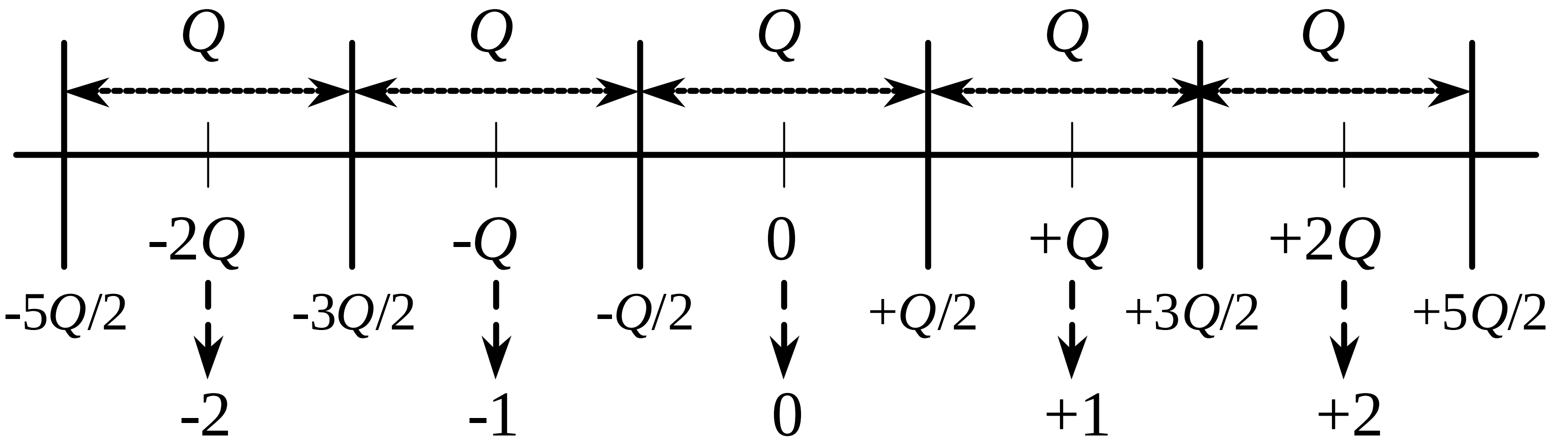
Quantization

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).

Systems Engineering Services

Uniform Threshold Quantizer (UTQ)

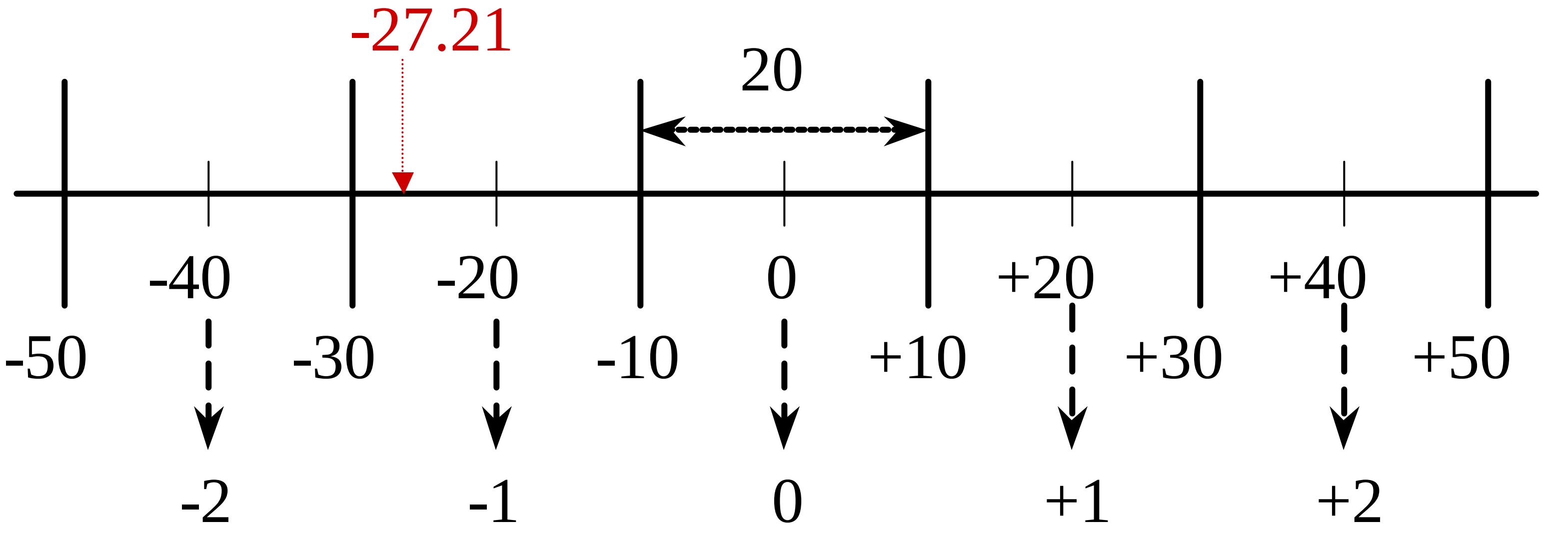


- In a **UTQ** quantizer, all bins have the same size. The bin size *Q* is called the quantizer **step size**. The quantization dequantization rule for a midpoint reconstruction is given by:
  - Quantization rule: $q = \mathrm{NINT}[y/Q]$
  - Dequantization rule: $z = q * Q$
  - Where y is the input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

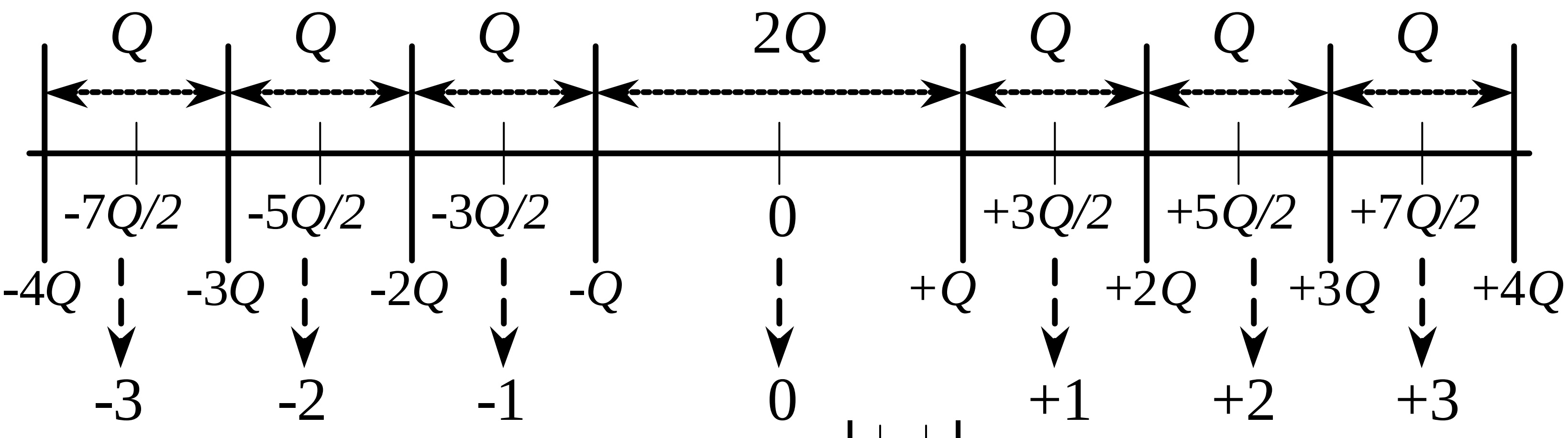
Systems Engineering Services

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

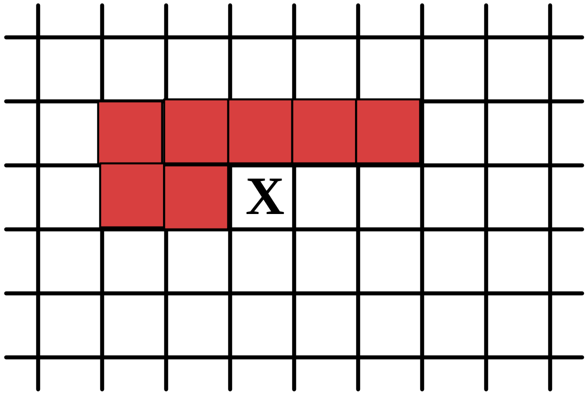
Line 1	0	0	4	0	0	0	1	0	1	1
Code 1	00	00	11	00	00	00	01	00	01	01
Code 2	0	0	111	0	0	0	10	0	10	10
Line 2	0	0	3	0	0	0	1	0	1	1
Code 1	00	00	10	00	00	00	01	00	01	01
Code 2	0	0	110	0	0	0	10	0	10	10

- 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.

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



Systems Engineering
Services

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

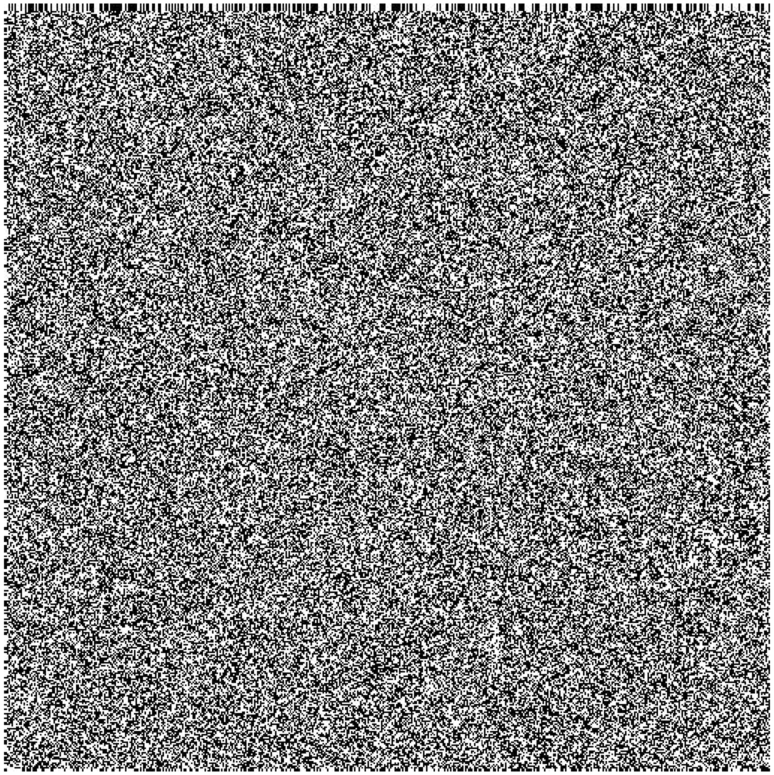
Example: Entropy of Lena LSB

No conditioning contexts

Entropy = **1.0** bit/pixel

7-neighbor conditioning context

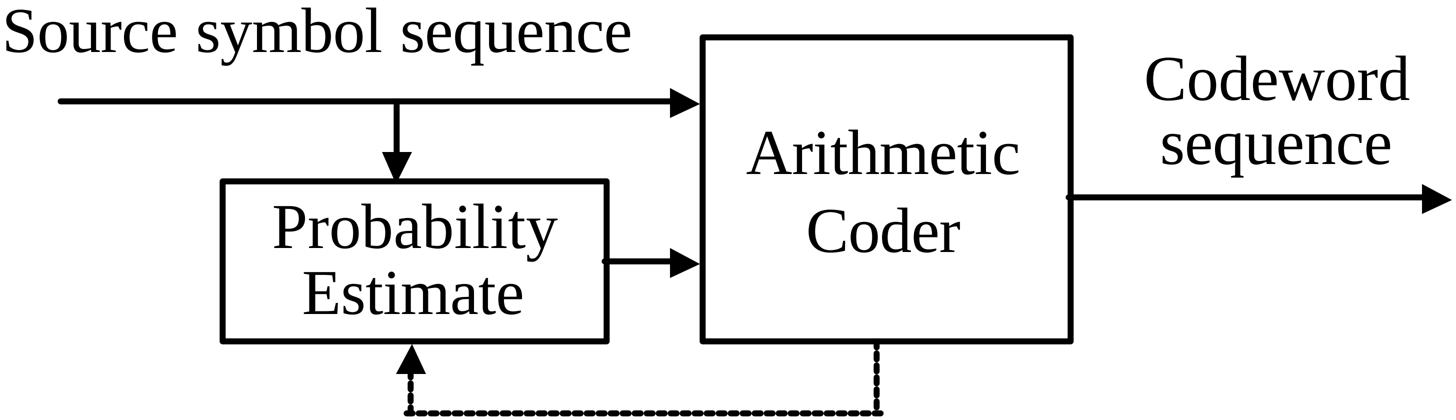
Entropy = **1.0** bits/pixel



Least significant bit plane

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**



Source symbol sequence

Probability Estimate

Arithmetic Coder

Codeword sequence

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.

$\text{YC}_\text{b}\text{C}_\text{r}$ Color Space

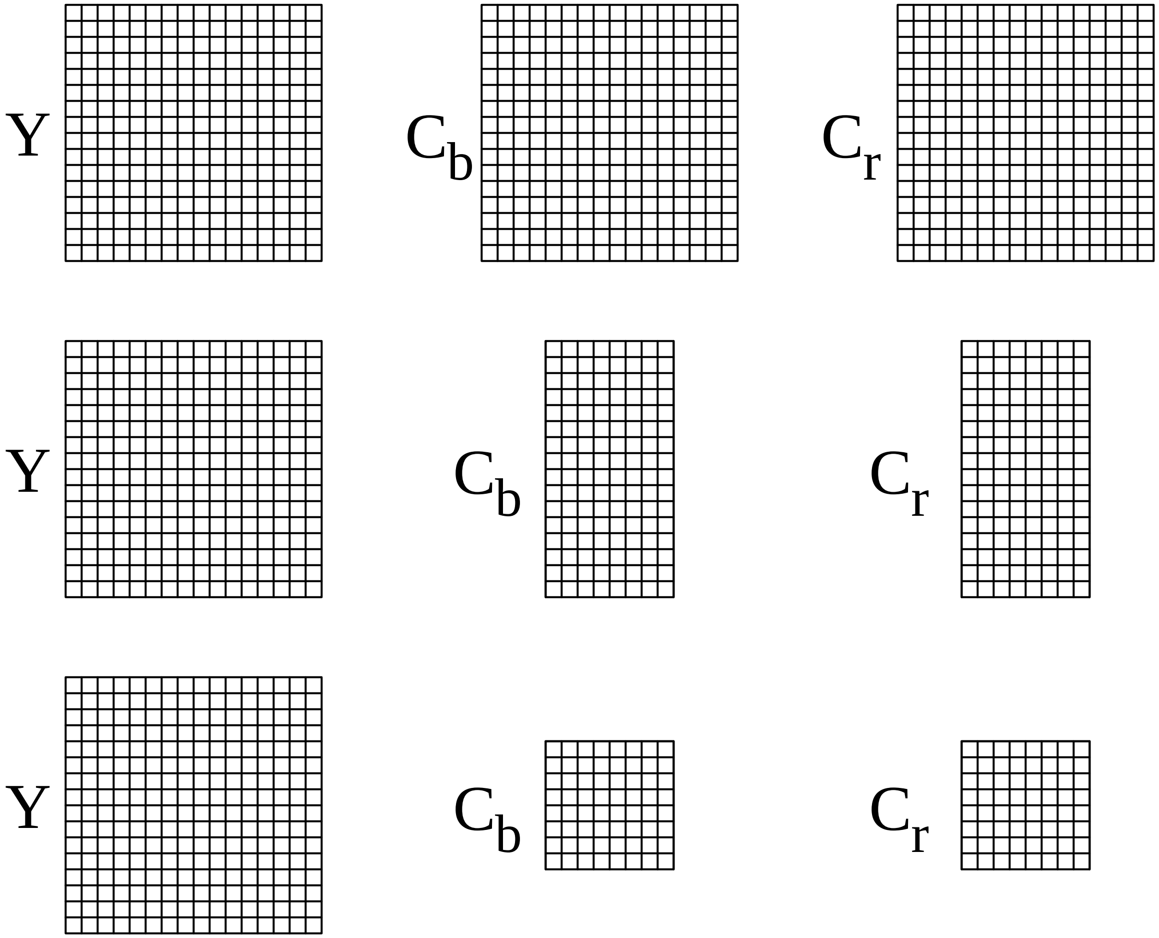
This is the most commonly used color coordinate system for the representation of image and video signals:

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

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

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

Chrominance Subsampling Formats

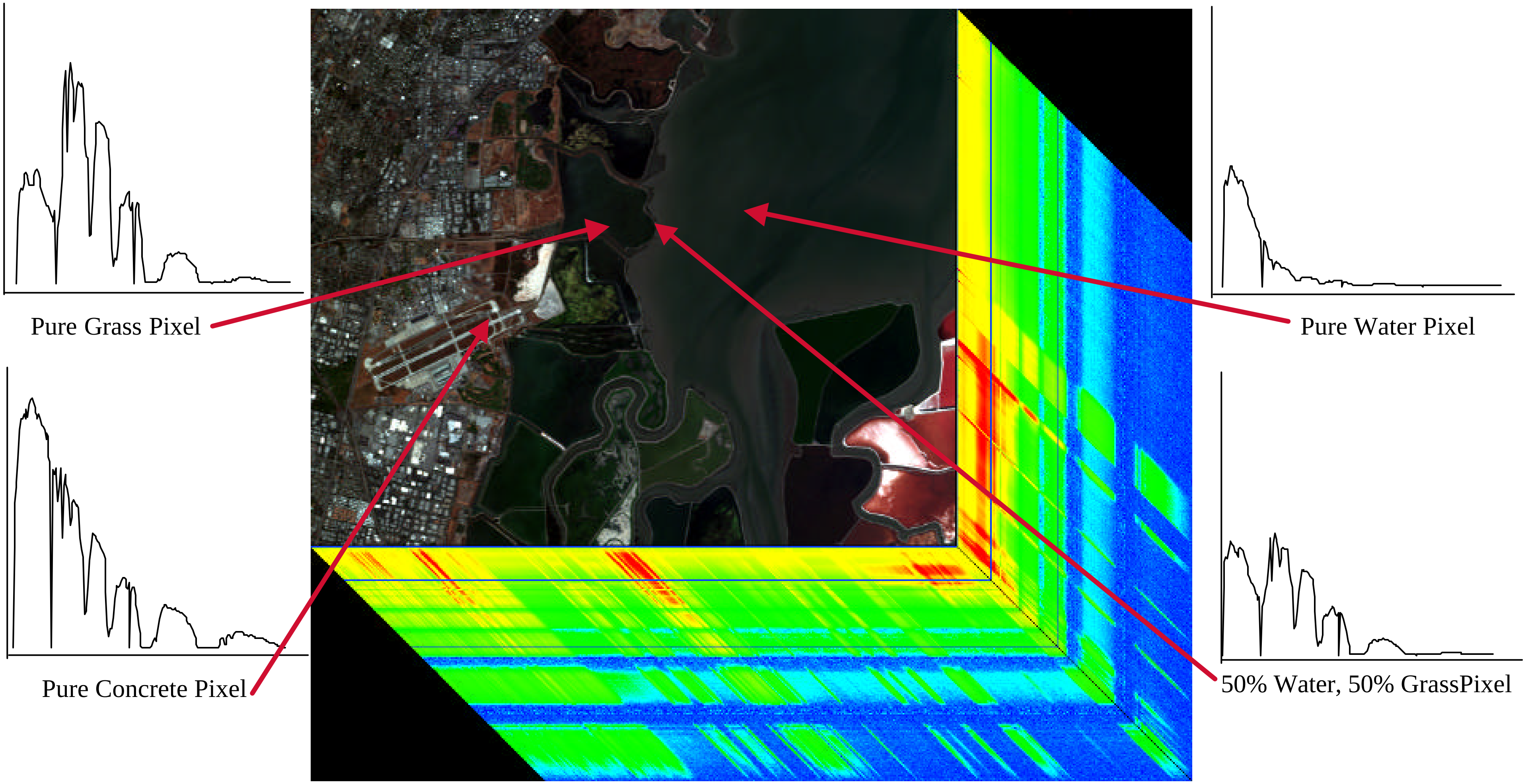


4:4:4
Format

4:2:2
Format

4:2:0
Format

Hyperspectral Information Cube (AVIRIS)



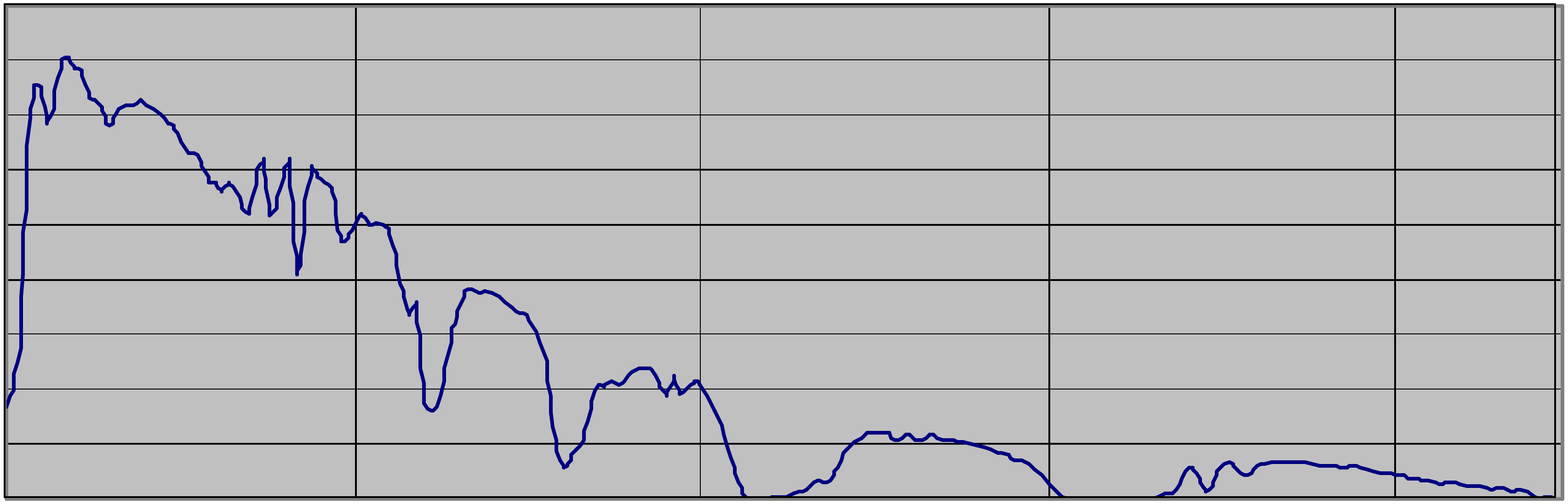
Pure Grass Pixel

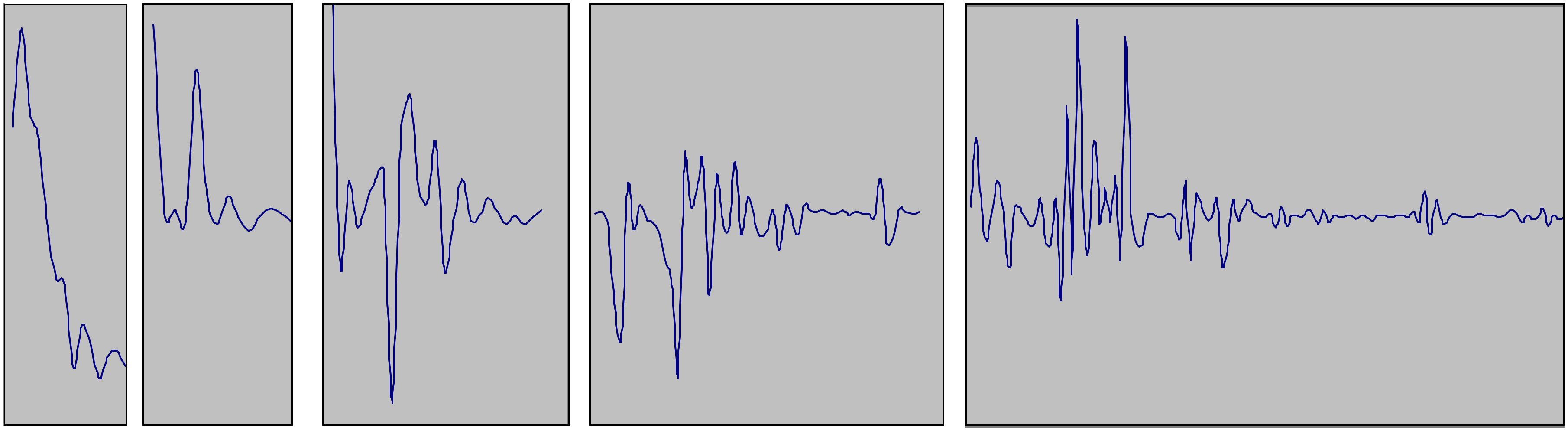
Pure Concrete Pixel

Pure Water Pixel

50% Water, 50% GrassPixel

Spectral Wavelet (Mean Signature)





Spectral Transforms

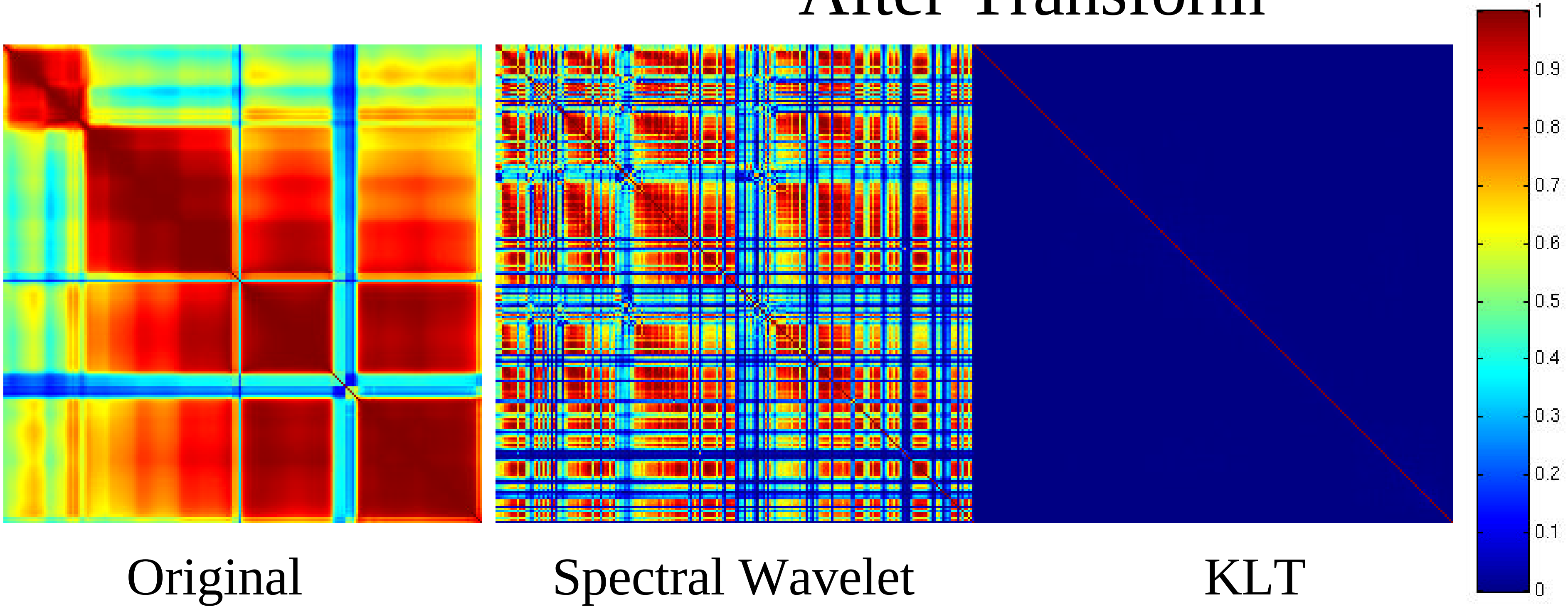
- Spectral Linear Prediction
  - Prediction for a band is generated by using a simple linear model:

$$\hat{x}_i = a \cdot \tilde{x}_{i-1} + b \cdot \tilde{x}_{i-2} + c$$

  - Coefficients are calculated using least-squares methods
  - Coefficients sent as overhead since the prediction is acausal
  - For lossy compression, the current band is fully reconstructed
- Principle Components/KLT
  - Generates custom coordinate axes based on measured data covariance, which maximize variance along each dimension
  - Optimal in terms of energy compaction
  - High complexity: covariance calculation + transform
    - Fast transforms (including integerized) are being researched

Spectral Correlation (AVIRIS)

After Transform



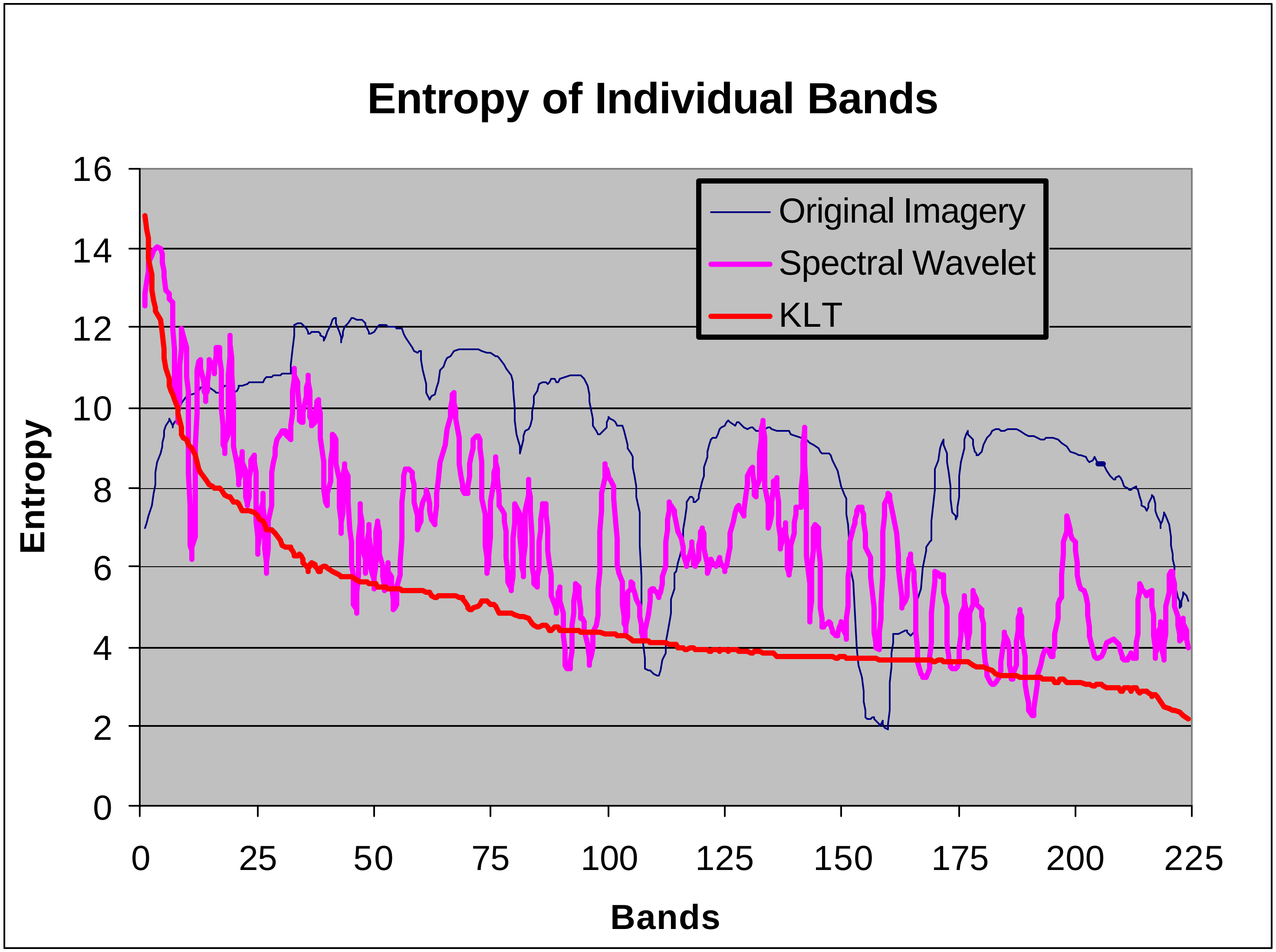
Original

Spectral Wavelet

KLT

Systems Engineering Services

Entropy of Bands



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

Image Compression Standards

- Binary (bi-level) images:
  - Group 3 & 4 (1980); JBIG (1994); JBIG2 (ongoing)
- Continuous-tone still images:
  - JPEG (1992); JPEG-LS (1998), JPEG-2000 (ongoing)
- Image sequences (moving pictures):
  - H.261 (1990); H.263 (1995); H.263+ (1997), H.263L
  - MPEG1 (1994); MPEG2 (1995);
  - MPEG4 (1999); MPEG7 (ongoing)

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

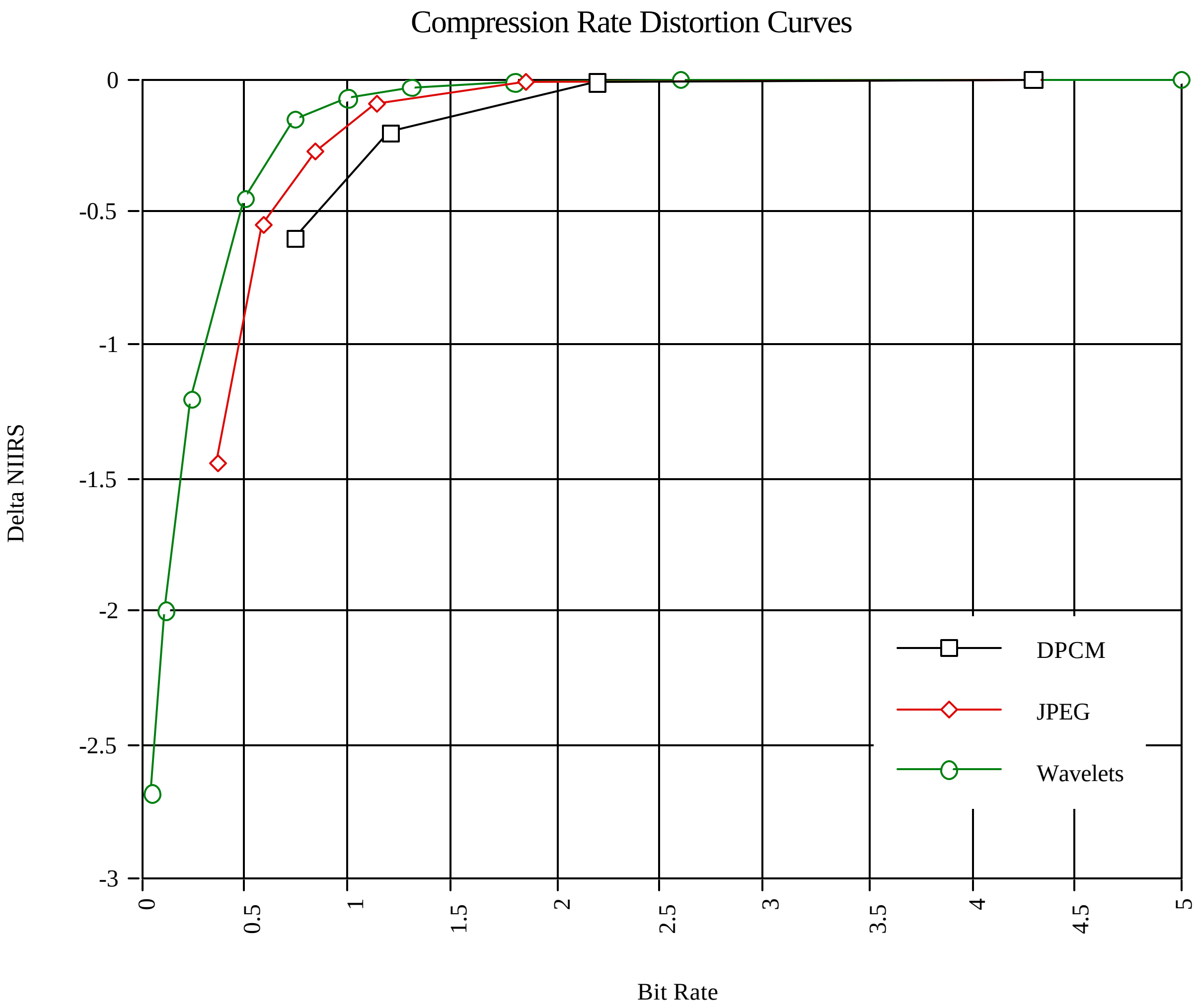
Compression Optimization

- 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.

NIMA

Systems Engineering Services

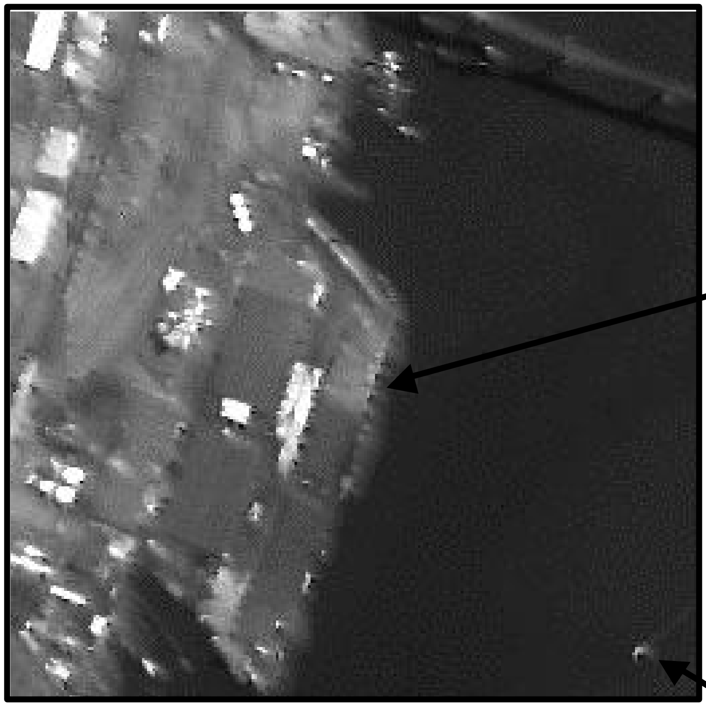
Compression Rate Distortion

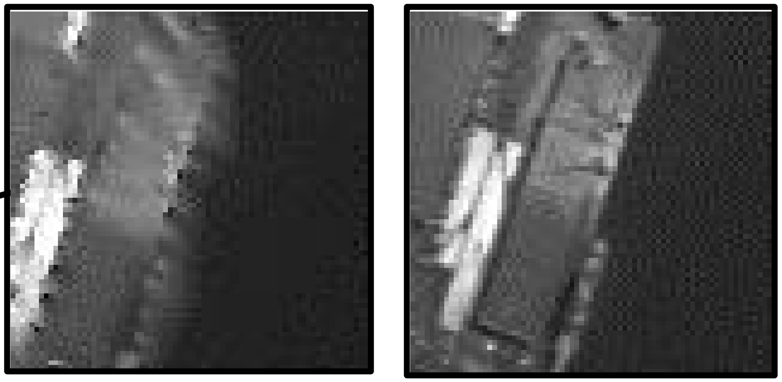


Compression Artifacts

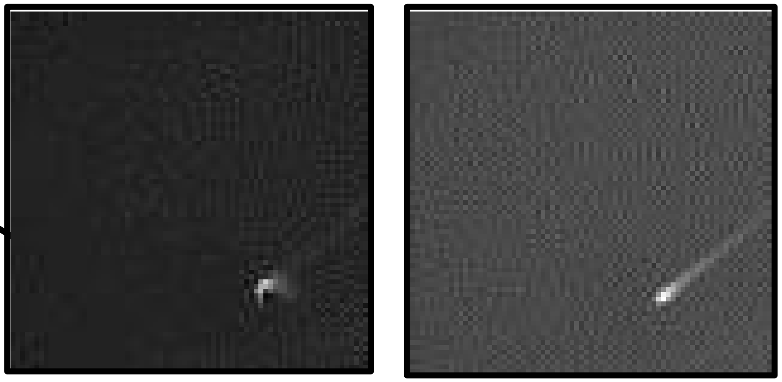
- Artifacts of compression are viewable when:
  - The compression ratio is pushed beyond the normal working environment of the given compression algorithm, or
  - the image is processed beyond the "normal" range of enhancements (i.e., sharpen, sharpen-more, DRA, TTC)
- Common artifacts include;
  - DPCM
    - 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)





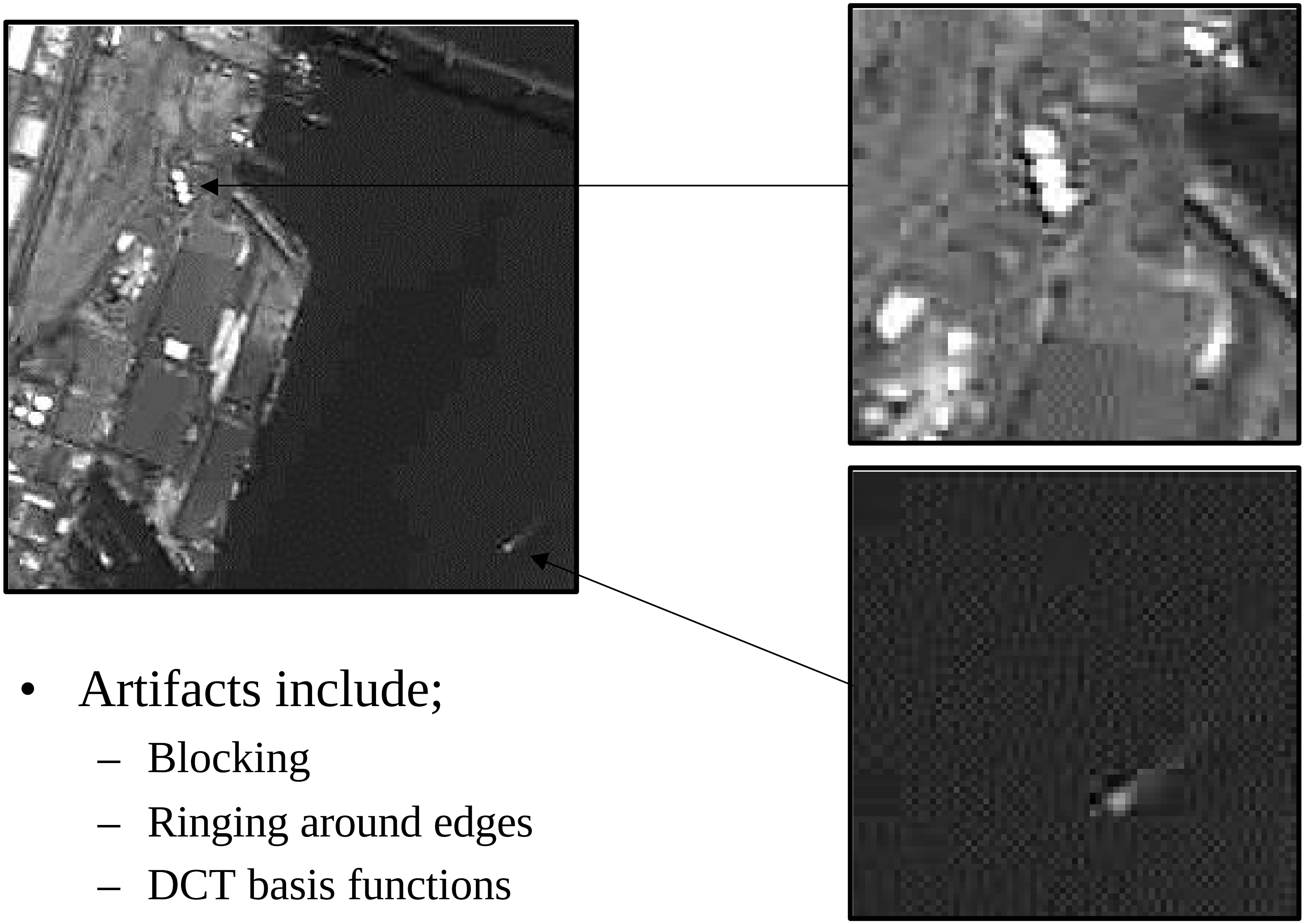
DPCM **Original**



DPCM **Original**

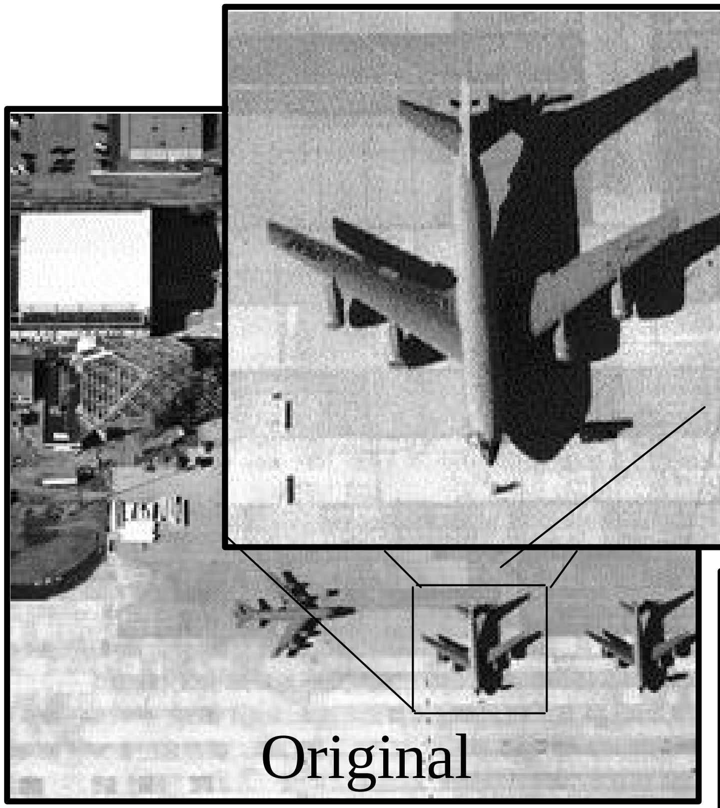
- Artifacts include;
  - Slope overload
  - Water-fall artifact

DCT Example (JPEG @ 0.4 bpp)



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

Wavelet Example (0.0625)


Original


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

Systems Engineering Services

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.

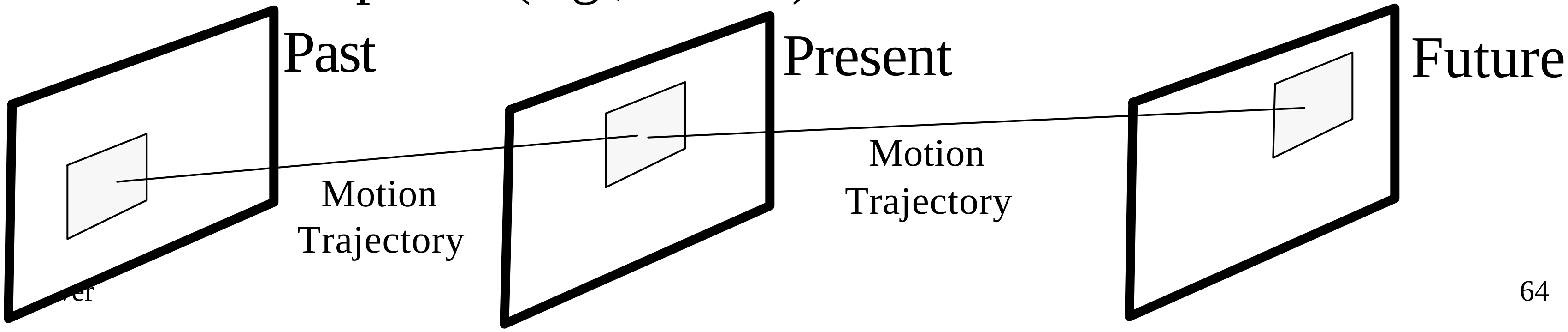
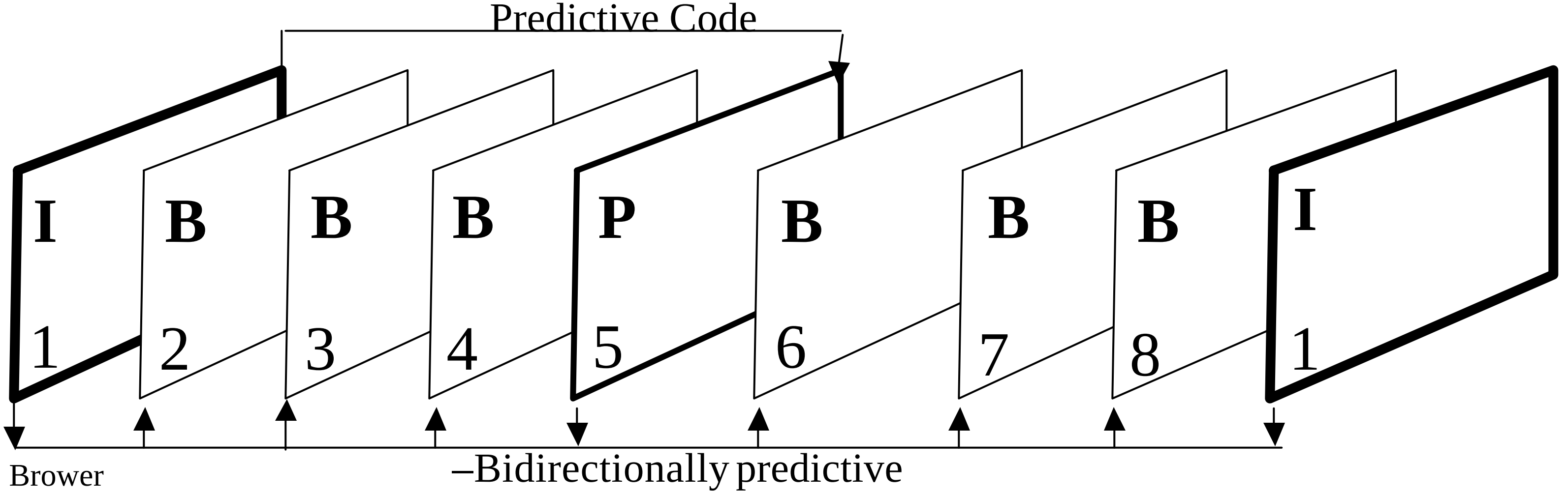


Image Sequence Compression (Video)

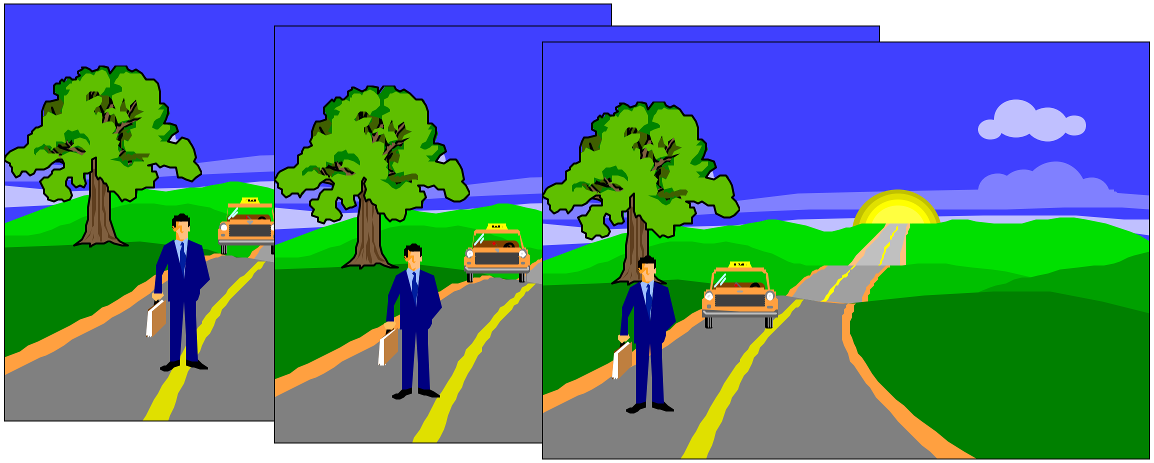
- The MPEG system specifies three types of frames within a sequence:
  - **Intra-coded picture** (I-frame): Coded independently from all other frames in the sequence. Uses the most number of bits.
  - **Predictive-coded picture** (P-frame): Coded based on a prediction from a past I- or P-frame. Uses 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)



Systems Engineering Services

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