

# DIGITAL IMAGE COMPRESSION



*Fernando Pereira*

*Instituto Superior Técnico*





# Dream or Nightmare ?

**“A image worths more than a thousand words” – visual information is an extremely powerful way to express a message or represent data.**

**Digital image systems generate huge amounts of data, e.g. many Megabytes for a single image.**



# Applications

★ Digital pictures



★ Image databases, e.g. museums, maps, various schemes, etc.

★ Desktop publishing

★ Colour fax

★ Medical images



★ ...

★ and Digital cinema



★ ...





## The Representation Problem ...

**A image is created and consumed as a set of  $M \times N$  luminance and chrominance samples with a certain number of bits per sample.**

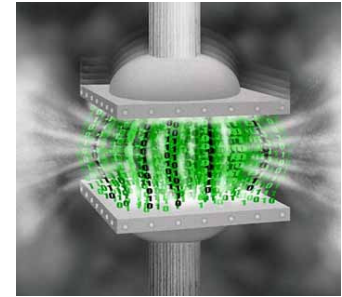
**Thus the total number of bits**

**- and so the memory and bandwidth – necessary to digitally represent an image is HUGE !!!**





# Image (Source) Coding Objective



**Image coding/compression deals with the efficient representation of images, satisfying the relevant requirements.**

**And these requirements keep changing, e.g., coding efficiency, error resilience, easy access, interaction, editing, to address new applications and functionalities ...**



# Where does Compression come from ?

★ **REDUNDANCY** – Regards the similarities, correlation and predictability of samples and symbols corresponding to the image/audio/video data.

-> redundancy reduction does not involve any information loss this means it is a reversible process -> *lossless coding*

★ **IRRELEVANCY** – Regards the part of the information which is imperceptible for the visual or auditory human systems.

-> irrelevancy reduction is an irreversible process -> *lossy coding*

**Source coding exploits these two concepts: for that, it is necessary to know the source statistics and the human visual/auditory systems characteristics.**



# Imagem Coding: Multiple Technical Solutions

★ DCT-based transform coding, e.g. JPEG standard

JPEG

★ Fractal-based coding

★ Vector quantization coding

★ Wavelet-based coding, e.g. JPEG 2000 standard

JPEG  
2000

★ Lapped biorthogonal-based transform coding, e.g. JPEG XR standard

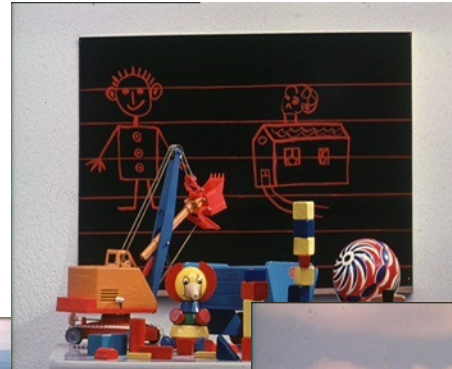
★ ...



# The JPEG Standard

(Joint Photographic Experts Group - ISO & ITU-T)

# Objective



**Definition of a generic compression standard for multilevel photographic images considering the requirements of most applications using.**

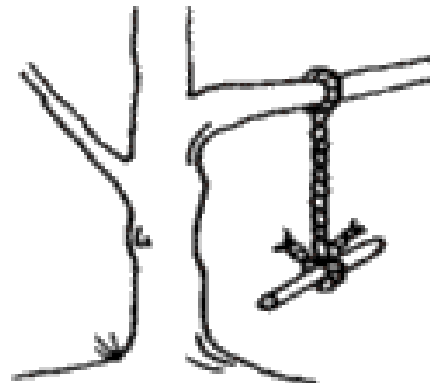


# Interoperability, thus Standards !

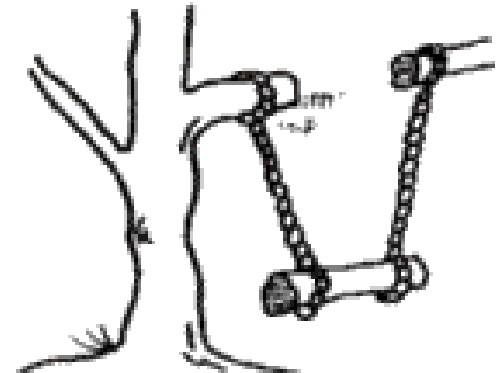
- ★ Image coding is used in the context of many applications where interoperability is an essential requirement.
- ★ The interoperability requirement is satisfied through the specification of **coding standards** which represent a voluntary agreement between multiple parties.
- ★ **In order to foster evolution and competition**, standards must offer interoperability through the specification of the smallest number of tools.

# The Importance of Good Requirements

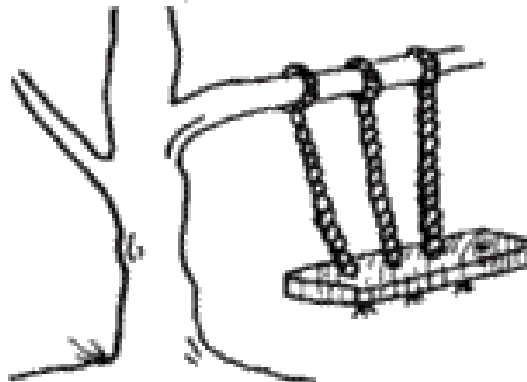
...



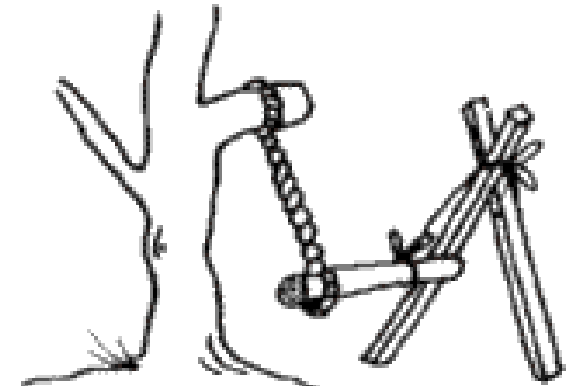
**What the user asked for**



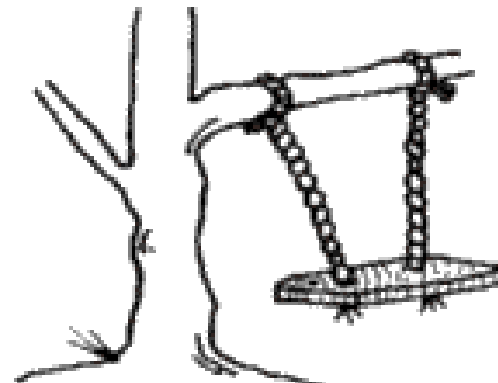
**How the analyst saw it**



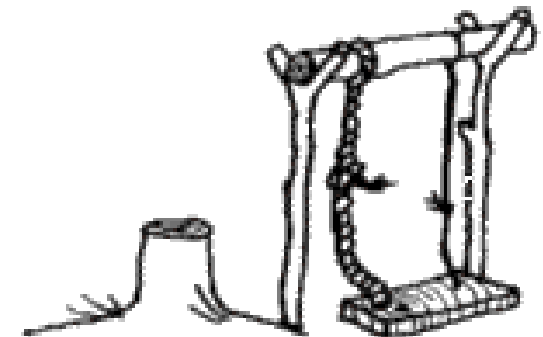
**How the system was designed**



**As the programmer wrote it**



**What the user really wanted**



**How it actually works**



# JPEG Standard Major Requirements

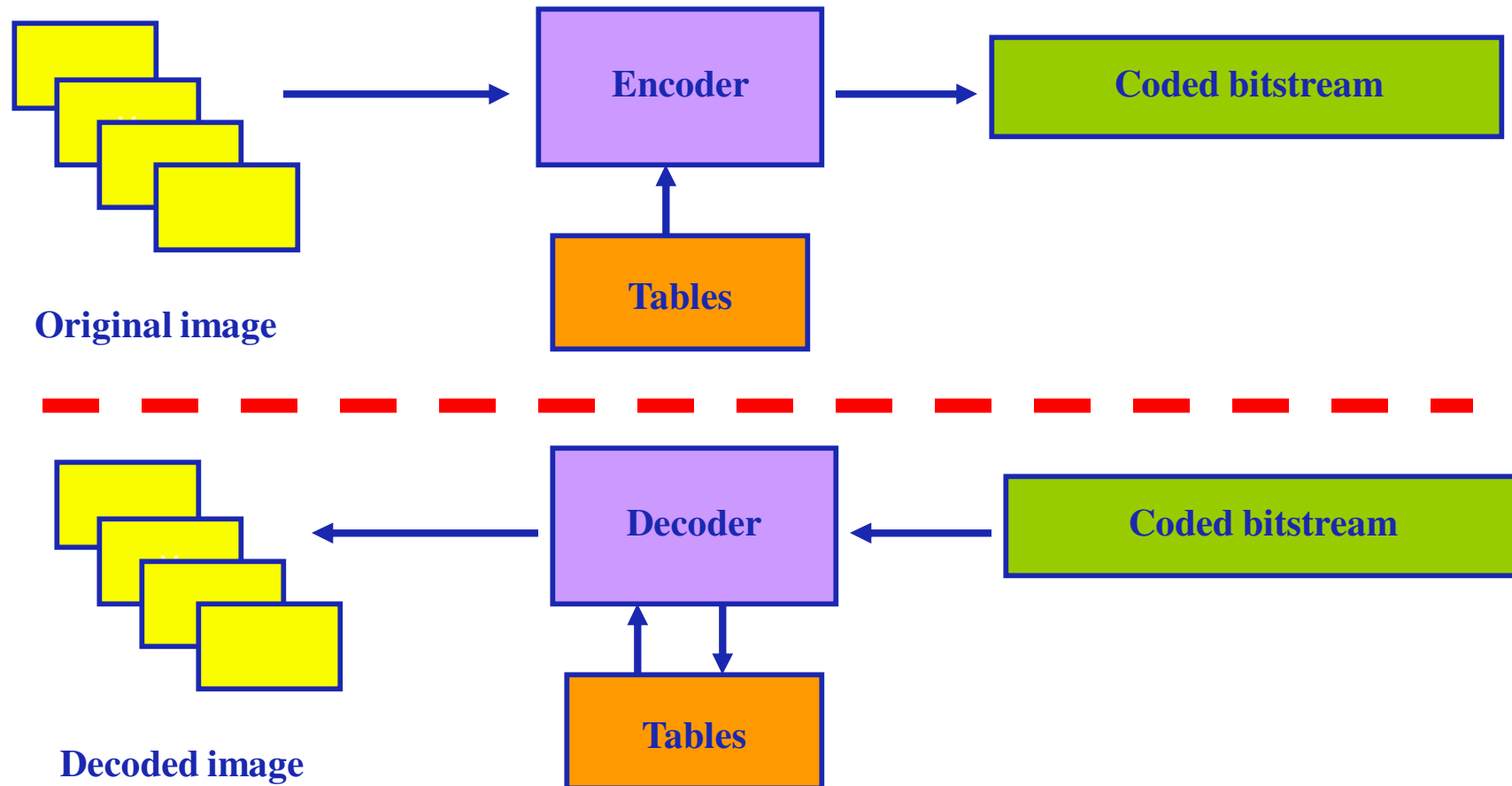
- ★ **Efficiency** - The standard must be based on the most efficient compression techniques, notably for very high quality.
- ★ **Compression/Quality Tunable** - The standard shall allow tuning the quality versus compression efficiency.
- ★ **Generic** - The standard must be applicable to any type of multilevel photographic images without restrictions in resolution, aspect ratio, color space, content, etc.
- ★ **Low Complexity** - The standard must be implementable with a reasonable complexity; notably, its software implementation on a large range of CPUs must be possible.
- ★ **Functional Flexibility** - The standard must provide various relevant operation modes, notably sequential, progressive, lossless and hierarchical.



# JPEG Normative Elements

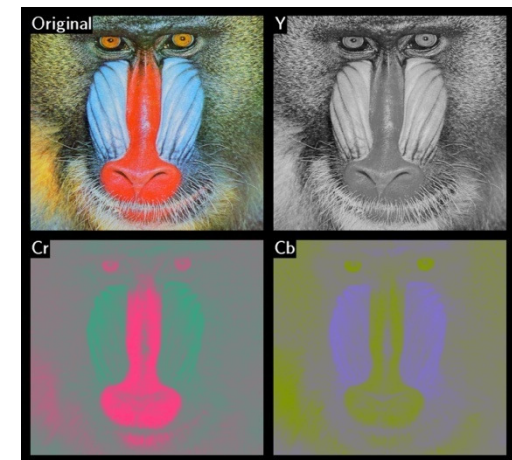
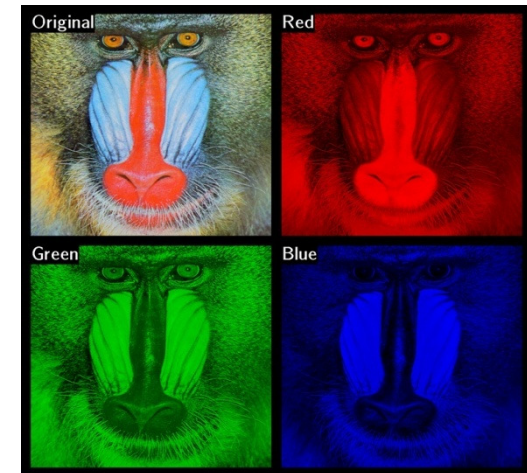
- ★ **ENCODER** – Based on the original input image and some tables, creates the coded bitstream using a certain number of compression techniques
- ★ **INTERCHANGE FORMAT** – Coded representation of the input image, including auxiliary tables if necessary
- ★ **DECODER** – Based on the coded bitstream and some tables, creates the decoded image using a certain number of decompression techniques

# JPEG Normative Elements



# What Images can JPEG Encode ?

- ★ Size between  $1 \times 1$  and  $65535 \times 65535$
- ★ 1 to 255 colour components or spectral bands
- ★ Each component,  $C_i$ , consists of a matrix with  $x_i$  columns and  $y_i$  lines
- ★ 8 or 12 bits per sample for DCT based compression
- ★ 2 to 16 bits per sample for lossless compression



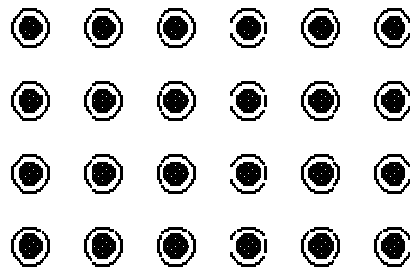


# ITU-R 601 Recommendation: a Typical Resolution

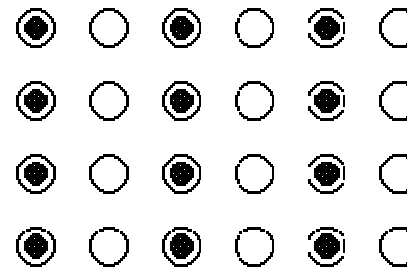
- ★ Most important standard PCM video format
- ★ Considers 625 and 525 lines systems (25 and 30 Hz) as well as 4:3 and 16:9 aspect ratios (576 lines for 25 Hz and 480 lines for 30 Hz systems)
- ★ Basic sampling rate: 13.5 MHz for the luminance and 6.75 MHz for the chrominances
- ★ Quantization: 8 bit/sample

Format	Resolution Y	Resolution U/V	Horizontal	Vertical
4:4:4	720 x 576	720 x 576	1:1	1:1
4:2:2	720 x 576	360 x 576	2:1	1:1
4:2:0	720 x 576	360 x 288	2:1	2:1
4:1:1	720 x 576	180 x 576	4:1	1:1
4:1:0	720 x 576	180 x 144	4:1	4:1

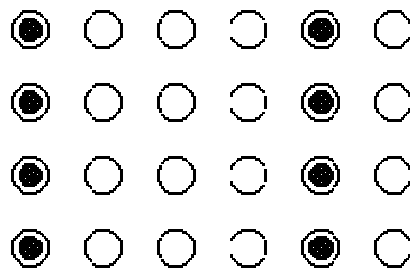
# Colour Subsampling Formats



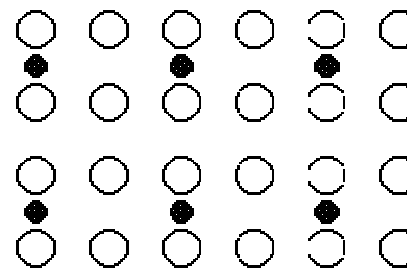
4:4:4






4:2:2



4:1:1

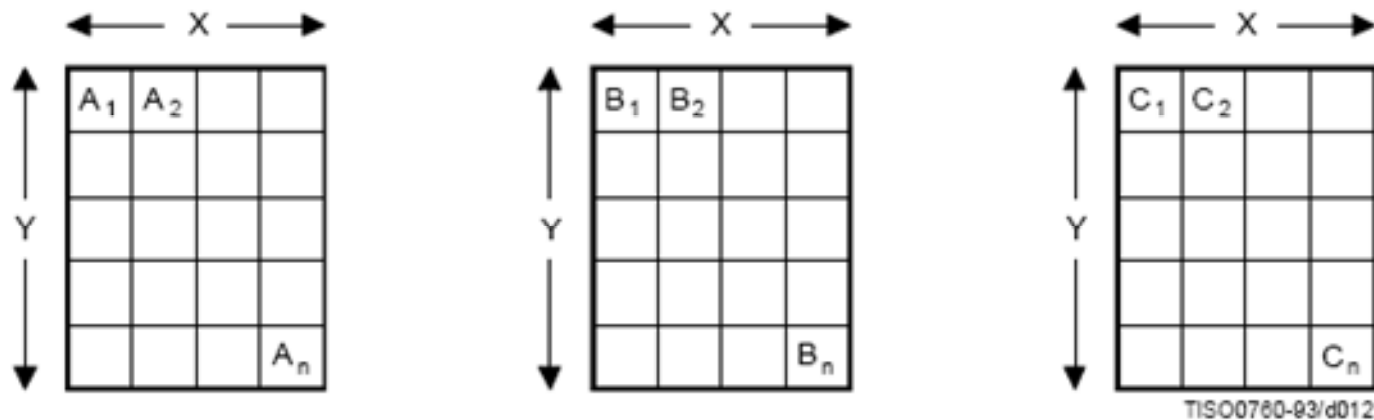


4:2:0

-  -- Pixel with only Y value
-  -- Pixel with only Cr and Cb values
-  -- Pixel with Y, Cr and Cb values

# Interleaving

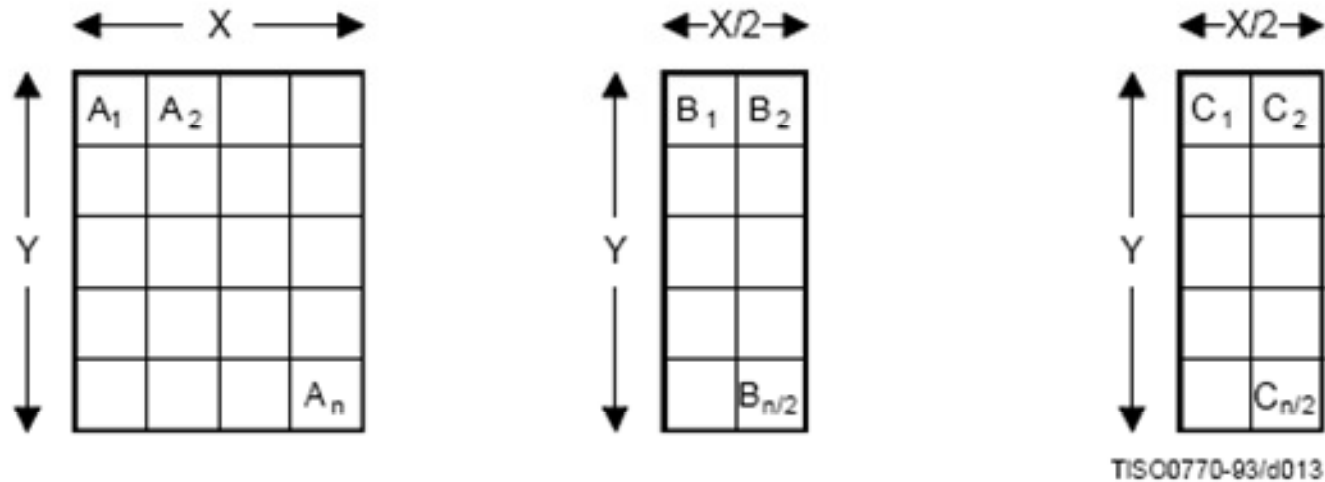
Since many applications must decode and visualize or print without large memory capacity, it is necessary to interleave the various image components with a finer granularity than the image level.



## Case 1: All components with the same resolution

Without interleaving (order):       $A_1, A_2, A_3, \dots, A_n \quad B_1, B_2, B_3, \dots, B_n \quad C_1, C_2, C_3, \dots, C_n$   
 With fine interleaving (order):       $A_1, B_1, C_1, A_2, B_2, C_2, A_3 \dots A_n, B_n, C_n$

# Interleaving



## Case 2: Components with different resolution

Without interleaving (order):  $A_1, A_2, A_3, \dots, A_n$   $B_1, B_2, B_3, \dots, B_{n/2}$   $C_1, C_2, C_3, \dots, C_{n/2}$

With fine interleaving (order):  $A_1, A_2, B_1, C_1, A_3, A_4, B_2, C_2, \dots, A_{n-1}, A_n, B_{n/2}, C_{n/2}$



# Types of Compression

- ★ **LOSSLESS** - The image is reconstructed with no losses, this means it is mathematically equal to the original; compression factors of about 2-3 may be achieved depending on the image content.
- ★ **LOSSY** – The image is reconstructed with losses but with a very high fidelity to the original, if desired (transparent coding); this type of coding allows to achieve higher compression factors, e.g. 10, 20 or more; in the JPEG standard, this type of coding is based on the Discrete Cosine Transform (DCT).

The most used JPEG coding solution is DCT based (lossy), called

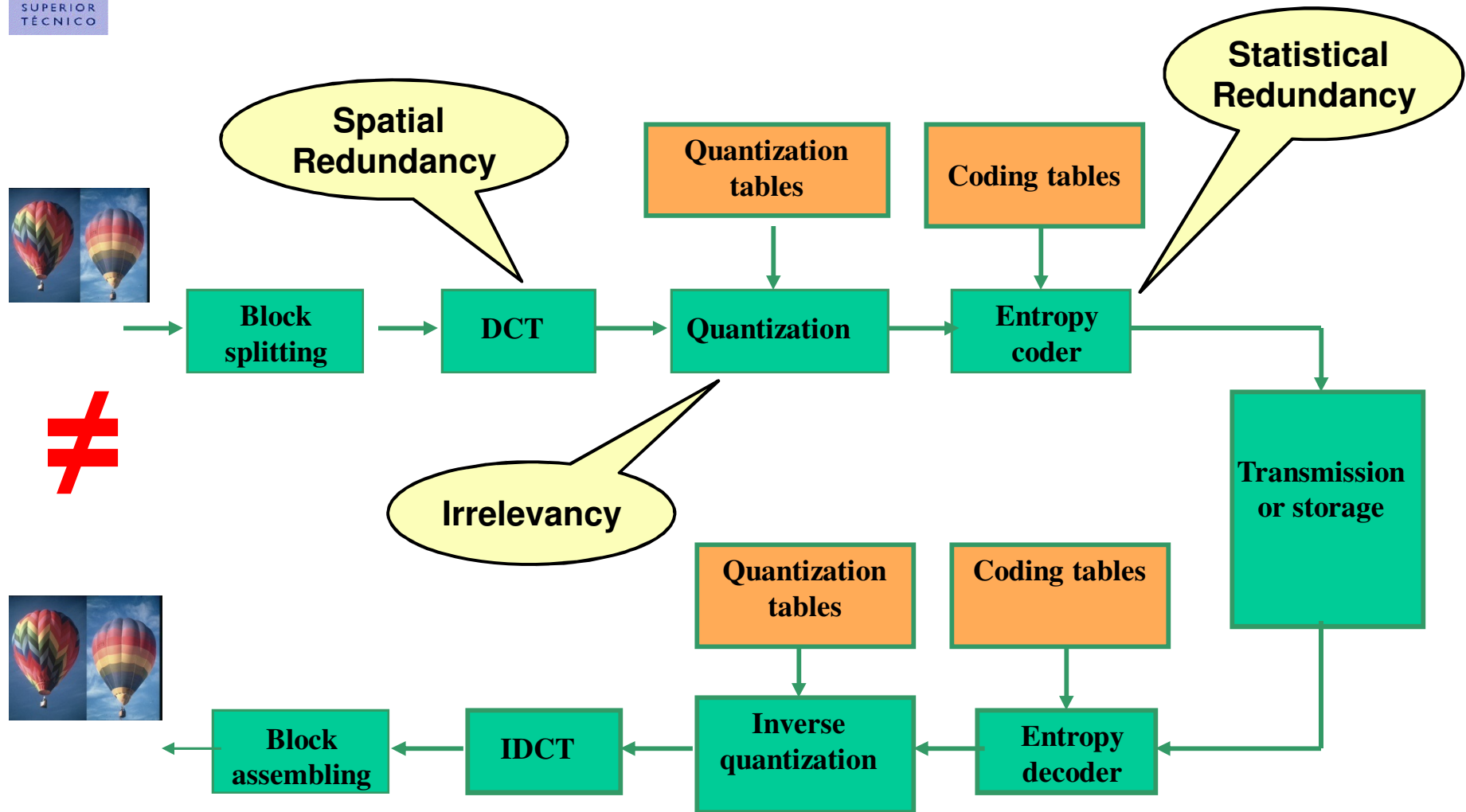
## **BASELINE SEQUENTIAL PROCESS**

and it is adequate to innumerable applications. This process is mandatory for all systems claiming JPEG compliance.



# JPEG Baseline Process

# DCT Based Image Coding



# Transform Coding

Transform coding involves the division of the image in blocks of  $N \times N$  samples to which the transform is applied, producing blocks with  $N \times N$  coefficients.

A transform is formally defined by its direct and inverse transform equations:

$$\begin{aligned}
 F(u,v) &= \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} f(i,j) A(i,j,u,v) \\
 f(i,j) &= \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u,v) B(i,j,u,v)
 \end{aligned}$$

Diagram annotations: A green oval labeled "Transform coefficients" points to  $F(u,v)$  in the first equation and  $F(u,v)$  in the second equation. A red oval labeled "Image block" points to  $f(i,j)$  in the first equation and  $f(i,j)$  in the second equation.

where

$f(i,j)$  – input signal (signal in space)

$A(i,j,u,v)$  – direct transform basis functions

$F(u,v)$  – transform coefficients (signal in frequency)

$B(i,j,u,v)$  – inverse transform basis functions



# Relevant Transform Characteristics

**Unitary transforms are used since they have the following characteristics:**

- ★ **Reversibility**
- ★ **Orthogonality of the transform basis functions**
- ★ **Energy conservation which means the energy in the transform domain is the same as in the spatial domain**

*Note 1: For unitary transforms,  $A^*A=AA^*=I_n$  where  $I_n$  is the identity matrix and  $*$  represents the transpose conjugate operation.*

*Note 2: The transpose matrix results by permuting the lines and columns and vice-versa which means that the transpose is a  $m \times n$  matrix if the original is a  $n \times m$  matrix.*

*Note 3: The conjugate matrix is obtained by substituting each element by its conjugate complex (imaginary part with changed signal).*



# What Shall the Transform Provide ?

- ★ **REVERSIBILITY** – The transform must be reversible since the image to transform has to be recovered again in the spatial domain.
- ★ **INCORRELATION** – The ideal transform shall provide coefficients which are uncorrelated this means each one carries additional/novel information.
- ★ **ENERGY COMPACTATION** – The major part of the signal energy shall be compacted in a small number of coefficients.
- ★ **IMAGE INDEPENDENT TRANSFORM BASIS FUNCTIONS** – Since images show significant statistical variations, the optimal transform should be image dependent; however, the use of image dependent transforms would require its computation as well as its storage and transmission; thus, an image independent transform is desirable even if at some cost in coding efficiency.
- ★ **LOW COMPLEXITY IMPLEMENTATIONS** – Due to the high number of operations involved, the transform shall allow low complexity/fast implementations.

# How to Interpret a Transform ?



The formula for the inverse transform

$$f(i,j) = \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} F(u,v) \cdot B(i,j,u,v)$$

Weights

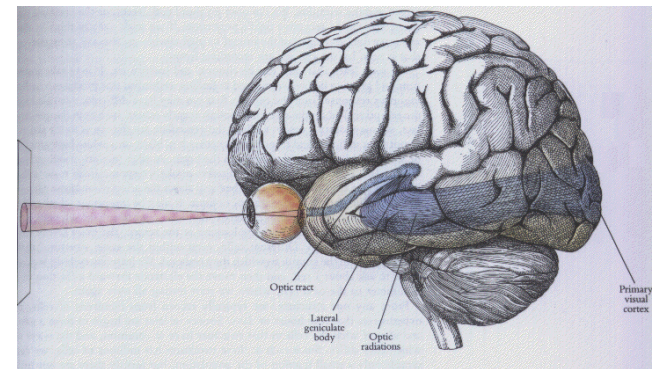
Basic image  
blocks

expresses that the transform may be interpreted as a decomposition of the image in terms of certain basic functions – the transform basis functions – adequately weighted by the transform coefficients.

*The Spectral Interpretation – As most transforms use basis functions with different frequencies (in a broad sense), the decomposition in basis functions through the transform coefficients assumes a spectral meaning where each coefficient represents the fraction of energy in the image corresponding to a certain basis function/frequency.*

# Advantages of the Spectral Interpretation

**The spectral interpretation allows to easily introduce in the coding process some relevant characteristics of the human visual system which are essential for efficient (lossy) coding.**



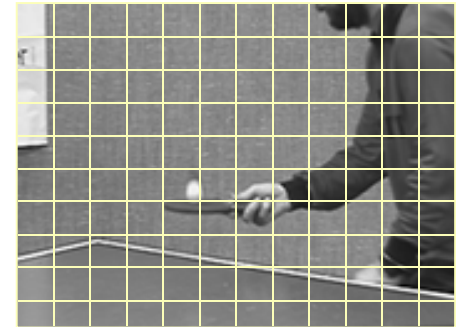
★ **The human visual system is less sensitive to the high spatial frequencies**

**->> coarser coding (through quantization) of the corresponding transform coefficients**

★ **The human visual system is less sensitive to very low or very high luminances**

**->> coarser coding (through quantization) of the DC coefficient for these conditions**

# Why do we Transform Blocks ?

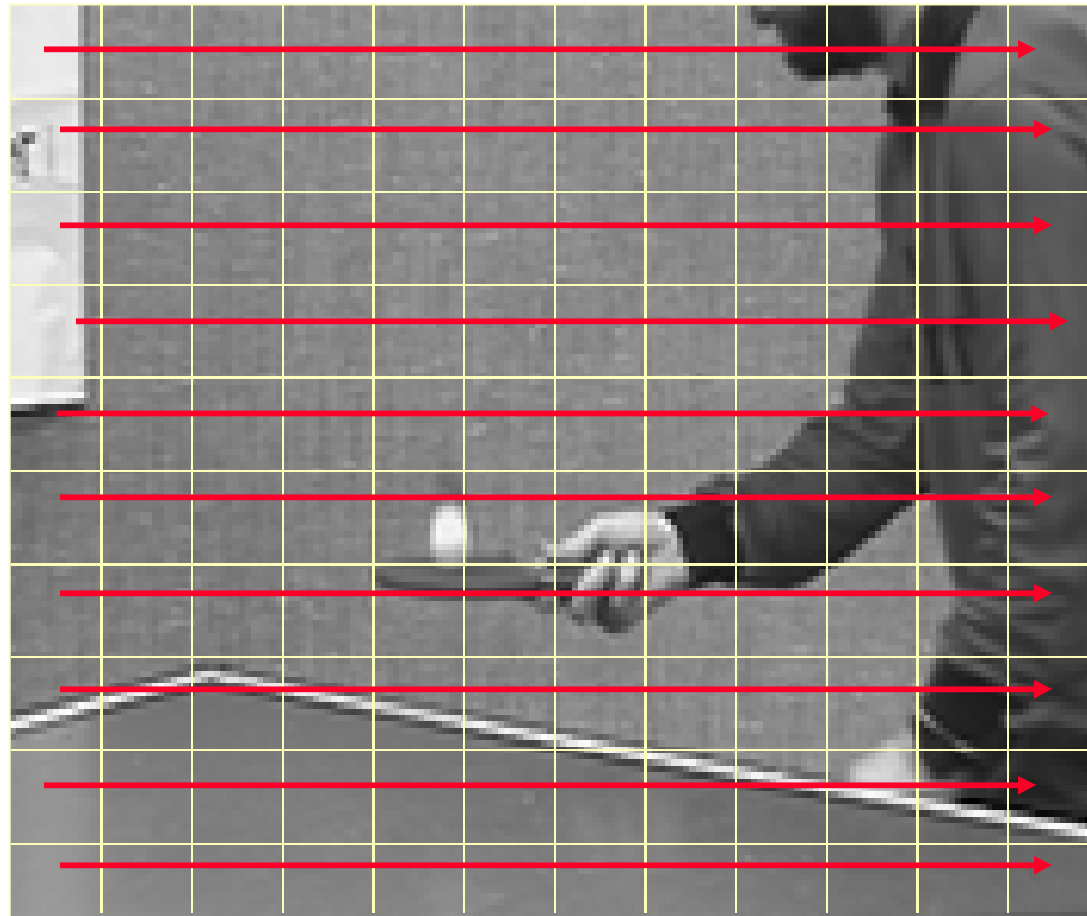


**Basically, the transform represents the original signal in another domain where it can be more efficiently coded by exploiting the spatial redundancy.**

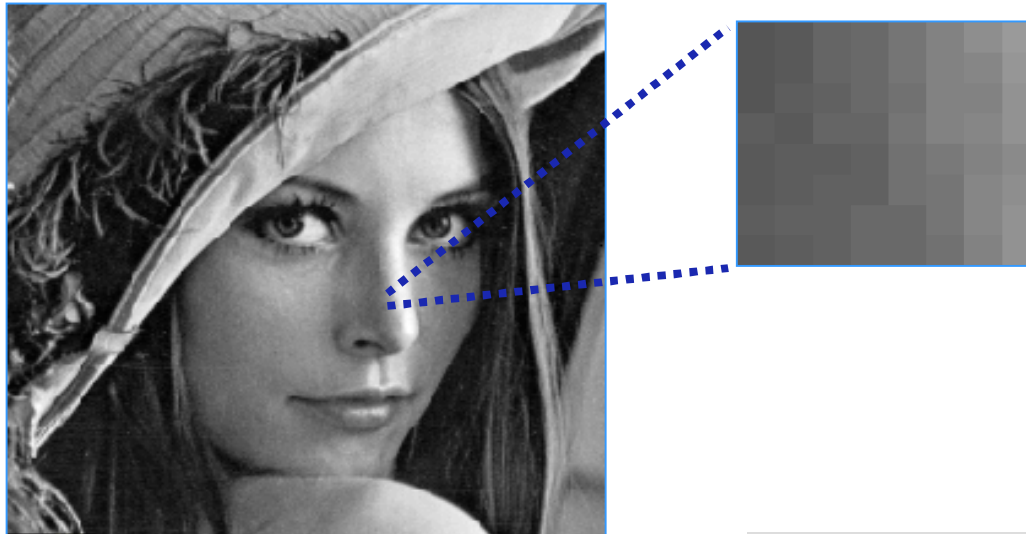
- ★ **The full exploitation of the spatial redundancy in the image would require applying the transform to blocks as big as possible, ideally to the full image.**
- ★ **However, the computational effort associated to the transform grows quickly with the size of the block used ... and the added spatial redundancy decreases ...**

**Applying the transform to blocks, typically of  $8 \times 8$  samples, is a good trade-off between the exploitation of the spatial redundancy and the associated computational effort.**

# JPEG Block Coding Sequence



# What is it Transformed ?



Y =

87	89	101	106	118	130	142	155
85	91	101	105	116	129	135	149
86	92	96	105	112	128	131	144
92	88	102	101	116	129	135	147
88	94	94	98	113	122	130	139
88	95	98	97	113	119	133	141
92	99	98	106	107	118	135	145
89	95	98	107	104	112	130	144

**Same (in parallel) for the chrominances !**



Luminance  
Samples,  $Y =$

87	89	101	106	118	130	142	155
85	91	101	105	116	129	135	149
86	92	96	105	112	128	131	144
92	88	102	101	116	129	135	147
88	94	94	98	113	122	130	139
88	95	98	97	113	119	133	141
92	99	98	106	107	118	135	145
89	95	98	107	104	112	130	144

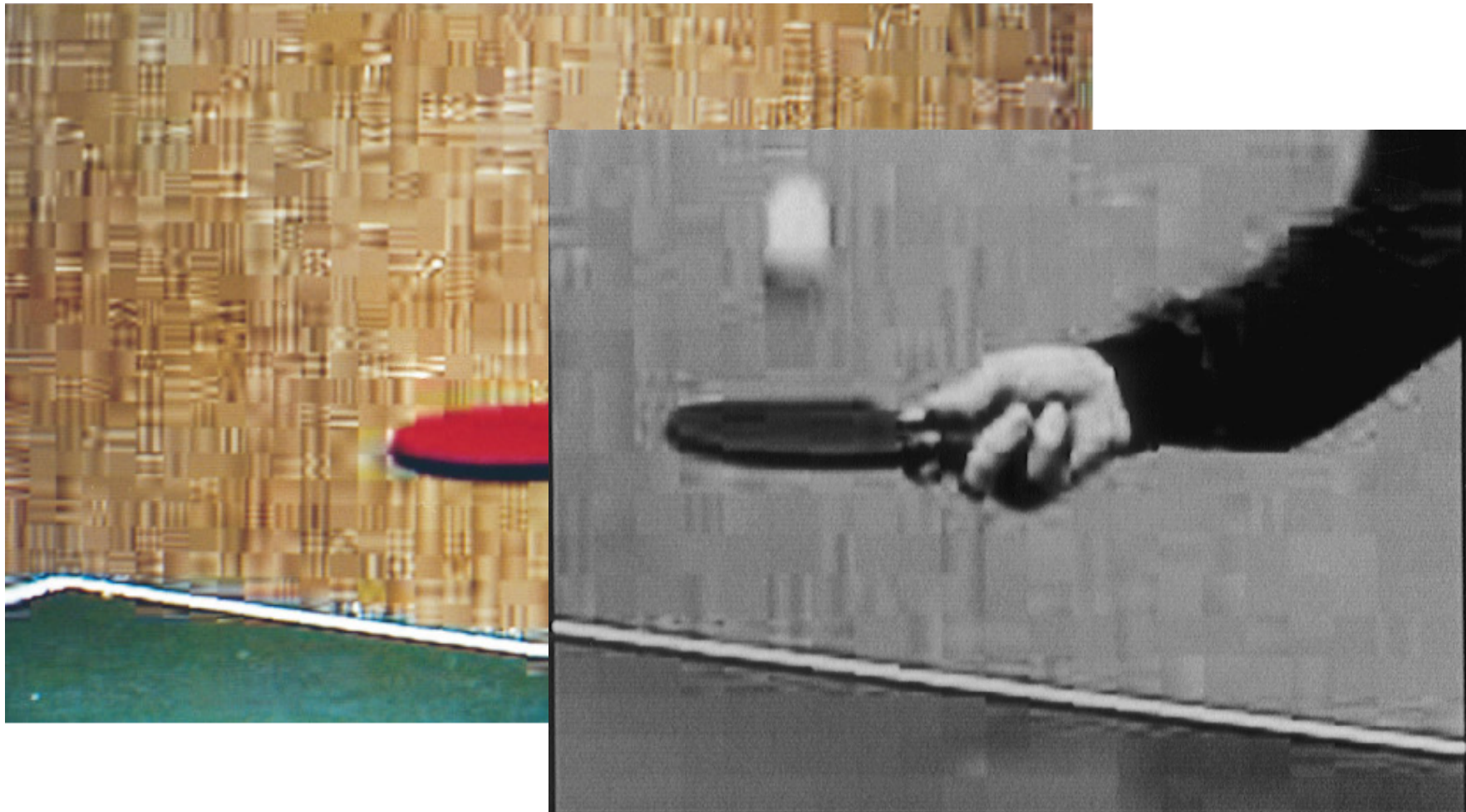
↓  
Transform  
↓

Transform  
Coefficients =

898.0000	-149.5418	26.6464	-14.0897	0.7500	-5.7540	3.5750	0.0330
12.1982	-16.5235	-7.6122	5.2187	-0.2867	-1.9909	8.4265	1.2591
5.3355	-2.6557	2.3410	-9.9277	2.4614	4.4558	-3.1945	-3.1640
1.9463	-2.7271	1.5106	2.8421	-2.1336	-2.7203	-2.7510	5.4051
0.7500	-2.0745	0.8610	0.2085	2.5000	1.8446	2.0787	2.4750
7.9536	-2.6624	2.6308	0.4010	0.4772	3.3000	1.7394	0.3942
-4.1042	-0.1650	-0.6945	0.0601	0.0628	-0.7874	-0.8410	0.3496
-3.4688	2.3804	0.1559	0.8696	0.1142	-0.5240	-3.9974	-5.6187



# The Block Effect ...





# Karhunen-Loève Transform (KLT)

The Karhunen-Loève Transform is typically considered the ideal transform because it achieves the

## MAXIMUM ENERGY COMPACTATION

this means, if a certain limited number of coefficients is coded, the KLT coefficients are always those containing the highest percentage of the total signal energy.

The KLT base functions are based on the eigen vectors of the covariance matrix for the image blocks.

## Why is KLT Never Used ?



**The use of KLT for image compression is practically irrelevant because:**

- ★ **KLT basis functions are image dependent requiring the computation of the image covariance matrix as well as its storage or transmission.**
- ★ **There are no fast algorithms for its computation.**
- ★ **There are other transforms without the drawbacks above but still with a energy compactation performance only slightly lower than that of KLT.**



# Discrete Cosine Transform (DCT)

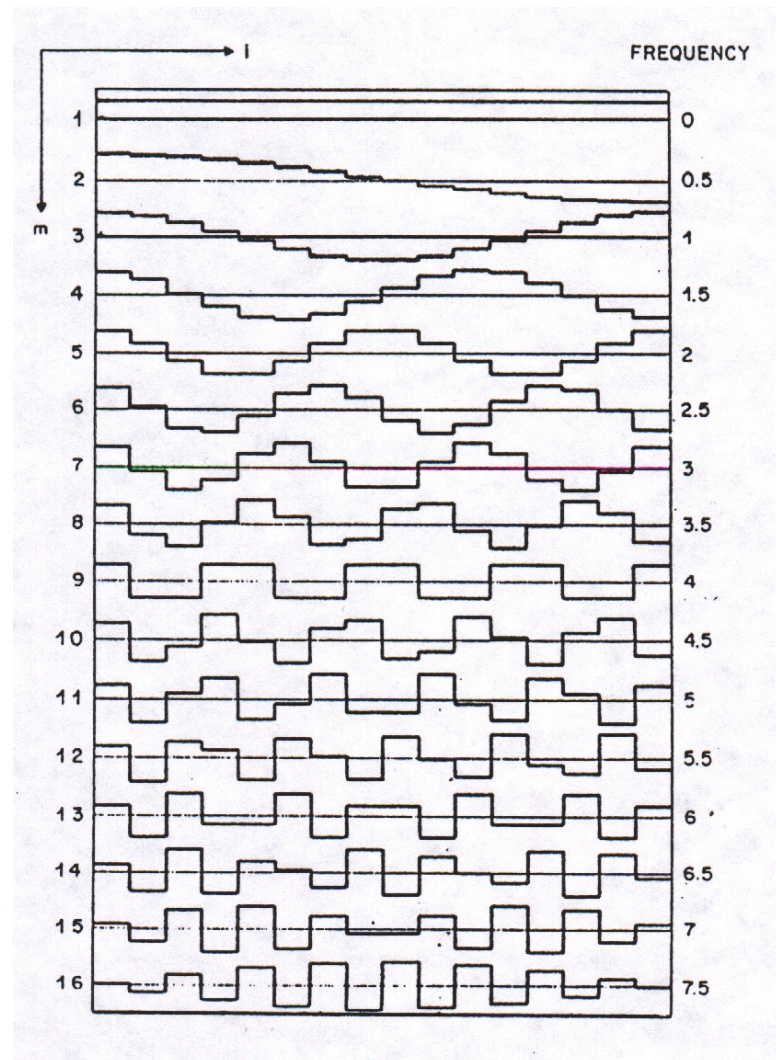
The DCT is one of the several sinusoidal transforms available; its basis functions correspond to discretized sinusoidal functions.

$$F(u, v) = \frac{2}{N} C(u) C(v) \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} f(j, k) \cos\left(\pi \frac{u(2j+1)}{2N}\right) \cos\left(\pi \frac{v(2k+1)}{2N}\right)$$

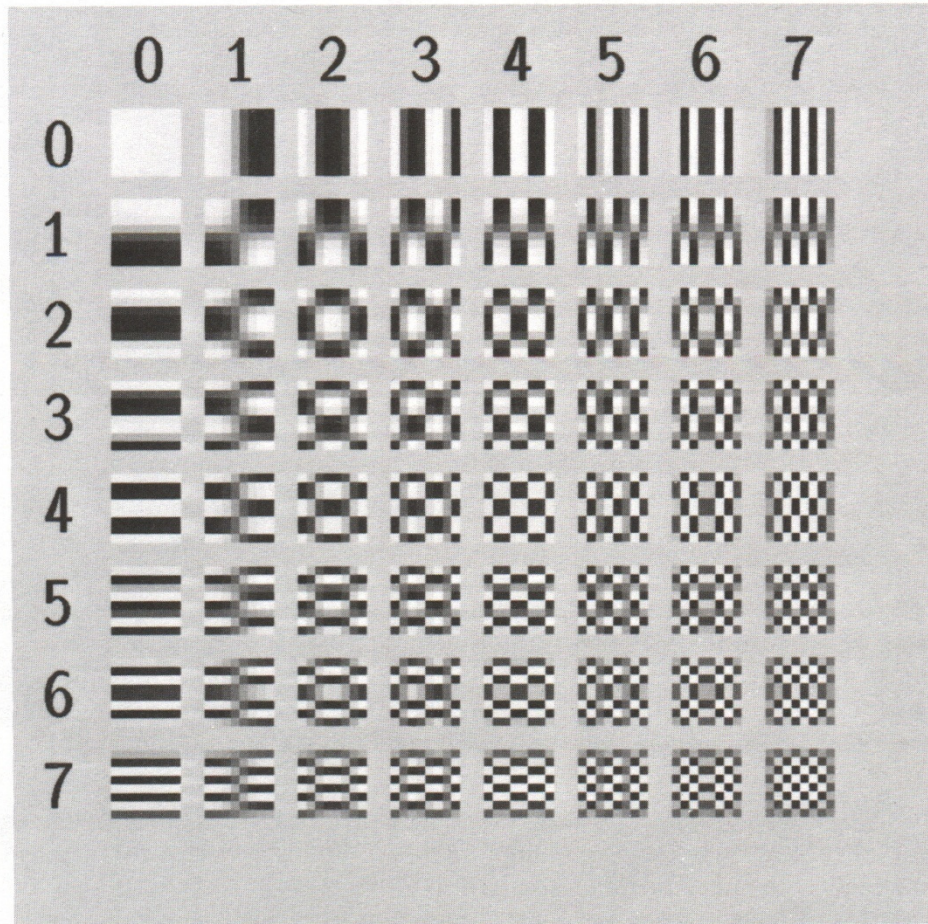
$$f(j, k) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u) C(v) F(u, v) \cos\left(\frac{u(2j+1)}{2N} \pi\right) \cos\left(\frac{v(2k+1)}{2N} \pi\right)$$

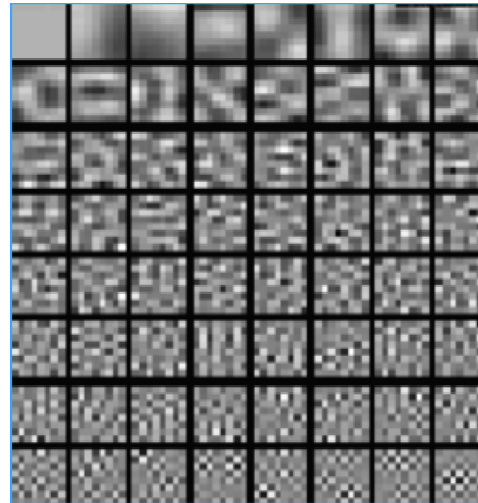
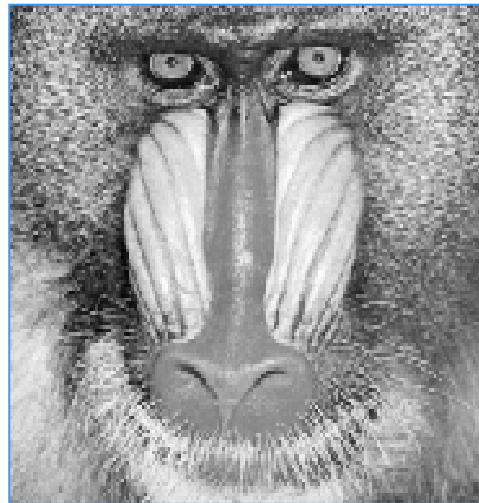
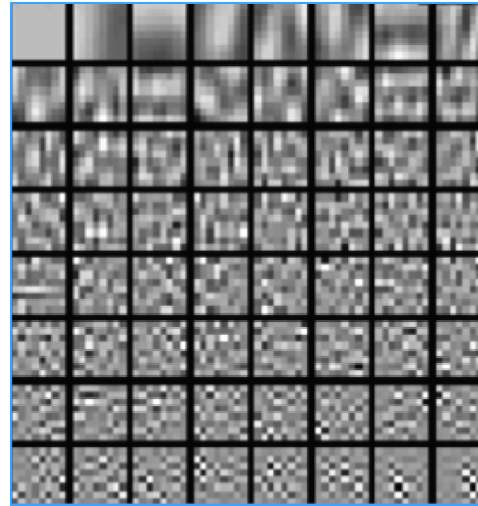
The DCT is the most used transform for image and video compression since its performance is close to the KLT performance for highly correlated signals; moreover, there are fast implementation algorithms available.

# DCT Unidimensional Basis Functions ( $N=8$ )



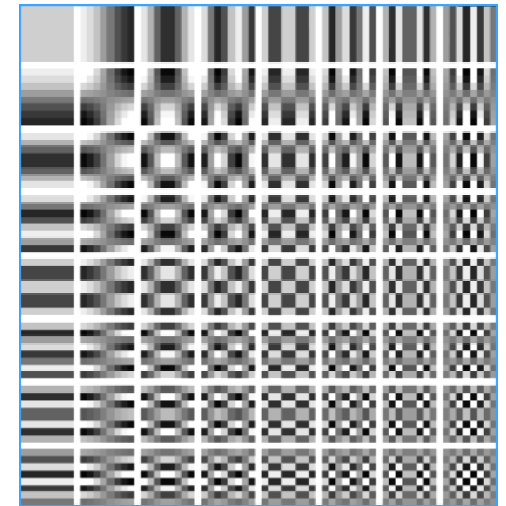
# DCT Bidimensional Basis Functions (N=8)





KLT

**DCT: Same basis functions for any image block !**

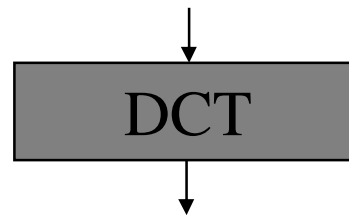


DCT



Luminance  
Samples, Y =

87	89	101	106	118	130	142	155
85	91	101	105	116	129	135	149
86	92	96	105	112	128	131	144
92	88	102	101	116	129	135	147
88	94	94	98	113	122	130	139
88	95	98	97	113	119	133	141
92	99	98	106	107	118	135	145
89	95	98	107	104	112	130	144



DCT  
Coefficients =

898.0000	-149.5418	26.6464	-14.0897	0.7500	-5.7540	3.5750	0.0330
12.1982	-16.5235	-7.6122	5.2187	-0.2867	-1.9909	8.4265	1.2591
5.3355	-2.6557	2.3410	-9.9277	2.4614	4.4558	-3.1945	-3.1640
1.9463	-2.7271	1.5106	2.8421	-2.1336	-2.7203	-2.7510	5.4051
0.7500	-2.0745	0.8610	0.2085	2.5000	1.8446	2.0787	2.4750
7.9536	-2.6624	2.6308	0.4010	0.4772	3.3000	1.7394	0.3942
-4.1042	-0.1650	-0.6945	0.0601	0.0628	-0.7874	-0.8410	0.3496
-3.4688	2.3804	0.1559	0.8696	0.1142	-0.5240	-3.9974	-5.6187



# DCT in JPEG

Since the DCT uses sinusoidal functions, it is impossible to perform computations with full precision. This leads to (slight) differences in the results for different implementations (mismatch).

- ★ In order to accommodate future implementation developments, the JPEG recommendation does not specify any specific DCT or IDCT implementation.
- ★ The JPEG recommendation specifies a fidelity/accuracy test in order to limit the differences caused by the freedom in terms of DCT and IDCT implementation.

*Note: The DCT is applied to the signal samples with  $P$  bits, with values between  $-2^{P-1}$  and  $2^{P-1}-1$  in order the DC coefficient is distributed around zero.*

# How Does the DCT Work ?

Spatial Domain

X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X
X	X	X	X	X	X	X	X

Frequency Domain

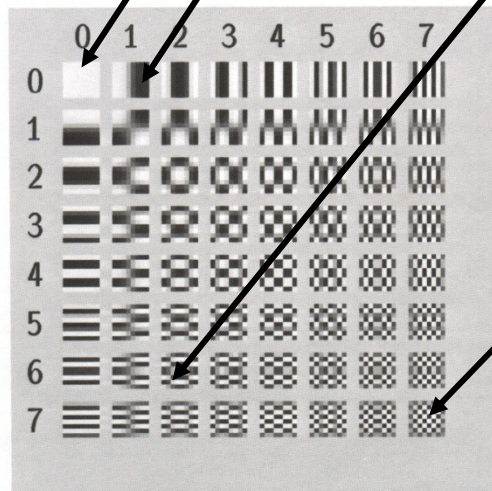
X	X						
X	X	X	X				
					X	X	
				X	X		
X	X	X					
		X	X				
							X

DCT

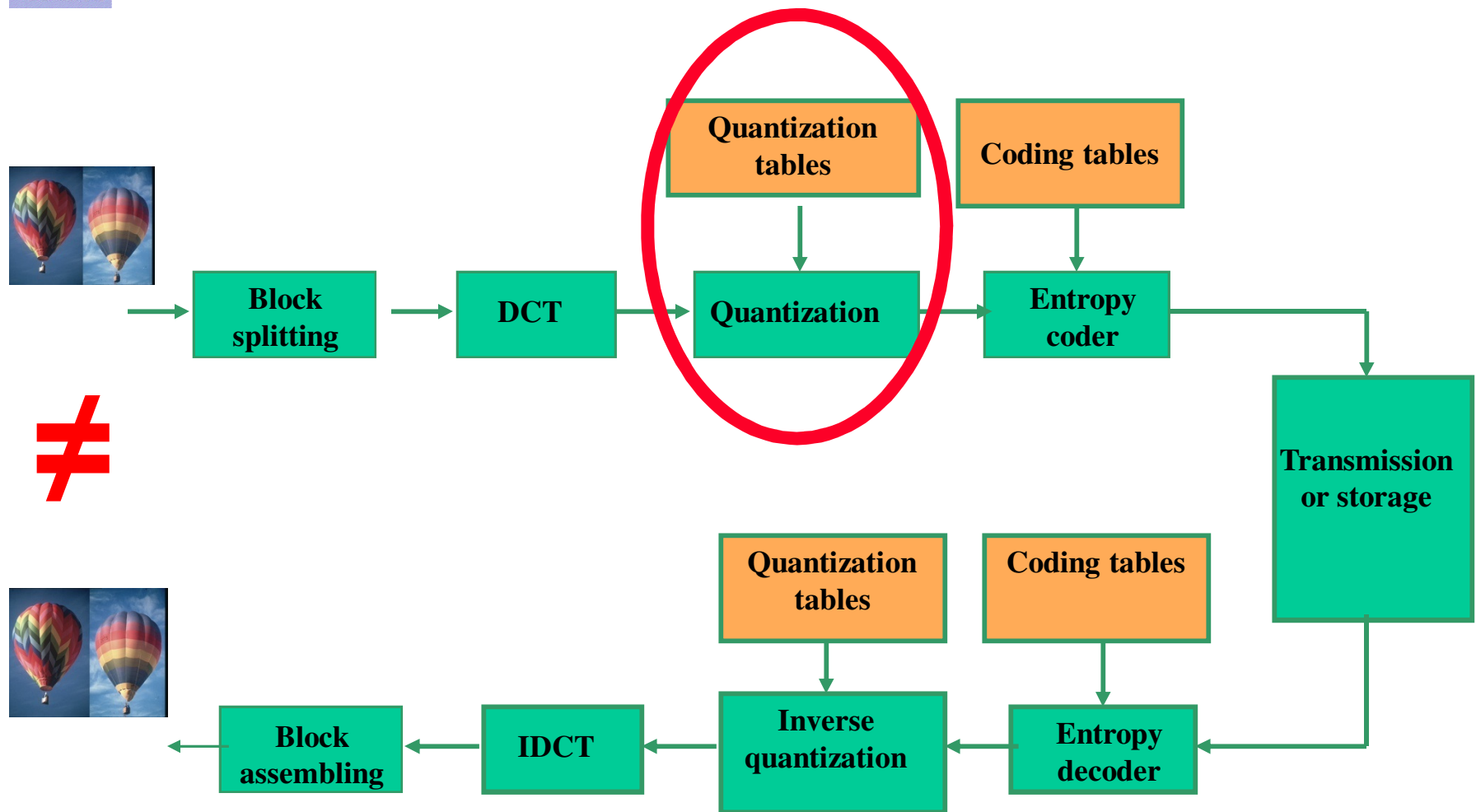


*Low frequency, high energy*

*High frequency, low energy*



# DCT Based Image Coding





# Quantization

**Quantization is the process by which irrelevancy or perceptual redundancy is reduced. This process is the major responsible for the quality losses in DCT based codecs (which may be transparent ;-).**

**Each quantization step may be selected taking into account the ‘minimum perceptual difference’ for the coefficient in question.**

**The quantization matrixes are not standardized but there is a default solution for ITU-R 601 resolution images (which still has to be signalled).**



# Quantization Matrices

JPEG suggests to quantize the DCT coefficients using the values for the ‘minimum perceptual difference’ for each coefficient or a multiple of them (for more compression); anyway, the quantization matrixes have to be always transmitted or signalled.

16	11	10	16	24	40	51	61
12	12	14	19	26	58	69	85
14	13	16	24	48	69	85	96
14	17	22	29	57	80	103	121
17	22	33	40	68	109	103	77
20	26	33	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

**Luminance**

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

**Chrominances**

**Situation:** Luminance and chrominance with 2:1 horizontal subsampling; samples with 8 bits (*Lohscheller*)

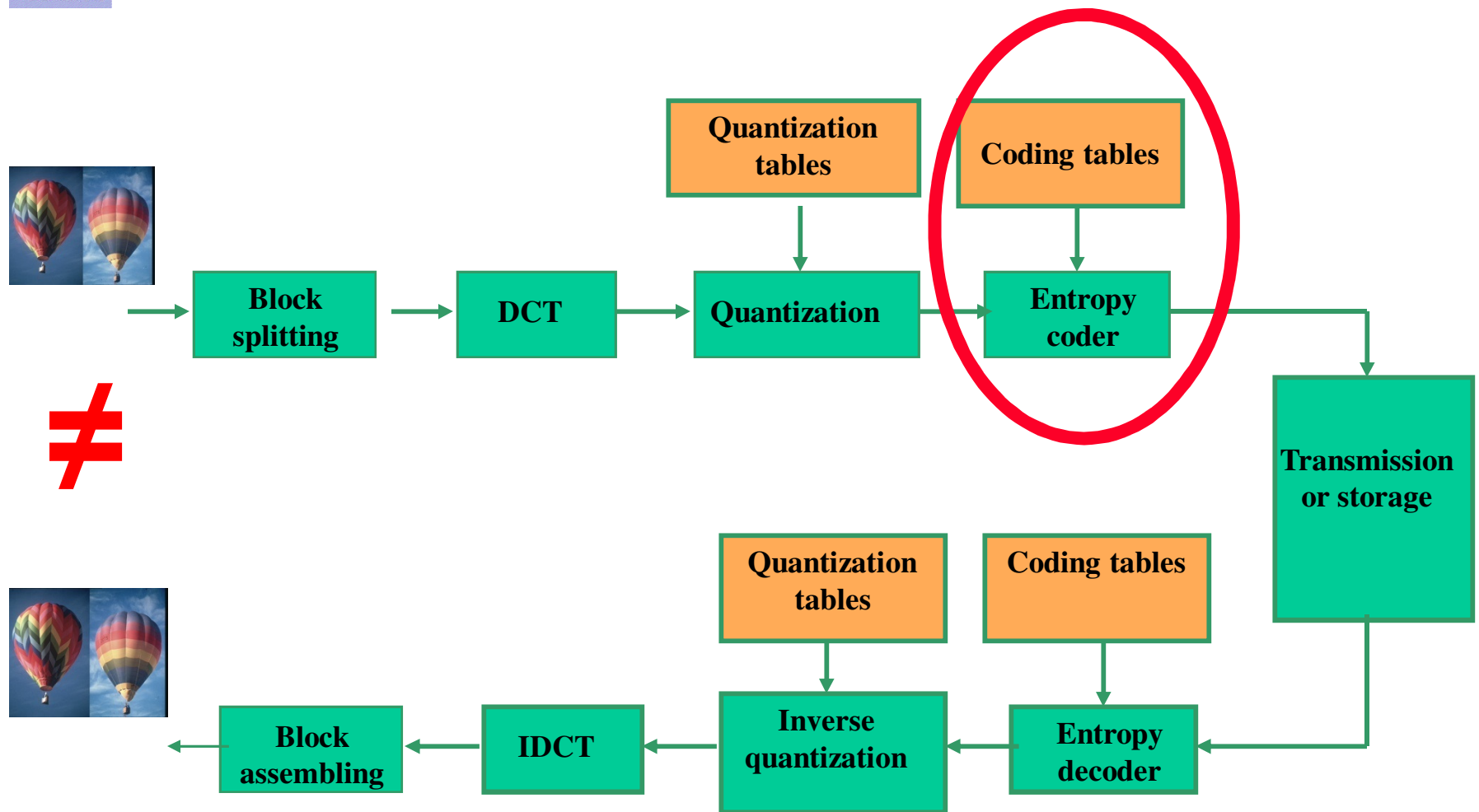
**Note:** Using as quantization steps these values divided by 2 guarantees decoded images with transparent quality.

$$\begin{bmatrix} 898.0000 & -149.5418 & 26.6464 & -14.0897 & 0.7500 & -5.7540 & 3.5750 & 0.0330 \\ 12.1982 & -16.5235 & -7.6122 & 5.2187 & -0.2867 & -1.9909 & 8.4265 & 1.2591 \\ 5.3355 & -2.6557 & 2.3410 & -9.9277 & 2.4614 & 4.4558 & -3.1945 & -3.1640 \\ 1.9463 & -2.7271 & 1.5106 & 2.8421 & -2.1336 & -2.7203 & -2.7510 & 5.4051 \\ 0.7500 & -2.0745 & 0.8610 & 0.2085 & 2.5000 & 1.8446 & 2.0787 & 2.4750 \\ 7.9536 & -2.6624 & 2.6308 & 0.4010 & 0.4772 & 3.3000 & 1.7394 & 0.3942 \\ -4.1042 & -0.1650 & -0.6945 & 0.0601 & 0.0628 & -0.7874 & -0.8410 & 0.3496 \\ -3.4688 & 2.3804 & 0.1559 & 0.8696 & 0.1142 & -0.5240 & -3.9974 & -5.6187 \end{bmatrix}$$

Quantizing ...

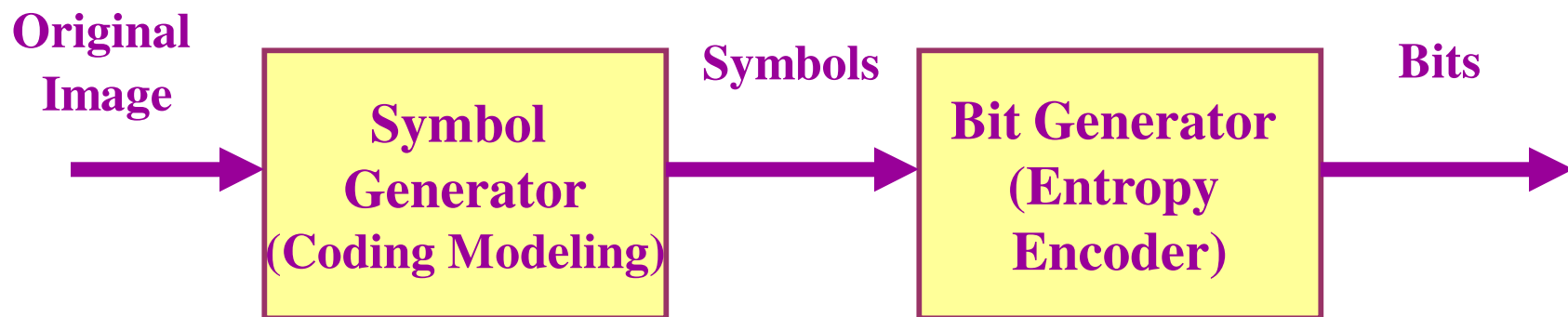
$$\begin{bmatrix} 56 & -14 & 3 & -1 & 0 & 0 & 0 & 0 \\ 1 & -1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

# DCT Based Image Coding





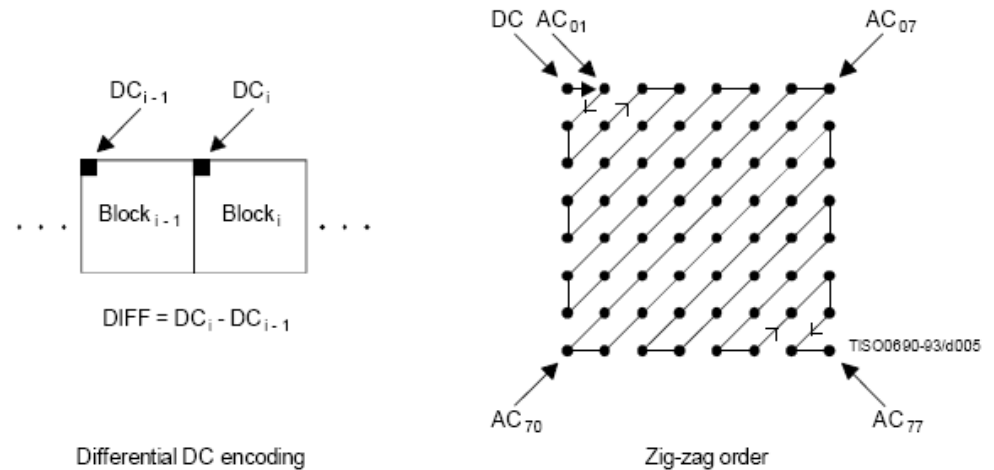
# JPEG Symbolic Model



**JPEG Model:** An image is represented as a sequence of (almost) independent  $8 \times 8$  samples blocks with each block represented by means of a zig-zag sequence of quantized DCT coefficients using *(run, level)* pairs, terminated by a *End of Block*.

# Generating the Symbols

The first step is to decide which symbols, this means (run,length) pairs, represent each  $8 \times 8$  block; these symbols will be entropy encoded.



- ★ The DC coefficient is treated differently (using differential prediction) because of the high correlation between the DC coefficients of adjacent  $8 \times 8$  blocks.
- ★ The remaining coefficients, after quantization, are zig-zag ordered in to facilitate entropy coding, coding the lower frequency coefficients before the higher frequency coefficients.

The precise definition of the symbols to encode depends on the DCT operation mode and the type of entropy coding.



# Entropy Coding

**Entropy coding uses the statistics of the symbols to code to reach (lossless) additional compression.**

**For JPEG Baseline, entropy coding includes two phases:**

- ★ (RUN, LEVEL) PAIRS TO SYMBOLS - Conversion of the sequence of (run, level) pairs associated to the DCT coefficients zig-zag ordered into an intermediary sequence of symbols (symbols 1 and 2 in the following)**
- ★ SYMBOLS TO BITS - Conversion of the sequence of intermediary symbols into a sequence of bits without externally identifiable boundaries**



# Entropy Coding: Intermediary Symbols

Each non-null AC coefficient is represented combining its quantization level (*amplitude*) with the number of null DCT coefficients preceding it in the zig-zag scanning (*position*) using a *run* in 0...62.

Each (*run, level*) pair associated to a non-null AC coefficient is represented by a pair of symbols:



Symbol 1 - Huffman (bidimensional)

Symbol 2 - VLI

- ★ *Run* - number of null DCT coefficients preceding the coefficient being coded in the zig-zag scanning
- ★ *Size* – number of bits used to code the *Level* (this means symbol 2)
- ★ *Level* - amplitude of the AC coefficient to be coded

Each DC coefficient is represented in the same way, with the *run* equal to zero.



# Entropy Coding: Generating the Bits



Symbol 1 - Huffman (bidimensional)    Symbol 2 - VLI

- ★ Symbol 1 for the DC and AC coefficients is coded with the Huffman table corresponding to the component in question.
- ★ Symbol 2 is coded with a *Variable Length Integer* (VLI) code which length depends on the level being coded.
- ★ VLI codes are VLC codes where the codeword length is previously indicated; they are based on a complement to 2 notation.
- ★ VLI codes may be computed instead of stored (important for big codes) and are not significantly less efficient than Huffman codes.



# Coding Tables (Symbols 1 and 2)

	0	1	2	Size	9	10
<b>Runlength</b>	0	EOB	<b>Run-size values</b>			
	.	X				
	.	X				
	.	X				
	15	ZRL				

**Bidimensional**  
*(run, size)*  
**coding**

Size	Amplitude
1	-1, 1
2	-3, -2, 2, 3
3	-7 ... -4, 4 ... 7
4	-15 ... -8, 8 ... 15
5	-31 ... -16, 16 ... 31
6	-63 ... -32, 32 ... 63
7	-127 ... -64, 64 ... 127
8	-255 ... -128, 128 ... 255
9	-511 ... -256, 256 ... 511
10	-1023 ... -512, 512 ... 1023

**Amplitude (*level*) coding**  
**VLI**

# VLI Coding Example: +12 and -12

<i>Run</i>	<i>Size</i>	<i>Level</i>
------------	-------------	--------------

0000 -15

0001 -14

0010 -13

0011 -12

0100 -11

0101 -10

0110 -9

0001 -8

1000 8

1001 9

1010 10

1011 11

1100 12

1101 13

1110 14

1111 15

Symbol 1 - Huffman (bidimensional)    Symbol 2 - VLI

1100

**+12 in binary  
after 'inverting' all bits**

The code for negative values is simply the 'inversion' of the code for positive values.

1100

**+12 em binário**



# Summary: How Does JPEG Compress ?

## ★ Spatial Redundancy - DCT

- **Image samples statistically dependent are converted into uncorrelated DCT coefficients with the signal energy concentrated in the smallest possible number of coefficients**

## ★ Irrelevancy

- **DCT coefficients are quantized using psychovisual criteria**

## ★ Statistical Redundancy

- **The statistic of the symbols is exploited using *run-length* coding and Huffman entropy coding (or arithmetic coding).**



# JPEG Extensions



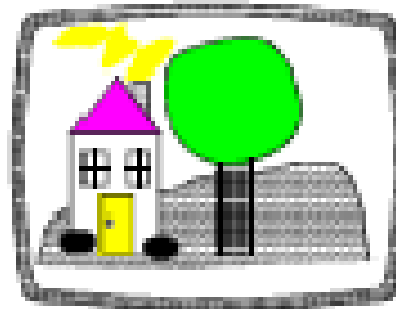
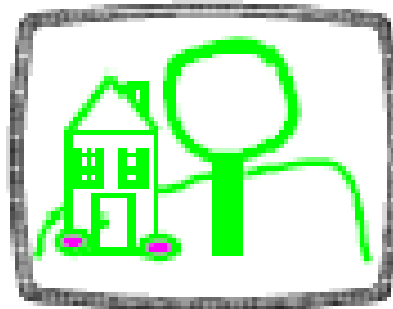
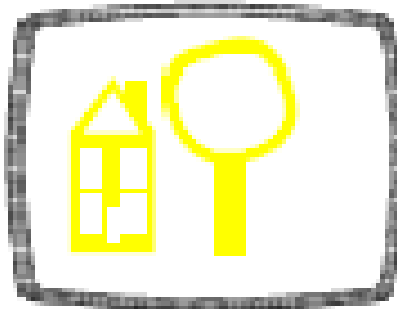
# JPEG Operation Modes

The various operation modes result from the need to provide a solution to a large range of applications with different requirements.

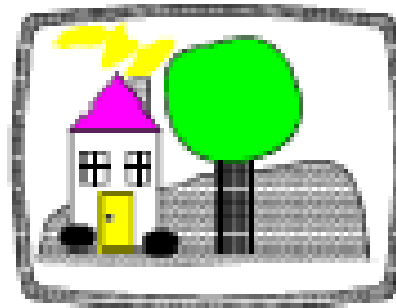
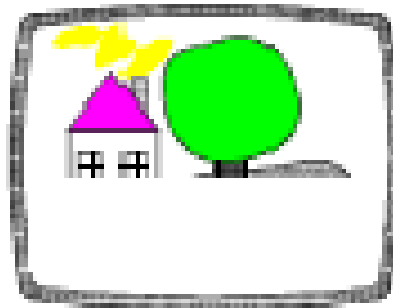
- ★ **SEQUENTIAL MODE** – Each image component is coded in a single scan (from top to bottom and left to right).
- ★ **PROGRESSIVE MODE** - The image is coded with several scans which offer a successively better quality.
- ★ **HIERARCHICAL MODE** - The image is coded in several resolutions exploiting mutual dependencies, with lower resolution images available without decoding higher resolution images.
- ★ **LOSSLESS MODE** – This mode guarantees the exact reconstruction of each sample in the original image.

For each operation mode, one or more codecs are specified; these codecs are different in terms of the sample precision (bit/sample) or the entropy coding method.

# Progressive versus Sequential Modes



Progressive



Sequential

TÉC00733-63/0909



# JPEG Progressive Mode

The image is coded with successive scans. The first scan gives very quickly an idea about the image content; after, the quality of the decoded image is progressively improved with the successive scans (layers).

The implementation of the progressive mode requires a memory with the size of the image able to store the quantized DCT coefficients (11 bits for the baseline process) which will be partially coded with each scan.

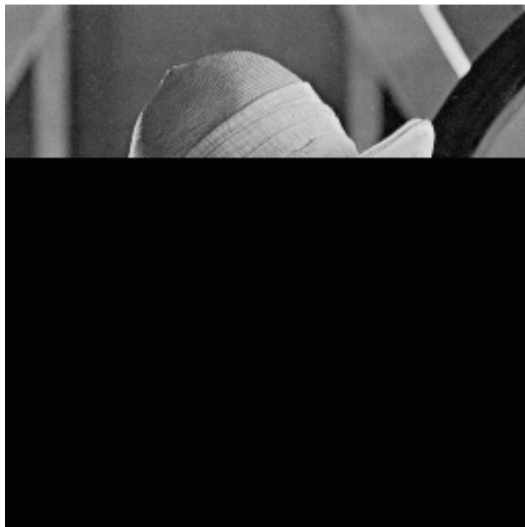
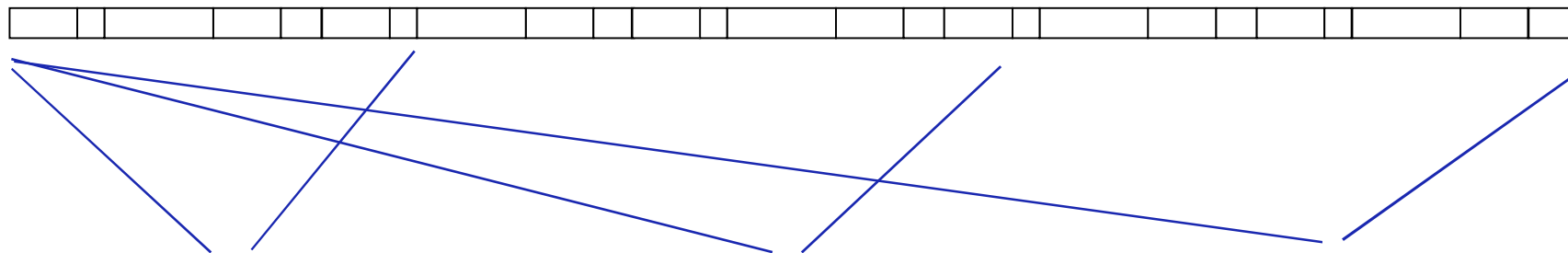
There are methods of implementing the progressive mode:

- ★ **SPECTRAL SELECTION** – Only a specified 'zone' of DCT coefficients is coded in each scan (typically goes from low to high frequencies)
- ★ **GROWING PRECISION** – DCT coefficients are coded with successively higher precision

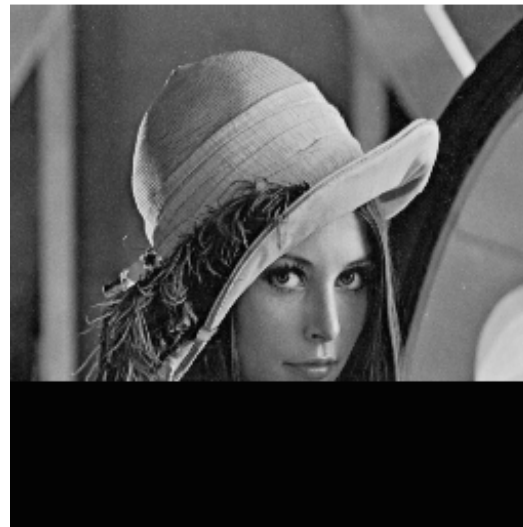
The spectral selection and successive approximations methods may be applied separately or together.

# Sequential Mode or No Scalability ...

NON scalable stream



Decoding 1



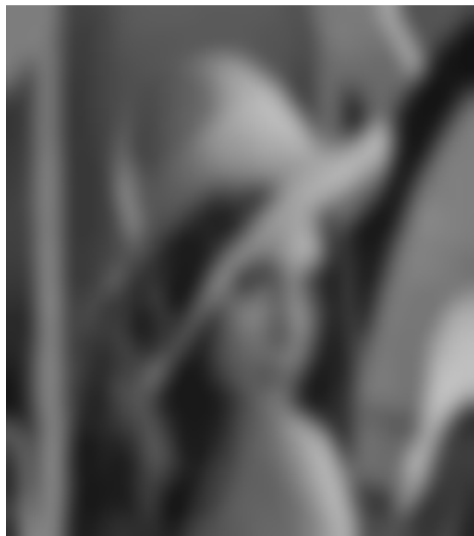
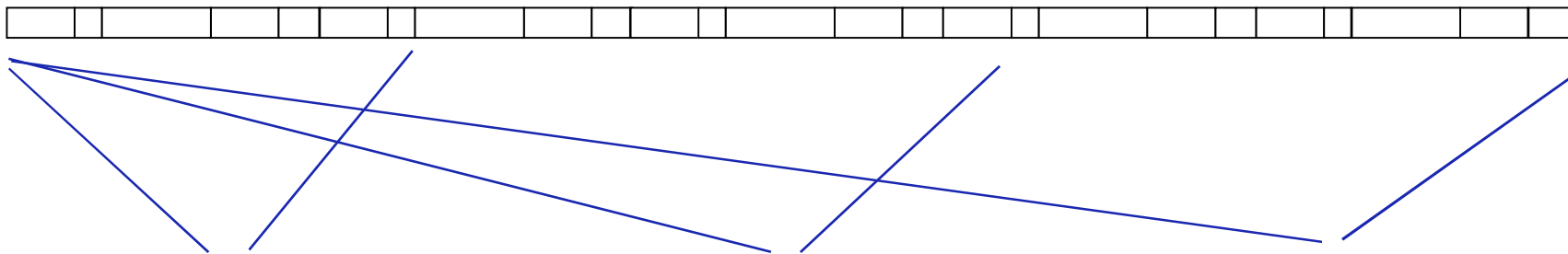
Decoding 2



Decoding 3

# Progressively More Quality: Quality or SNR Scalability

Scalable stream



Decoding 1



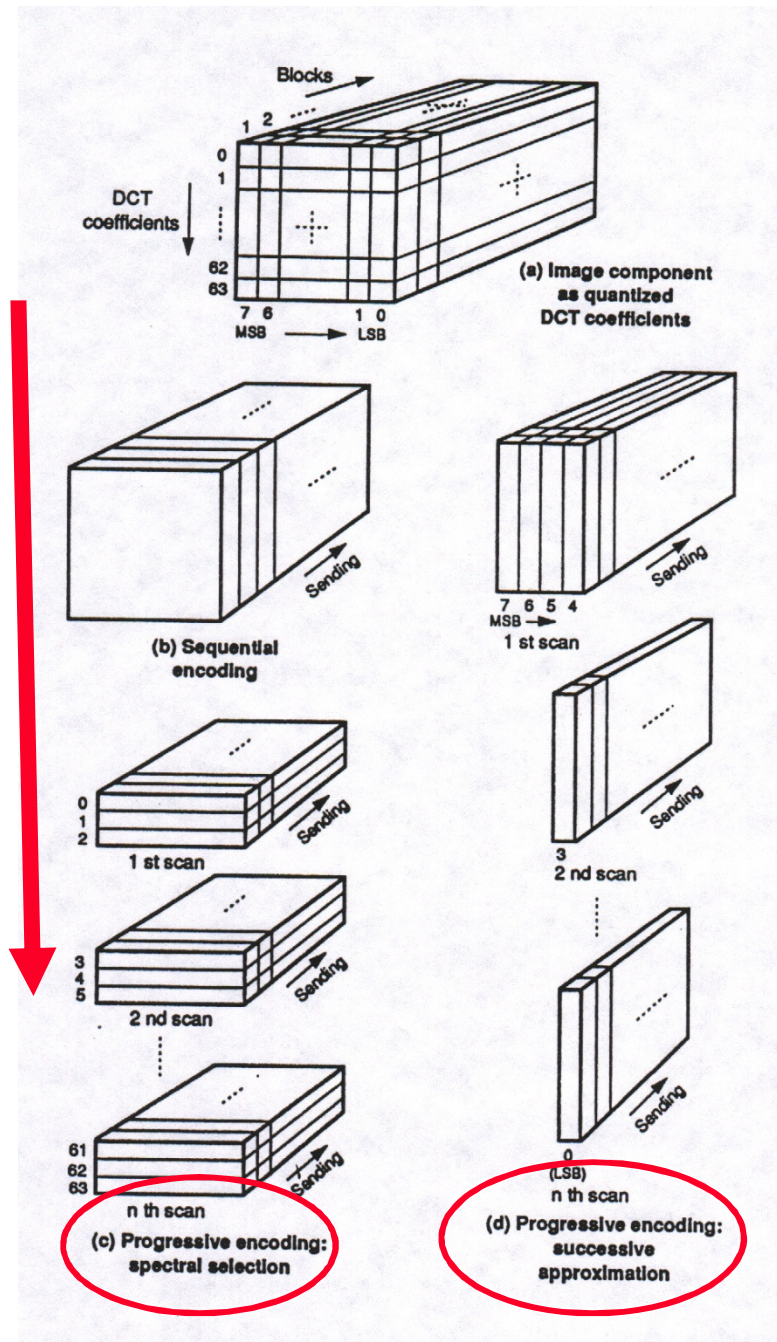
Decoding 2



Decoding 3

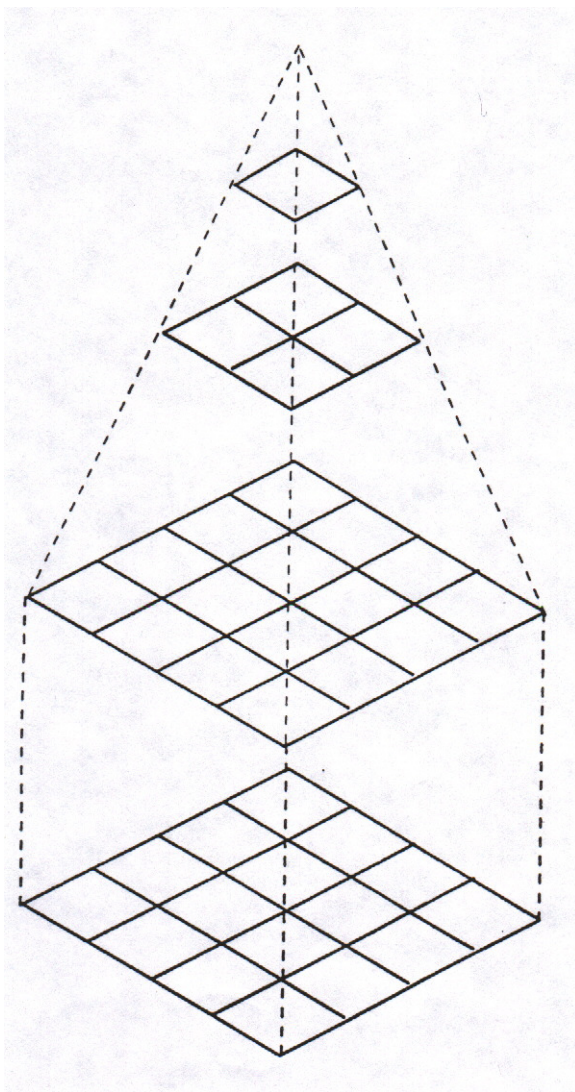
# Progressive Modes: Spectral Selection and Growing Precision

Increasing  
number of  
coefficients

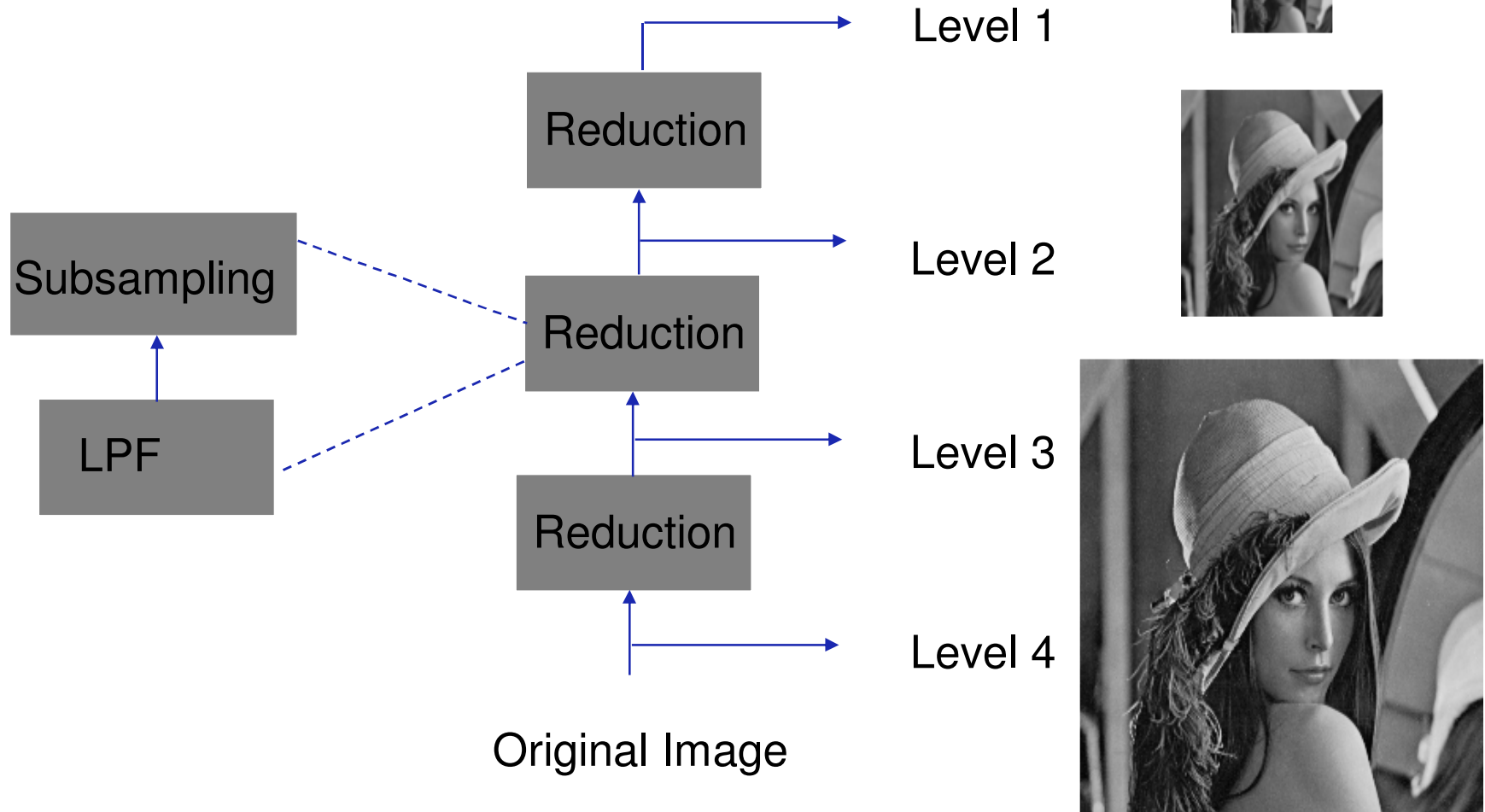


Increasing  
precision for  
each coefficient

# Hierarchical Mode

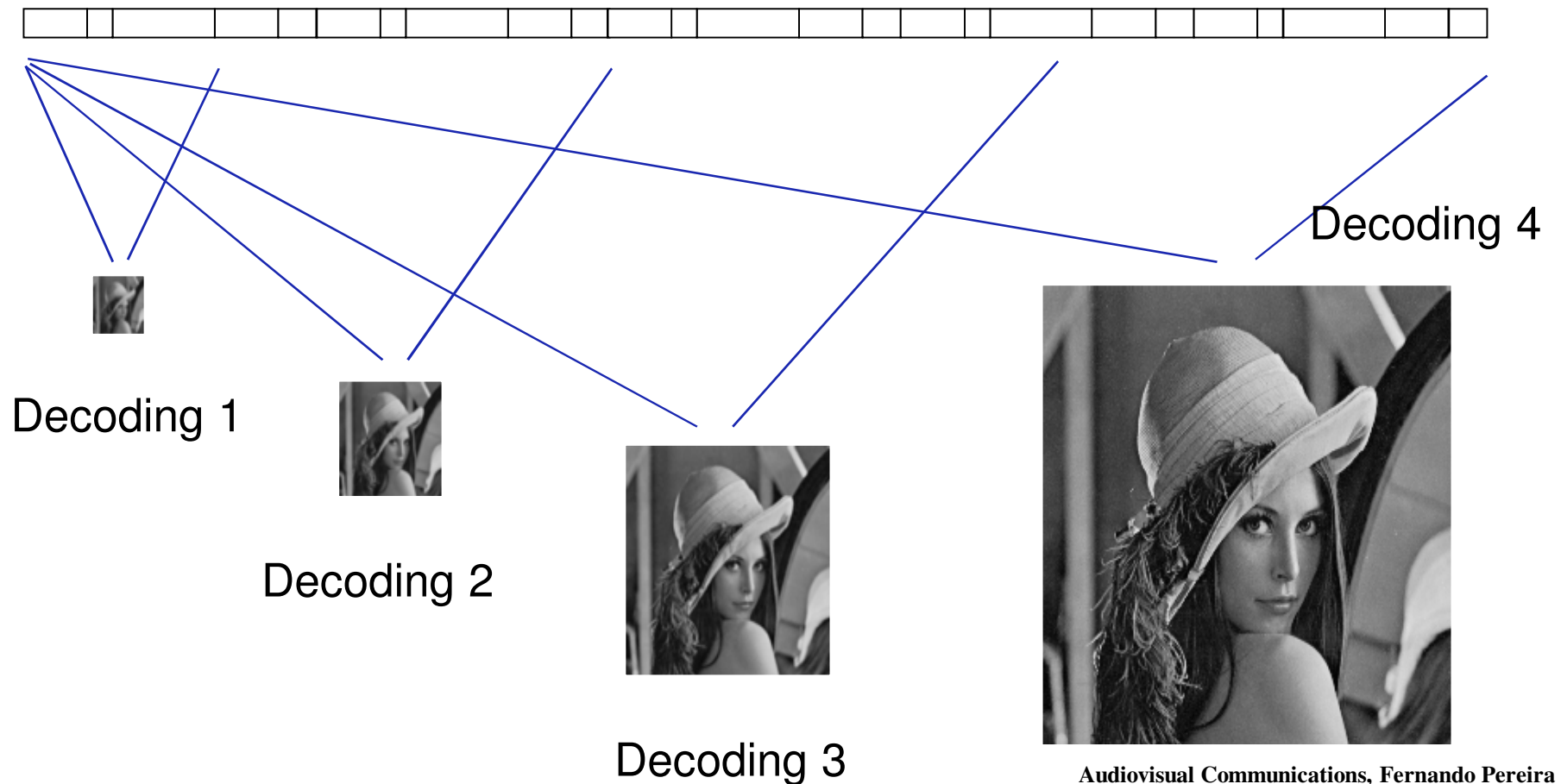


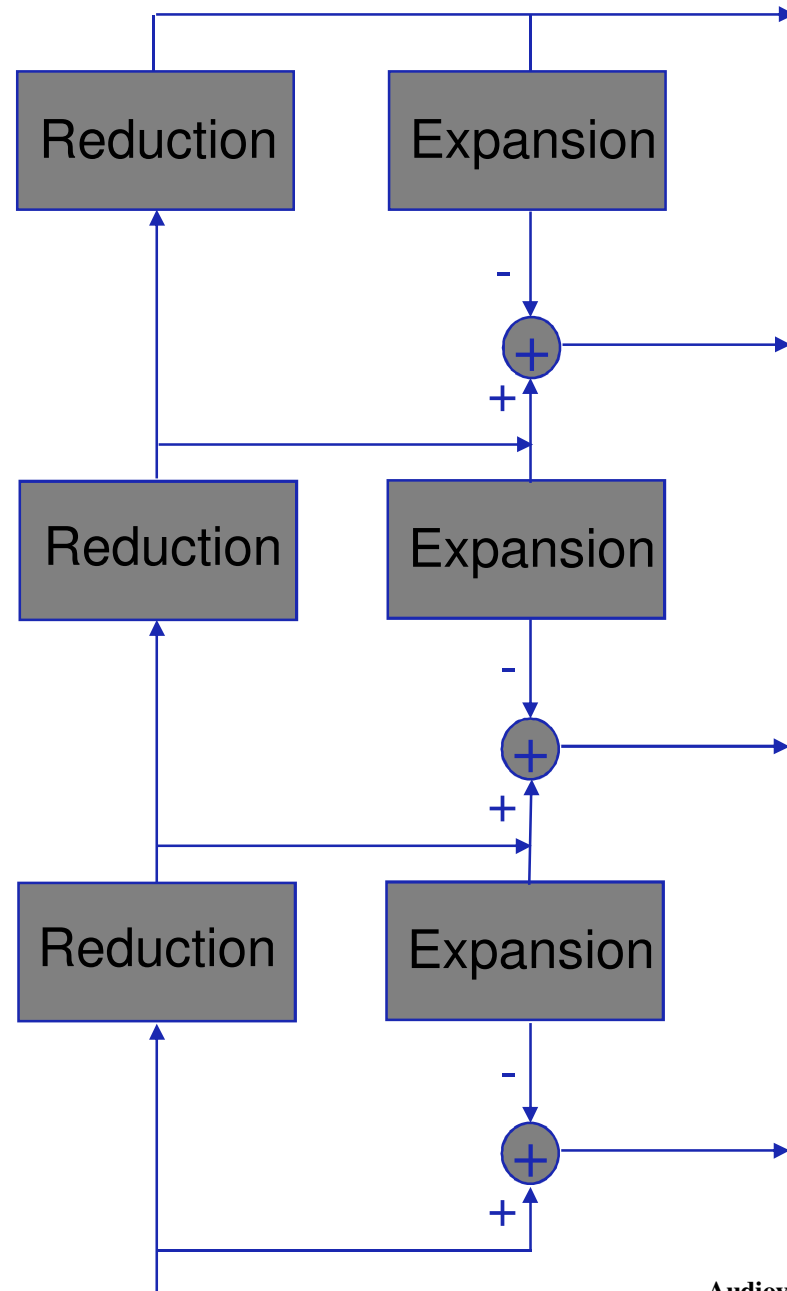
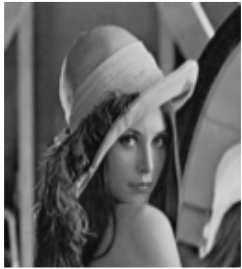
- ★ **The hierarchical mode implements a pyramidal coding of the image with several resolutions. Each (higher) resolution multiplies by 2 the number of vertical and horizontal samples.**
- ★ **JPEG hierarchical coding may integrate in the various layers, lossless coding as well as DCT based coding.**



# Hierarchical Mode or Spatial Scalability ...

Scalable stream







# JPEG Lossless Mode

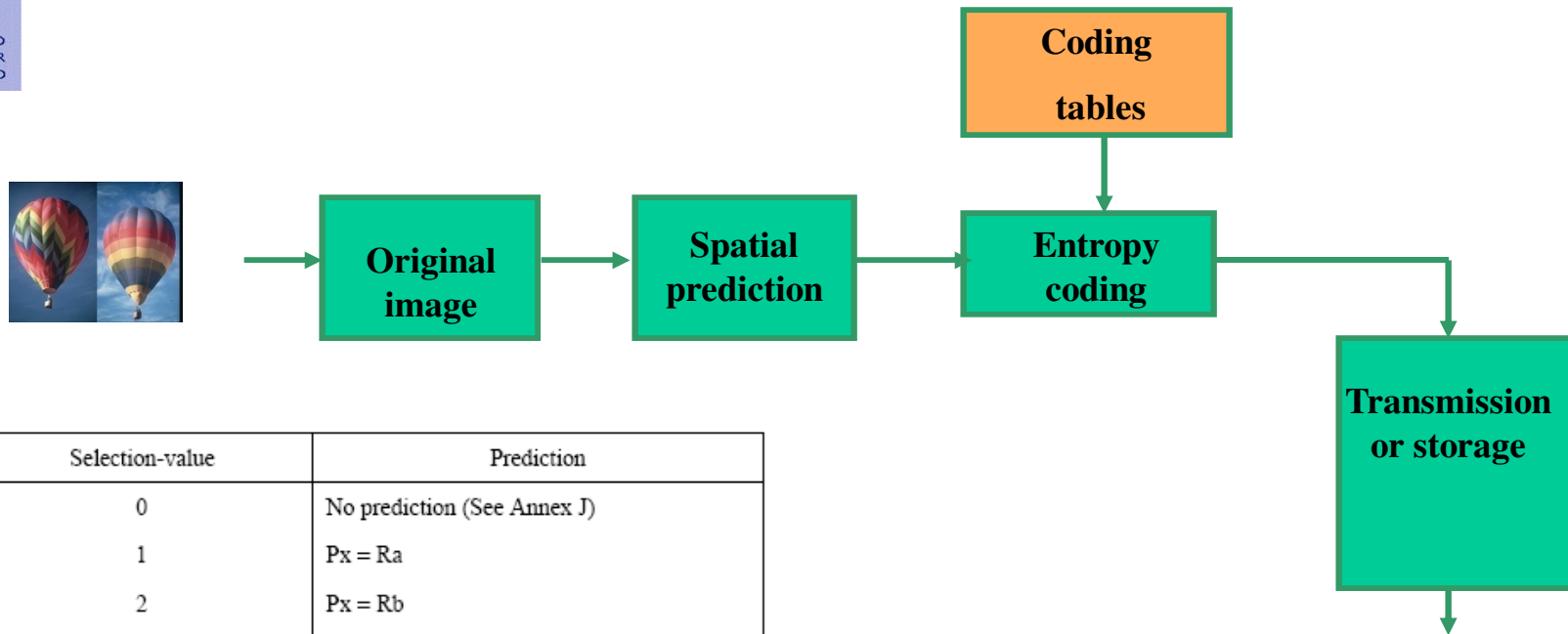
**The JPEG lossless mode is based on a spatial predictive scheme. The prediction combines the values of, at most, 3 adjacent pixels. Finally, the prediction mode and the prediction error are coded.**

**The definition of a DCT based lossless mode would require a much more precise definition of the codecs.**

**Two codecs are specified for the lossless mode: one using Huffman coding and another using arithmetic coding.**

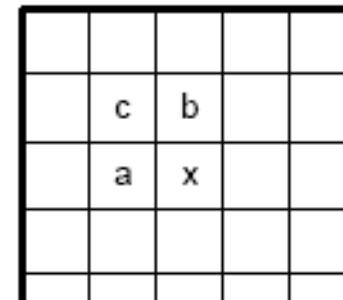
- ★ The codecs may use any precision between 2 and 16 bit/sample.**
- ★ The JPEG lossless mode offers  $\approx 2:1$  compression for colour images of medium complexity.**

# Lossless Coding



Selection-value	Prediction
0	No prediction (See Annex J)
1	$P_x = R_a$
2	$P_x = R_b$
3	$P_x = R_c$
4	$P_x = R_a + R_b - R_c$
5	$P_x = R_a + ((R_b - R_c)/2)^{a)}$
6	$P_x = R_b + ((R_a - R_c)/2)^{a)}$
7	$P_x = (R_a + R_b)/2$

a) Shift right arithmetic operation



**$P_x$  is the prediction and  $R_a$ ,  $R_b$ , and  $R_c$  are the reconstructed samples immediately to the left, above, and diagonally to the left of the current sample.**

**$x$  is the sample to code**



# Compression versus Quality



**JPEG offers the following levels of compression/quality for sequential DCT based coding, considering colour images with medium complexity:**

- ★ **0.25 - 0.5 bit/pixel – medium to good quality; enough for some applications**
- ★ **0.5 - 0.75 bit/pixel – good to very good quality; enough for many applications**
- ★ **0.75 - 1.5 bit/pixel – excellent quality; enough for most applications**
- ★ **1.5 - 2.0 bit/pixel – transparent quality; enough for the most demanding applications**

**These compression/quality levels are only indicative since the compression always depends on the specific image content, notably if there is more or less spatial redundancy.**

**The quality level may be controlled through the quantization steps.**

# JPEG Test Images



**Barb 1**



**Barb 2**

# JPEG Test Images



**Board**



**Boats**



# JPEG Test Images



**Hill**



**Hotel**

# JPEG Test Images



**Zelda**



**Toys**



# Performance Experiment

## Conditions:

- ★ Baseline coding process (DCT based), using the quantization tables suggested in the JPEG standard and Huffman/VLI coding with optimized tables and ITU-T 601 spatial resolution.
- ★ A JPEG with optimized tables is simply a JPEG stream including custom Huffman tables created after the statistical analysis of the image's unique content.

## Conclusions:

- ★ Most of the signal energy is concentrated on the luminance component.
- ★ Most of the bits are used for AC DCT coefficients.
- ★ *Barb1* and *Barb2* test images, which are richer in high frequencies, lead to lower compression factors, although still within the JPEG compression/quality targets.



# Performance Results

Imagem	Coef. DC Lum (byte)	Coef DC crom (byte)	Coef AC Lum (byte)	Coef AC Crom (byte)	Global (byte)	Factor Comp.	Ritmo (bit/pel)	SNR Y (dB)	SNR U (dB)	SNR V (dB)
Zelda	4208	2722	19394	3293	29617	28.00	0.571	38.09	42.01	40.98
Barb1	4520	2926	40995	4878	53319	15.56	1.028	33.39	38.38	39.01
Boats	3833	2255	29302	3755	39145	21.19	0.755	35.95	41.13	40.13
Black	3497	2581	21260	6015	33353	24.87	0.643	37.75	40.09	38.23
Barb2	4223	2933	41613	7246	56014	14.81	1.080	32.37	37.05	36.09
Hill	4007	2206	34890	3727	44830	18.50	0.865	34.31	39.83	38.09
Hotel	4239	2708	35520	6658	49125	16.88	0.945	34.55	37.95	36.99



# **JPEG Summary: Baseline Process**

**Mandatory for all JPEG codecs !**

**DCT Based**

**Original image: samples with 8 bits per component**

**Sequential mode**

**Huffman coding: 2 AC and 2 DC tables**

**Images with 1 to 4 components**

**Interleaving enabled**



# **JPEG Summary: DCT based Extension**

**DCT based**

**Original image: samples with 8 or 12 bits for each component**

**Sequential and Progressive modes**

**Huffman or arithmetic coding: 4 AC and 4 DC tables**

**Images with 1 to 4 components**

**Interleaving enabled**



# **JPEG Summary: Hierarchical Process**

**Hierarchical mode**

**Multiple frames (differential or not)**

**Using DCT based extension ou lossless coding**

**Images with 1 to 4 components**

**Interleaving enabled**



# **JPEG Summary: Lossless Coding Process**

**Spatial predictive coding (not DCT based)**

**Original image: samples with 2 to 16 bits per component**

**Sequential scanning (lossless)**

**Huffman coding: 4 tables**

**Images with 1 to 4 components**

**Interleaving enabled**



# The JPEG 2000 Standard



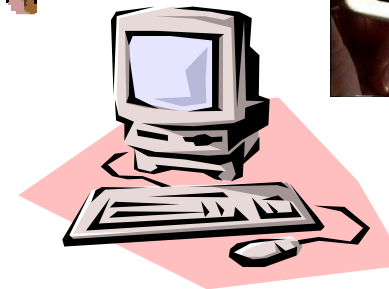
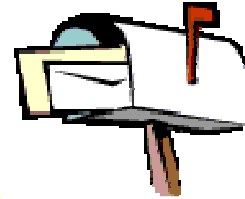
# Why Another Image Compression Standard?

To address areas where the current image compression standards fail to produce the best quality or performance, notably:

- ★ Low bitrate compression, for example below 0.25 bpp (bits per pixel)
- ★ Lossless and lossy compression: no current standard can provide superior lossy and lossless compression in a single bitstream
- ★ Computer generated imagery: JPEG was optimized for natural imagery and does not perform well on computer generated imagery
- ★ Transmission in noisy environments: JPEG has provisions for resynchronization but image quality suffers dramatically when bit errors happen
- ★ Compound documents: JPEG is seldom used in the compression of compound documents because of its poor performance when applied to bilevel (e.g. text) imagery
- ★ Random bitstream access and processing
- ★ Open architecture: desirable to allow optimizing the system for different image types and applications
- ★ Progressive transmission by pixel accuracy and resolution

# JPEG 2000 Target Applications

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



★ Digital Photography



★ Remote Sensing

★ Facsimile

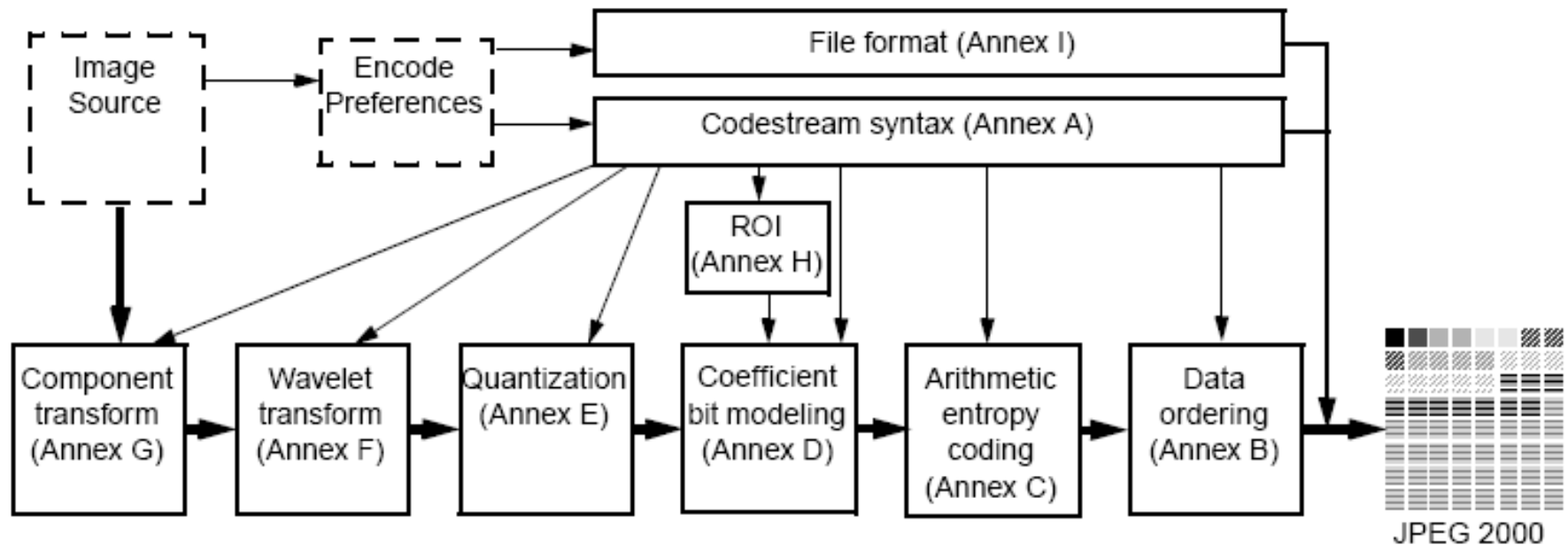


★ Medical

★ Digital Libraries

★ E-Commerce

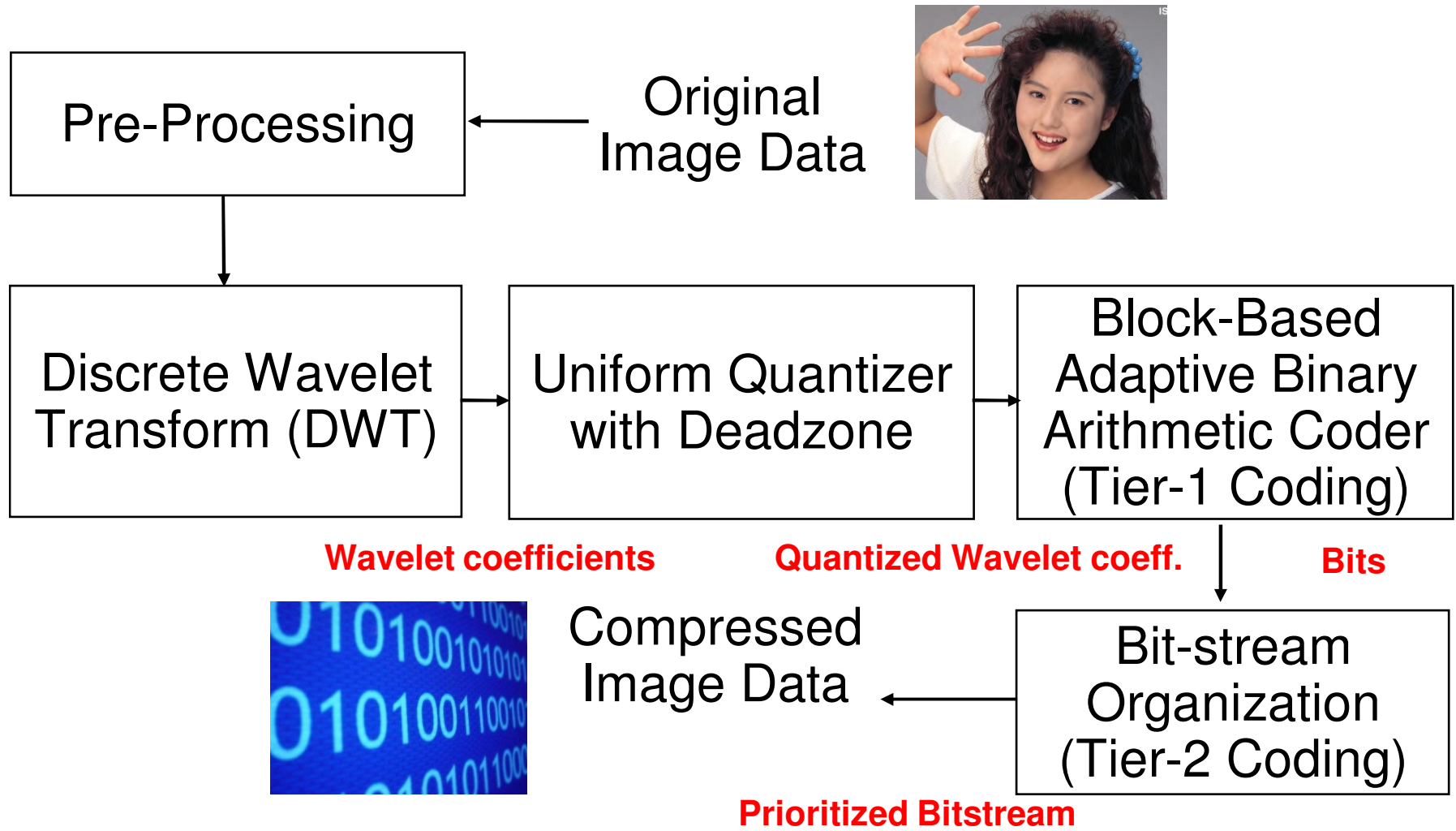
# JPEG 2000 Encoder Architecture



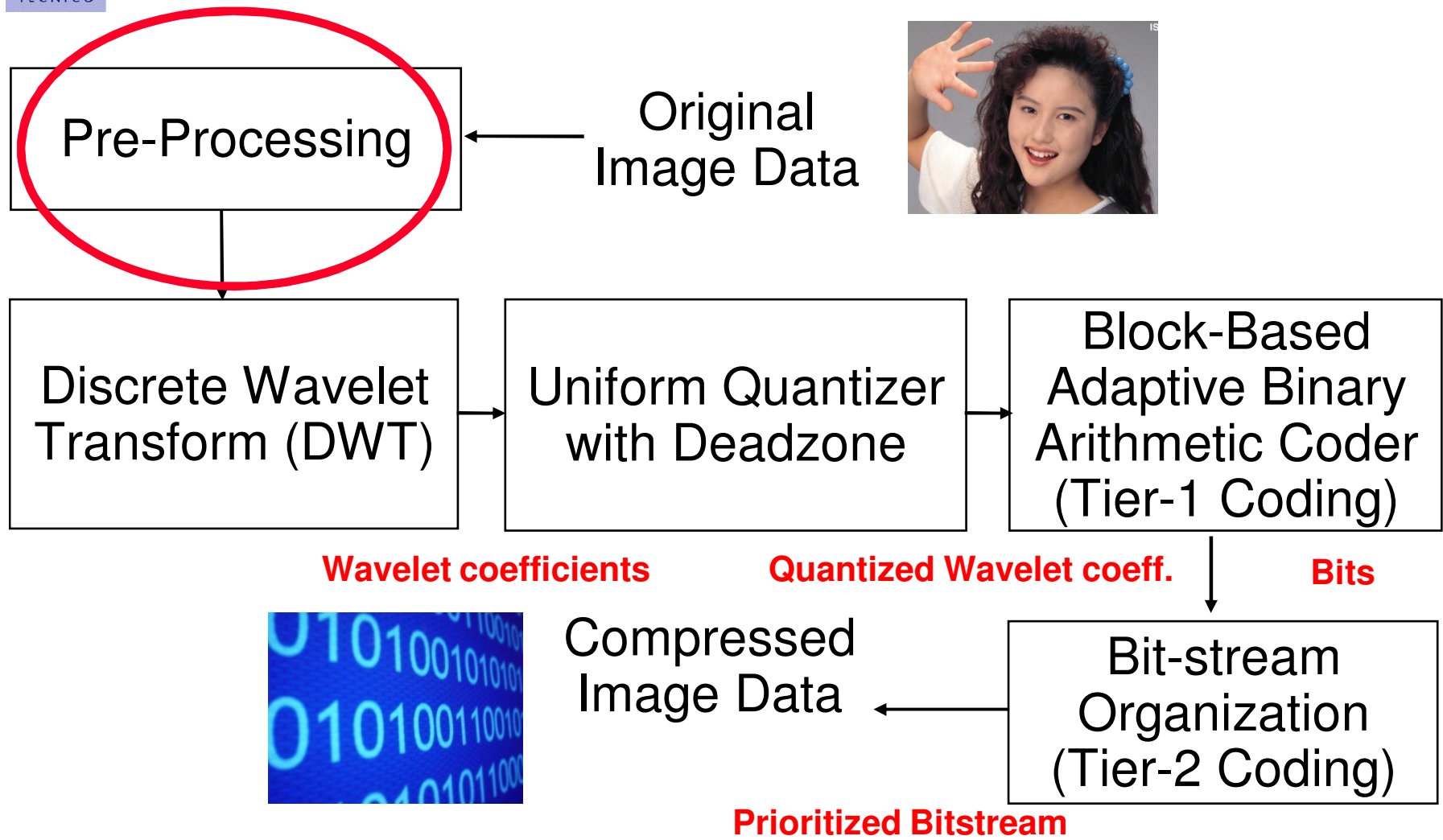
**JPEG 2000 encoder is applied to the full image or to a set of independent mosaics – tiles - providing spatial random access.**

*A mosaic is a rectangular part of the image; typically, the image is divided in all similar mosaics.*

# JPEG 2000 Main Encoder Modules



# JPEG 2000 Main Encoder Modules





# JPEG 2000 Pre-Processing

## ★ Tile partition:

- Each image may be coded as a whole or divided in tiles
- Each component of each tile is encoded independently, e.g. Y,  $C_r$ ,  $C_b$

## ★ DC level shifting:

- Unsigned sample values (0- $\rightarrow$ 255)  $\rightarrow$  Signed values (-127- $\rightarrow$ 128)
- To have zero-average signals

## ★ Colour transformation:

- To decorrelate the colour data
  - RGB  $\rightarrow$  YCbCr (ICT)
  - RGB  $\rightarrow$  YUV (RCT)



# Irreversible Colour Transform (ICT)

- ★ The ICT is the same as the *conventional* YCbCr transform for the representation of image and video signals.
- ★ A colour transformation is applied to achieve higher compression efficiency.

$$Y = 0.299 (R - G) + G + 0.114 (B - G)$$

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

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

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



# Reversible Color Transform (RCT)

- ★ The ICT is not capable of lossless coding !
- ★ The reversible color transform (RCT) is an integer-to-integer approximation intended for lossless coding.

## Forward RCT:

$$Y = \left\lfloor \frac{1}{4}(R + 2G + B) \right\rfloor$$

$$C_b = B - G$$

$$C_r = R - G$$

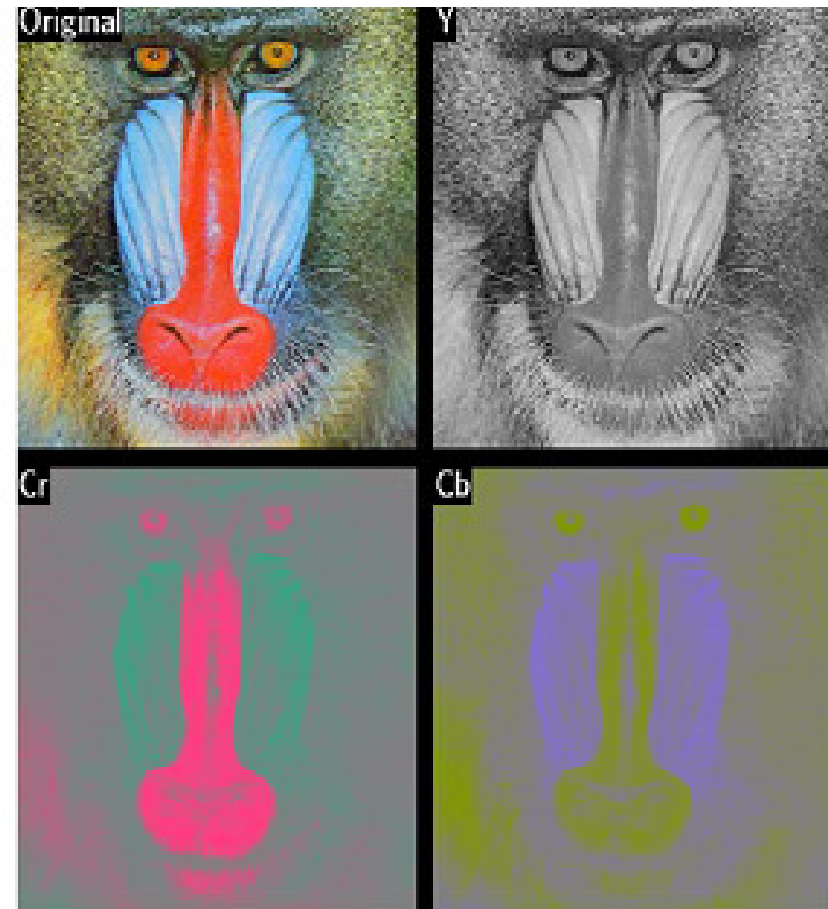
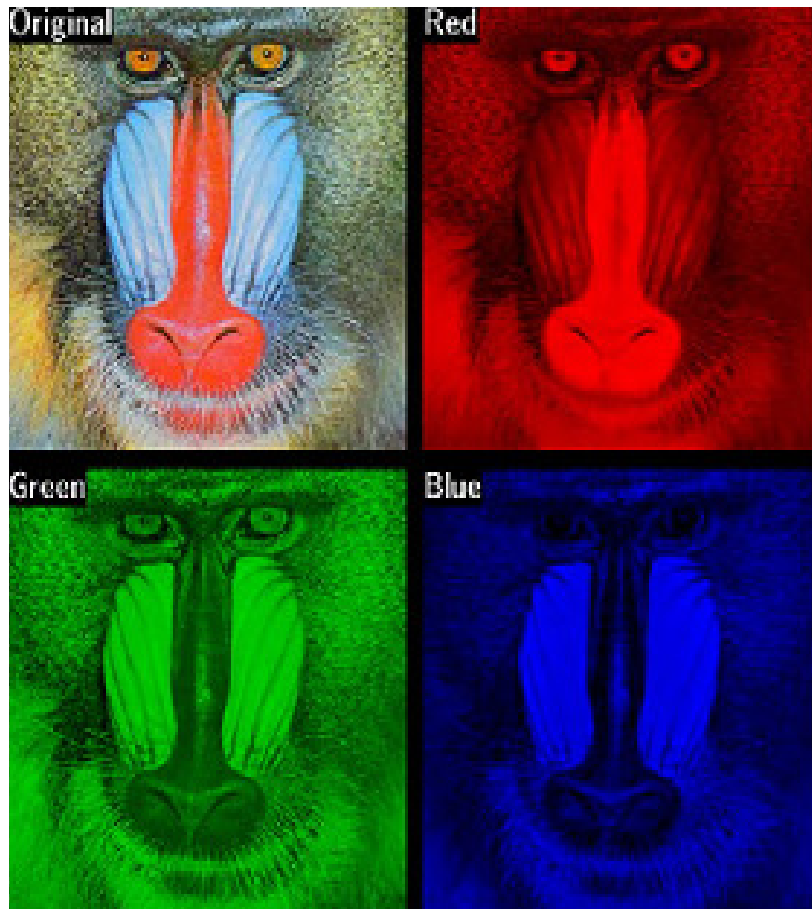
## Inverse RCT:

$$G = Y - \left\lfloor \frac{1}{4}(C_b + C_r) \right\rfloor$$

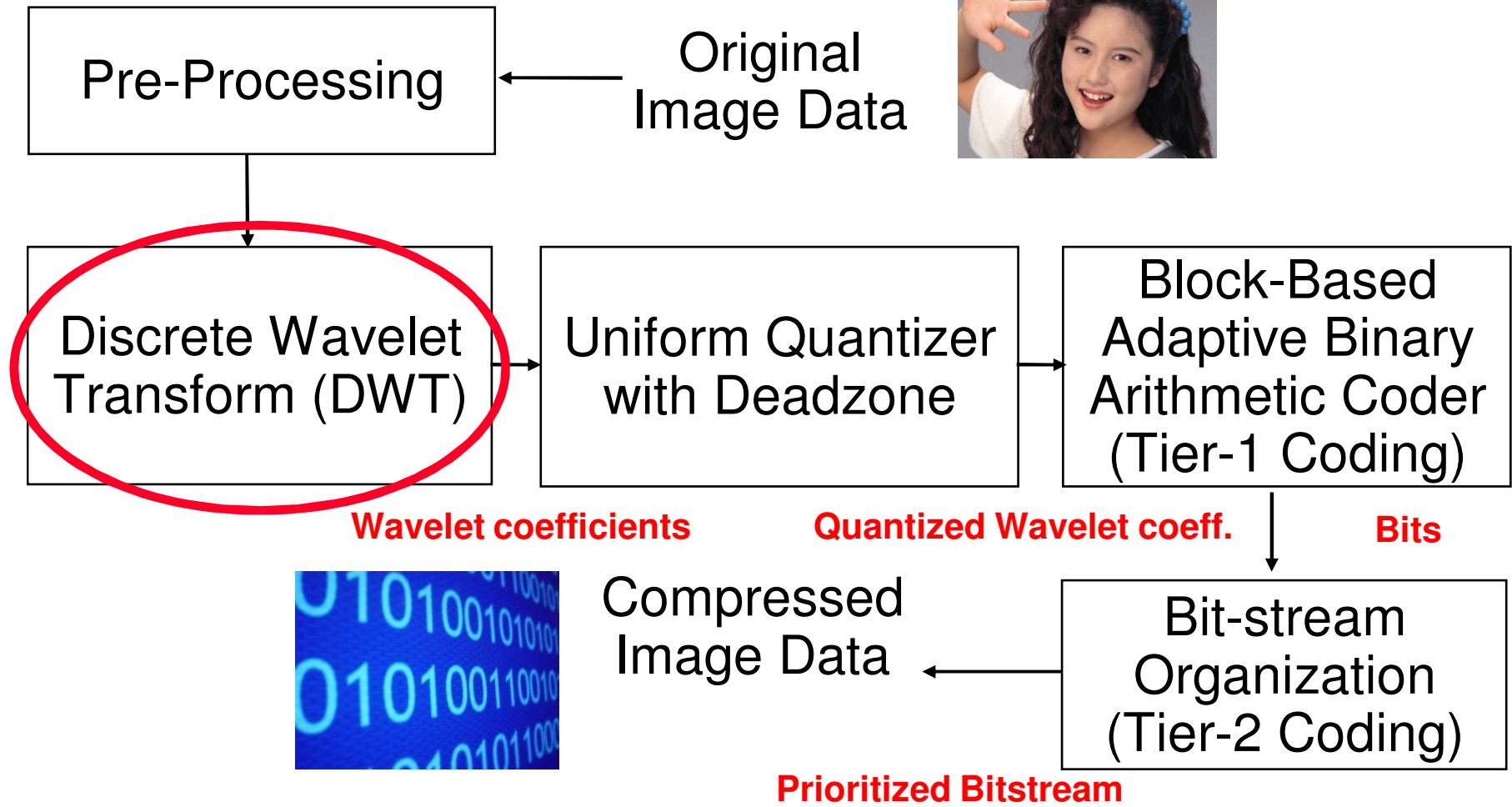
$$R = C_r + G$$

$$B = C_b + G$$

# Colour Transformation Example



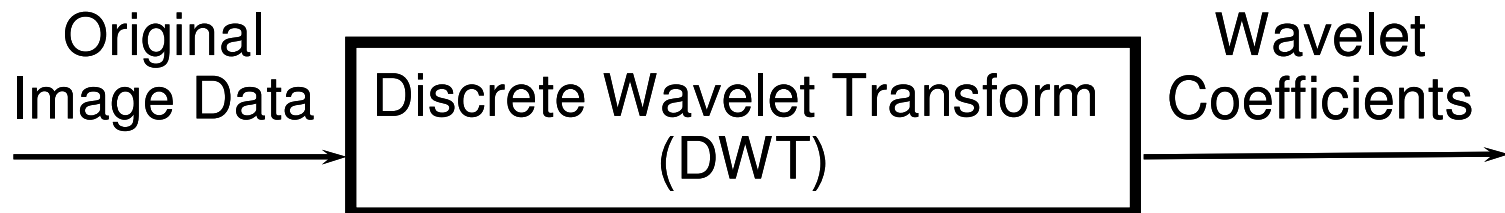
# JPEG 2000 Main Encoder Modules



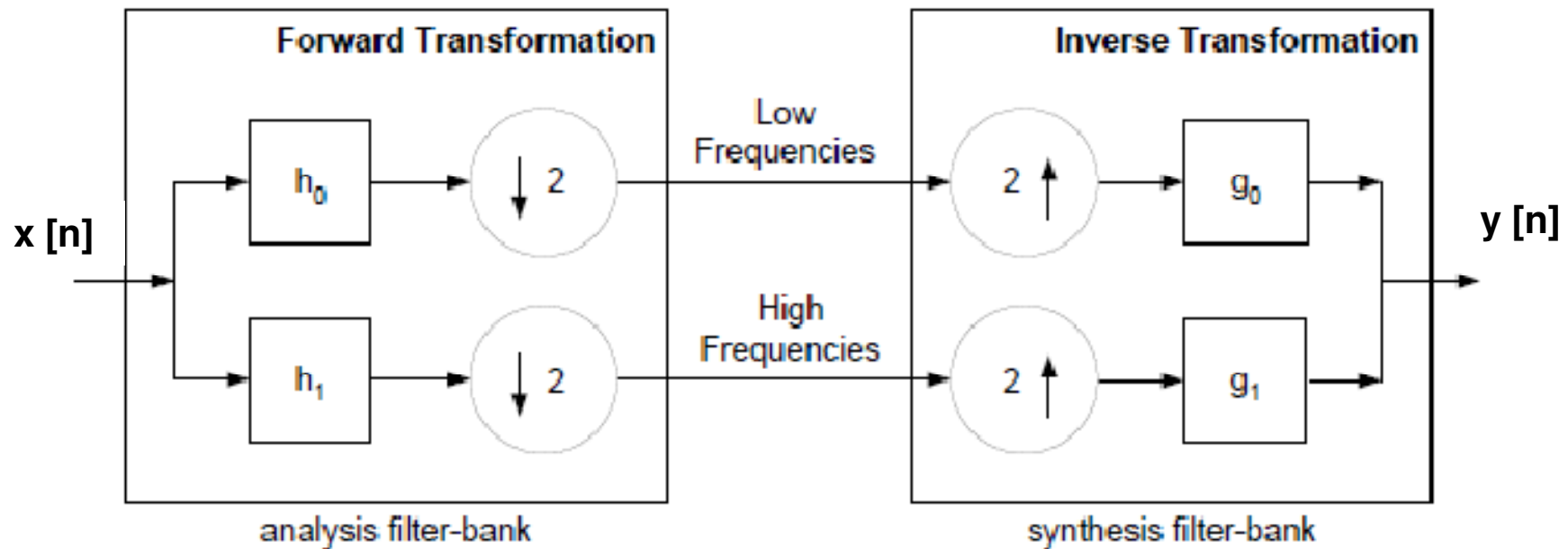


# JPEG 2000: The Wavelet Transform

- ★ **Multi-resolution image representation is inherent to the Discrete Wavelet Transform (DWT).**
- ★ **The full frame/tile nature of the transform decorrelates the image across a larger scale and eliminates blocking artifacts at high compression.**
- ★ **The use of integer DWT filters allows for both lossless and lossy compression within a single compressed JPEG 2000 bitstream.**
- ★ **DWT provides a frequency band decomposition of the image where each subband can be quantized according to its visual importance.**
- ★ **Two DWT filters are specified in JPEG 2000 Part I: irreversible Daubechies (9,7) and reversible (5,3); JPEG 2000 Part II allows using arbitrary filters.**



# 1D Bi-Orthogonal DWT: Filtering + Subsampling



Bi-orthogonal filter bank:

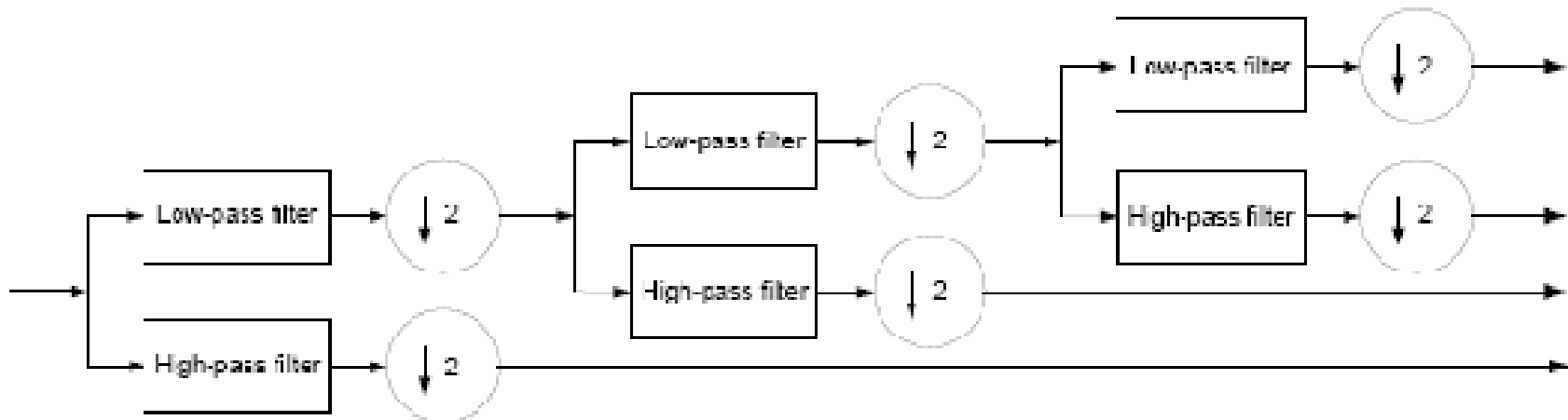
- $h_0$  is orthogonal to  $g_1$
- $h_1$  is orthogonal to  $g_0$

$$y_{\text{low}}[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h_0[2n-k]$$

$$y_{\text{high}}[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h_1[2n-k]$$

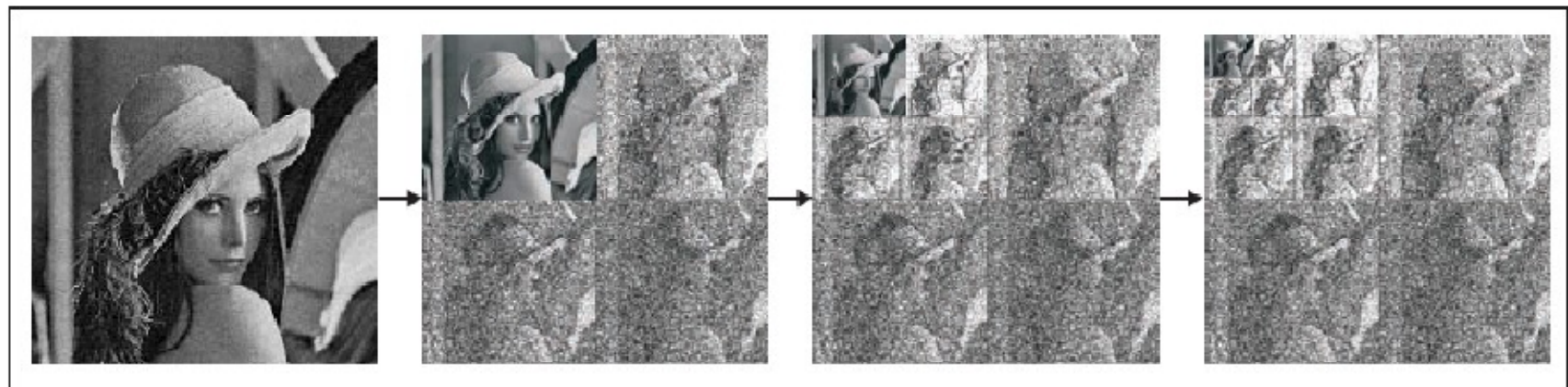
# 1D Dyadic Decomposition

- ★ After a decomposition, most of the energy is located in the low-pass band.
- ★ Successive applications of the filters on the low-pass outputs results in a dyadic decomposition, i.e. the number of coefficients for each novel lower band is half the number for the previous decomposition.



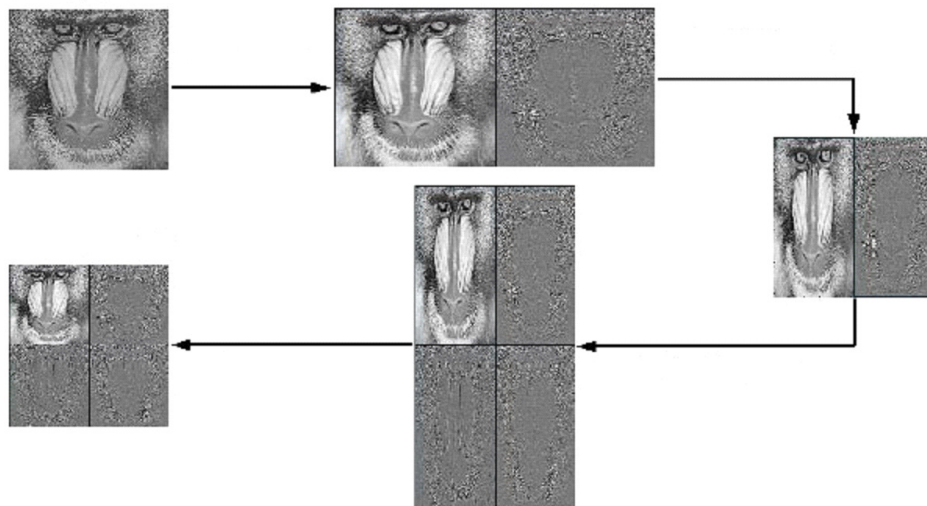
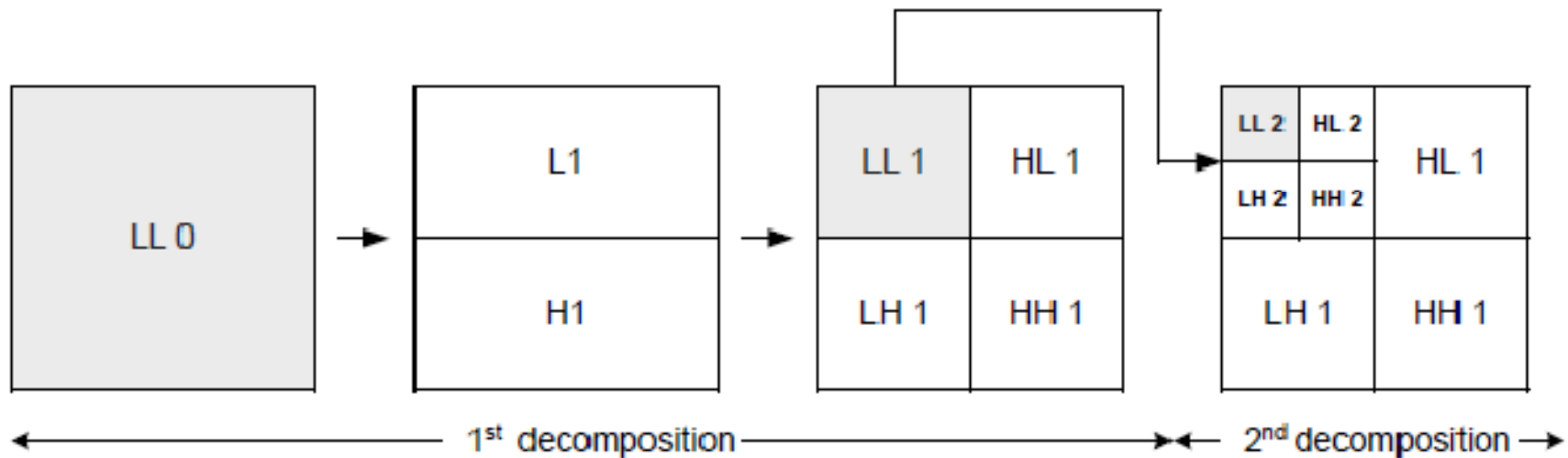
# 2D Dyadic Decomposition

- ★ After a decomposition, most of the energy is located in the low-pass band.
- ★ Successive applications of the filters on the low-pass outputs results in a dyadic decomposition, i.e. the number of coefficients for each novel lower band is  $1/4$  the number for the previous decomposition.



**Example with 3 decompositions !**

# 2D Bi-Orthogonal DWT: Filtering + Subsampling



**The bidimensional (2D) transformation results from applying a unidimensional (1D) transformation, first to the rows and after to the columns.**

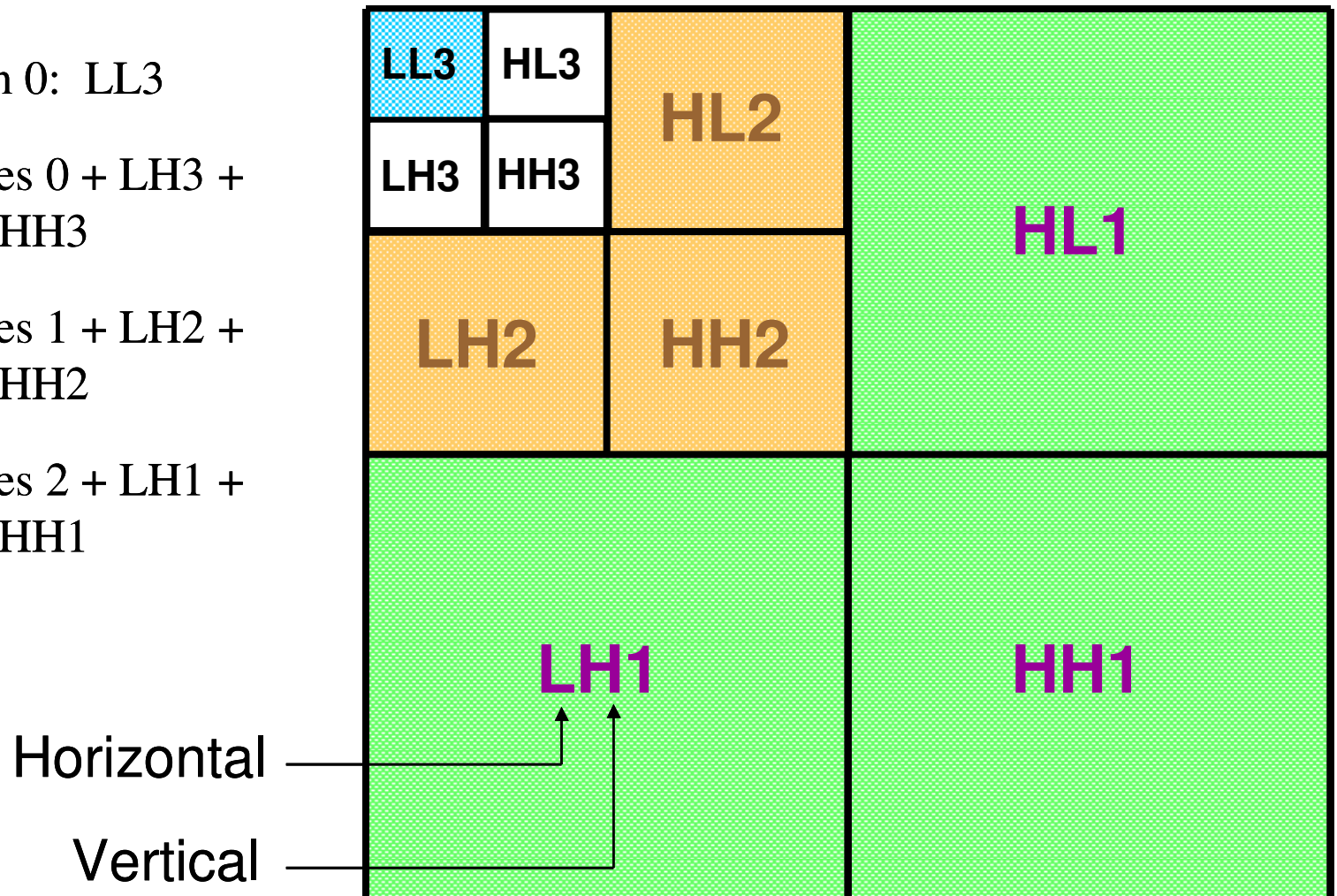
# 2D Wavelet (Dyadic) Decomposition

Resolution 0: LL3

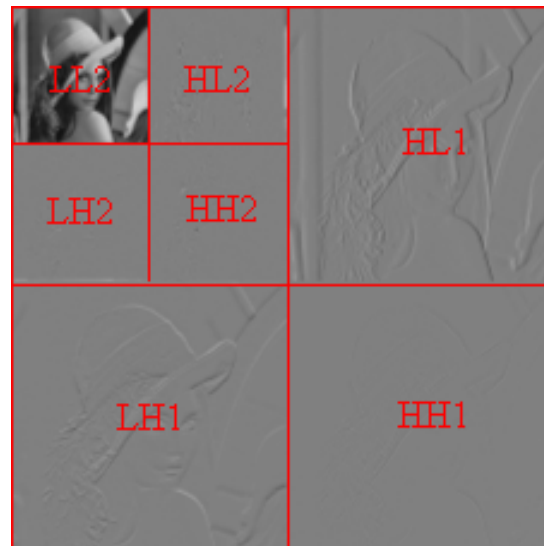
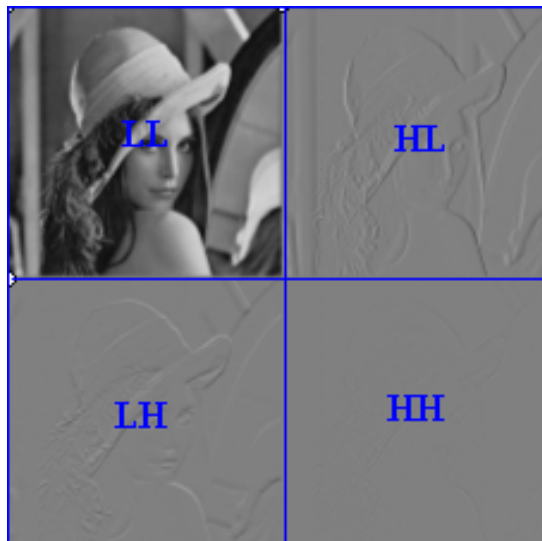
Res 1: Res 0 + LH3 +  
HL3 + HH3

Res 2: Res 1 + LH2 +  
HL2 + HH2

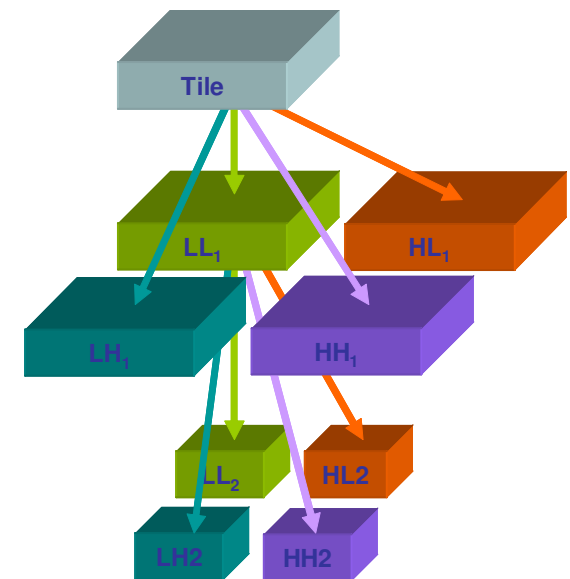
Res 3: Res 2 + LH1 +  
HL1 + HH1



# Two-Levels DWT Decomposition



Usually, the DWT is applied 4 to 8 times to an image; in JPEG 2000, five (5) decompositions are used by default.





# JPEG 2000 DWT Filters

## ★ Irreversible Daubechies (9,7)

$n$	$h_0(n)$	$n$	$h_1(n)$
0	+0.602949018236	-1	+1.115087052456
$\pm 1$	+0.266864118442	-2, 0	-0.591271763114
$\pm 2$	-0.078223266528	-3, 1	-0.057543526228
$\pm 3$	-0.016864118442	-4, 2	+0.091271763114
$\pm 4$	+0.026748757410		

## ★ Reversible (5,3), derived from Le Gall (5,3)

Le Gall (5,3)  
(not exactly JPEG 2000's)

$n$	$h_0(n)$	$n$	$h_1(n)$
0	+6/8	-1	+1
$\pm 1$	+2/8	-2, 0	-1/2
$\pm 2$	-1/8		

## ★ In addition, Part II allows for arbitrary filters (user defined)



# Irreversible Daubechies (9,7) Filter

$$h_0(z) = h_0^0 + h_0^1(z^1 + z^{-1}) + h_0^2(z^2 + z^{-2}) + h_0^3(z^3 + z^{-3}) + h_0^4(z^4 + z^{-4})$$

$$h_1(z) = h_1^0 + h_1^1(z^1 + z^{-1}) + h_1^2(z^2 + z^{-2}) + h_1^3(z^3 + z^{-3})$$

$$h_0^0 = 0.602949018236$$

$$h_0^1 = 0.266864118443$$

$$h_0^2 = -0.078223266529$$

$$h_0^3 = -0.016864118443$$

$$h_0^4 = 0.026748757411$$

$$h_1^0 = 1.115087052456994$$

$$h_1^1 = -0.5912717631142470$$

$$h_1^2 = -0.05754352622849957$$

$$h_1^3 = 0.09127176311424948$$

## Reversible Le Gall (5/3) Filter

$$h_0(z) = -1/8 z^{-2} + 1/4 z^{-1} + 3/4 + 1/4 z^1 - 1/8 z^2$$

$$h_1(z) = -1/2 z^{-1} + 1 - 1/2 z^1$$

$$y_{\text{low}}[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h_0[2n-k]$$

$$y_{\text{high}}[n] = \sum_{k=-\infty}^{\infty} x[k] \cdot h_1[2n-k]$$

**Filtering equations**



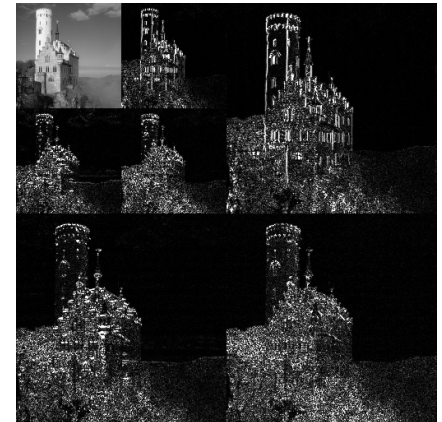
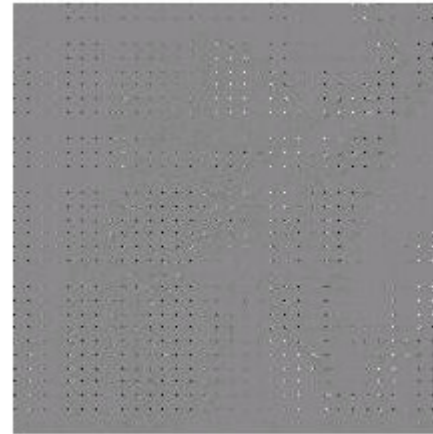
# JPEG 2000 Artefacts: Ringing

**Ringling, not  
block effect !**





# DCT versus DWT



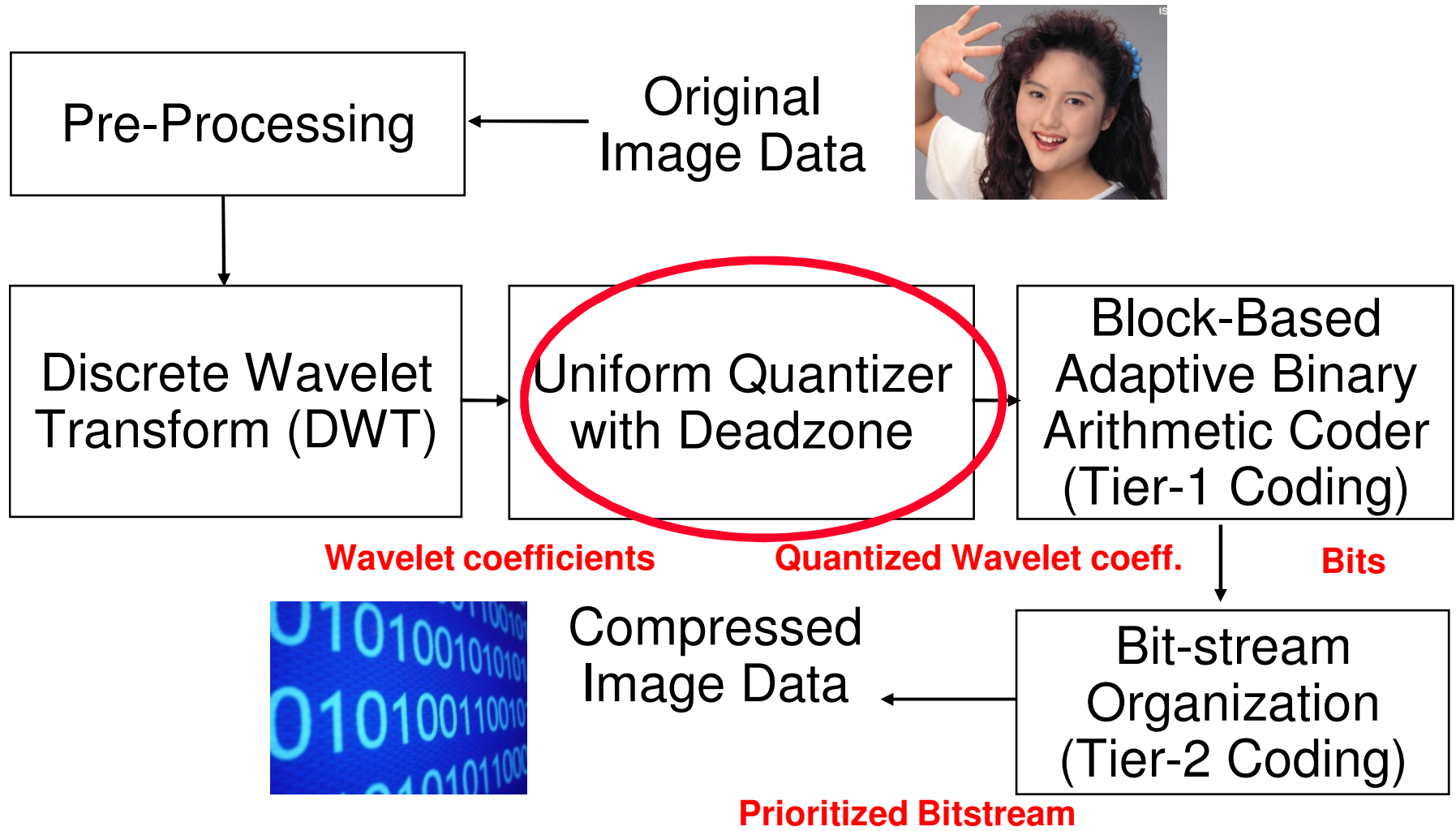
## SIMILAR

1. The total number of coefficients per image

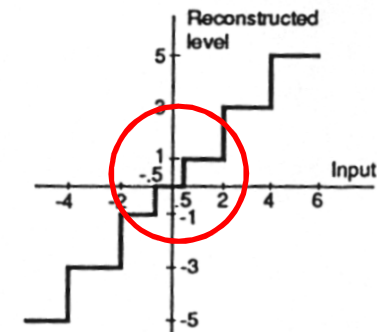
## DIFFERENT

1. Block-based DCT leads to block artifacts while image-based DWT leads to ringing artifacts
2. DCT coefficients are for a block while DWT coefficients are for the whole image
3. DCT coefficients are for a specific spatial resolution while DWT coefficients are implicitly for multiple spatial resolutions

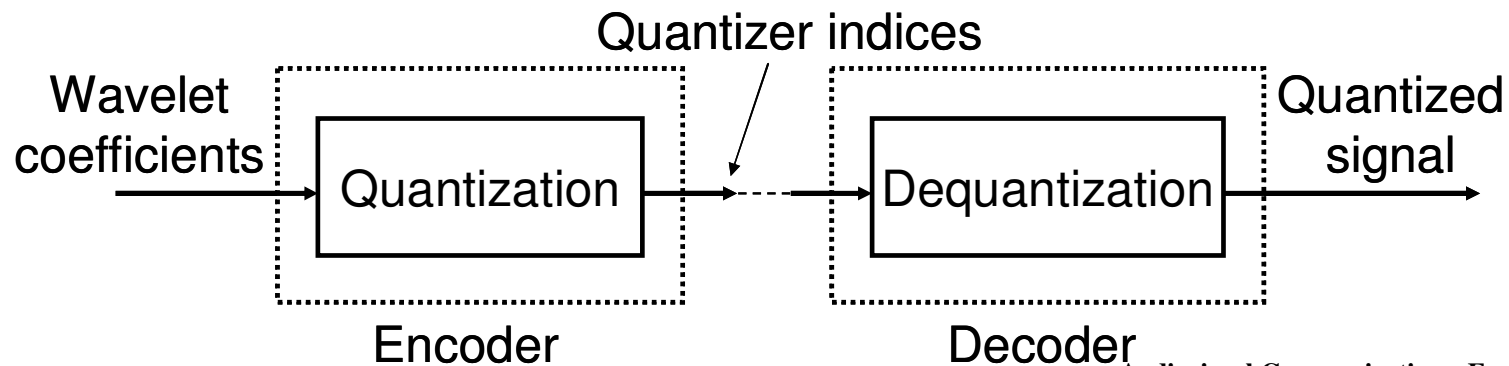
# JPEG 2000 Main Encoder Modules



# JPEG 2000: Quantization

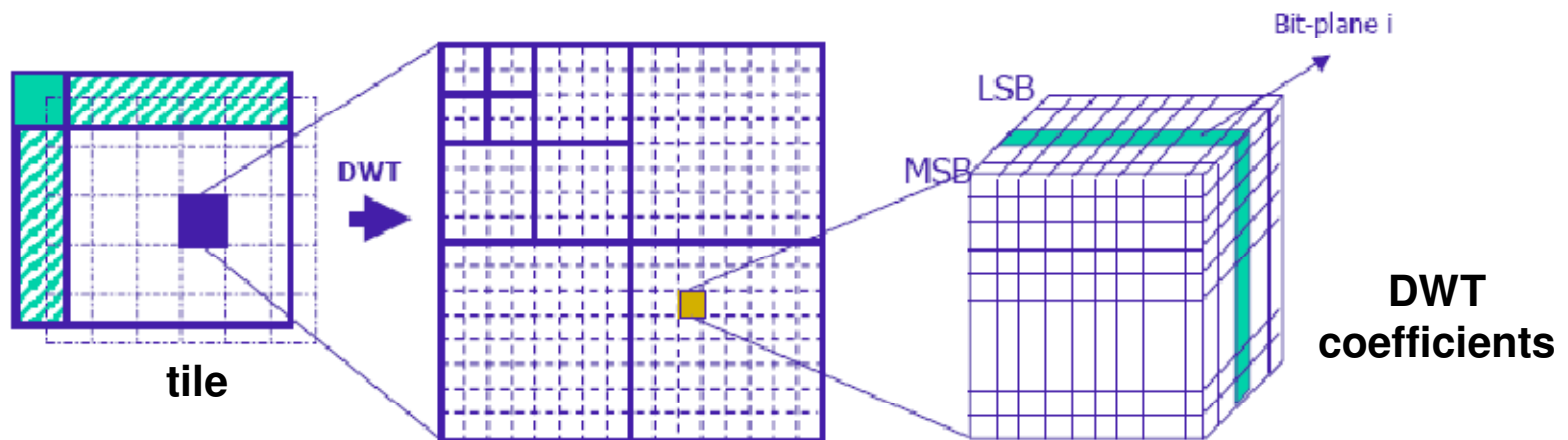


- ★ Quantization defines the trade-off between compression and quality.
- ★ Uniform quantization with deadzone is used to quantize all the wavelet coefficients.
- ★ For each subband  $b$ , a basic quantizer step size  $\Delta_b$  is selected by the user and used to quantize all the coefficients in that subband; this defines the precision of the wavelet coefficients.
- ★ The choice of the quantizer step size for each subband can be based on visual models, such as the contrast sensitivity function (CSF); this allows achieving higher compression ratios for the same visual quality.

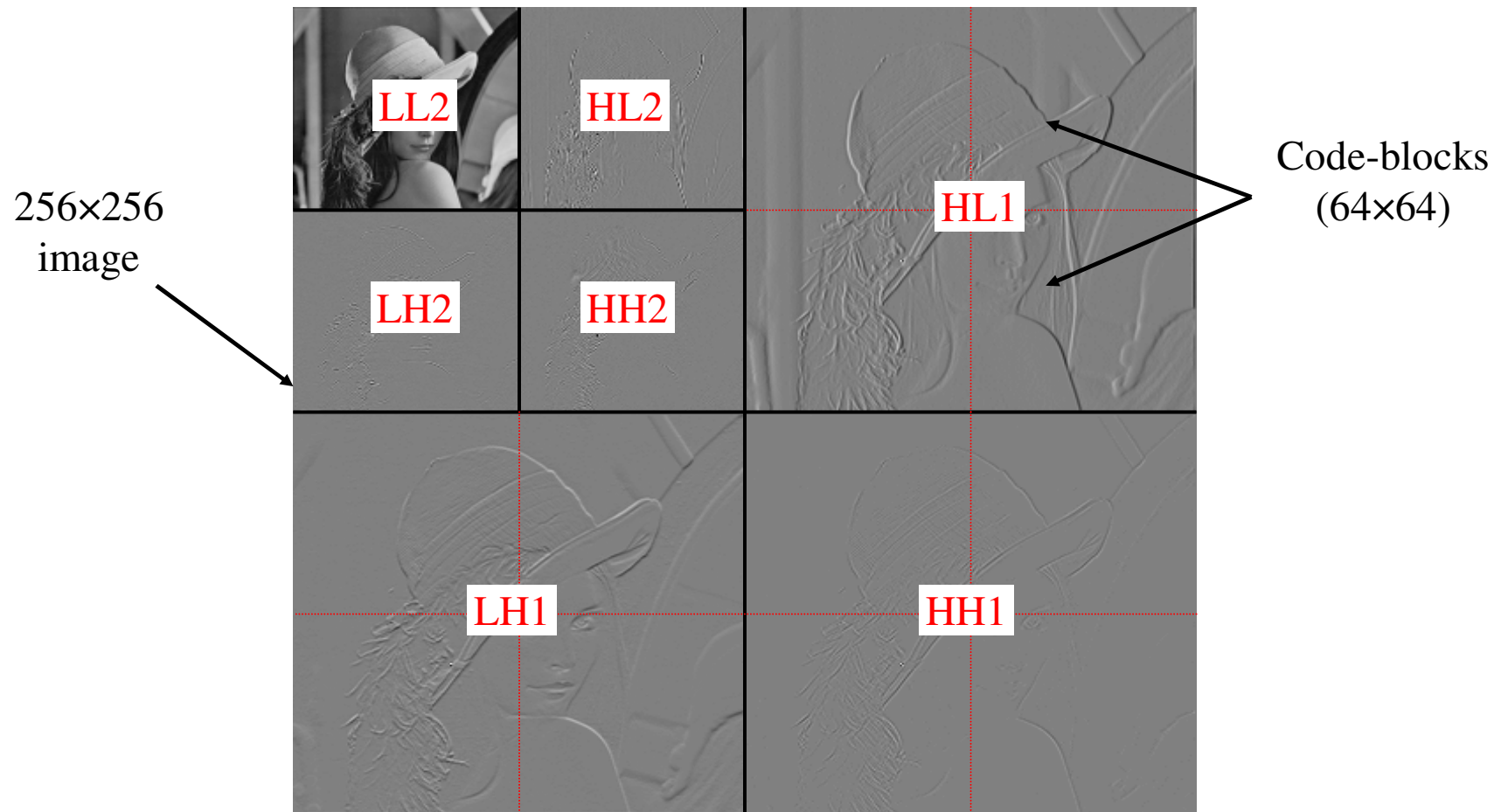


## Code-blocks and Bitplanes ...

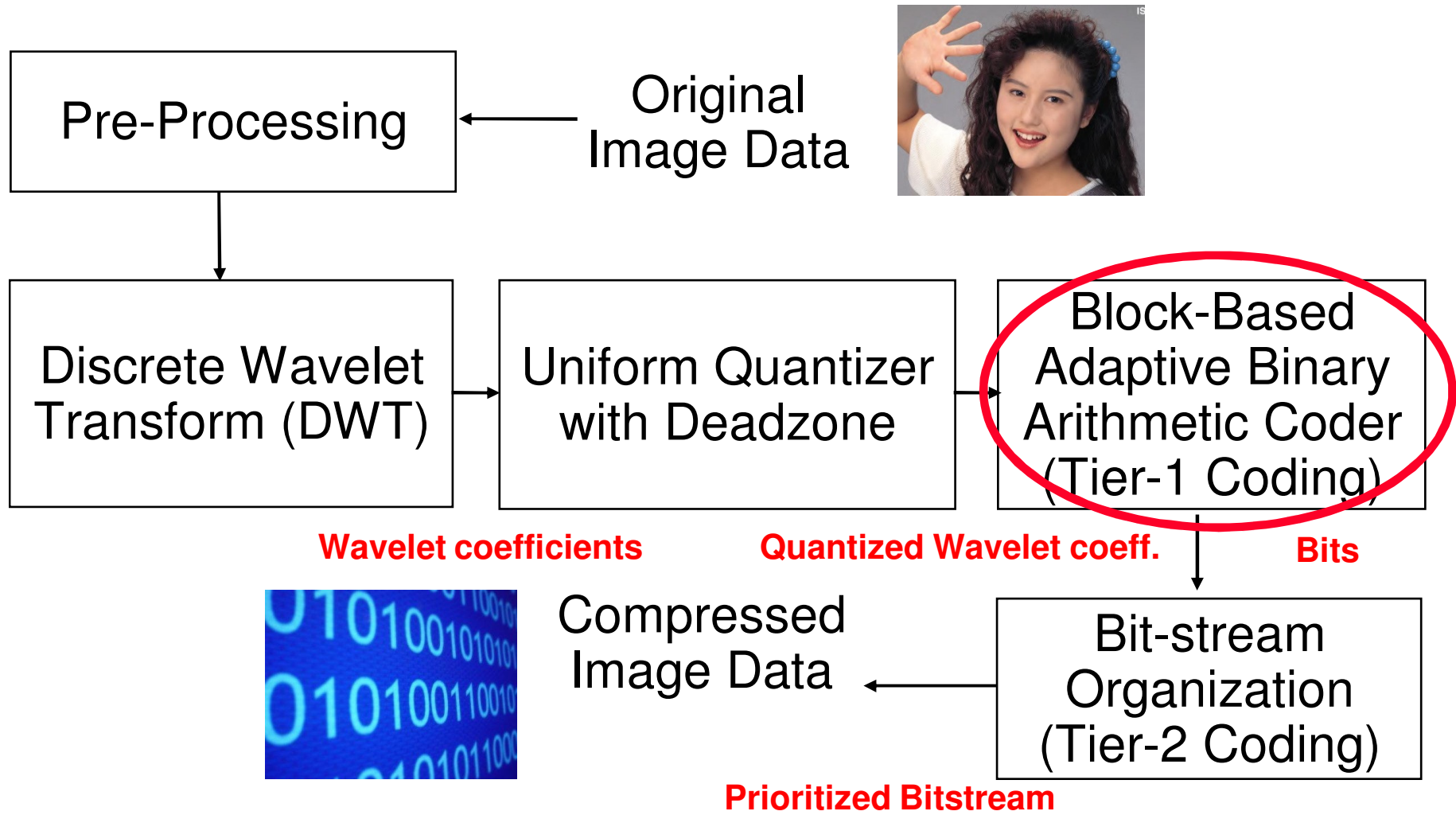
- ★ **CODE-BLOCK** - Each sub-band from each tile component is partitioned into code-blocks (typically  $64 \times 64$  DWT coefficients).
- ★ **BITPLANE** - Each code-block will be independently entropy-encoded, bitplane by bitplane.



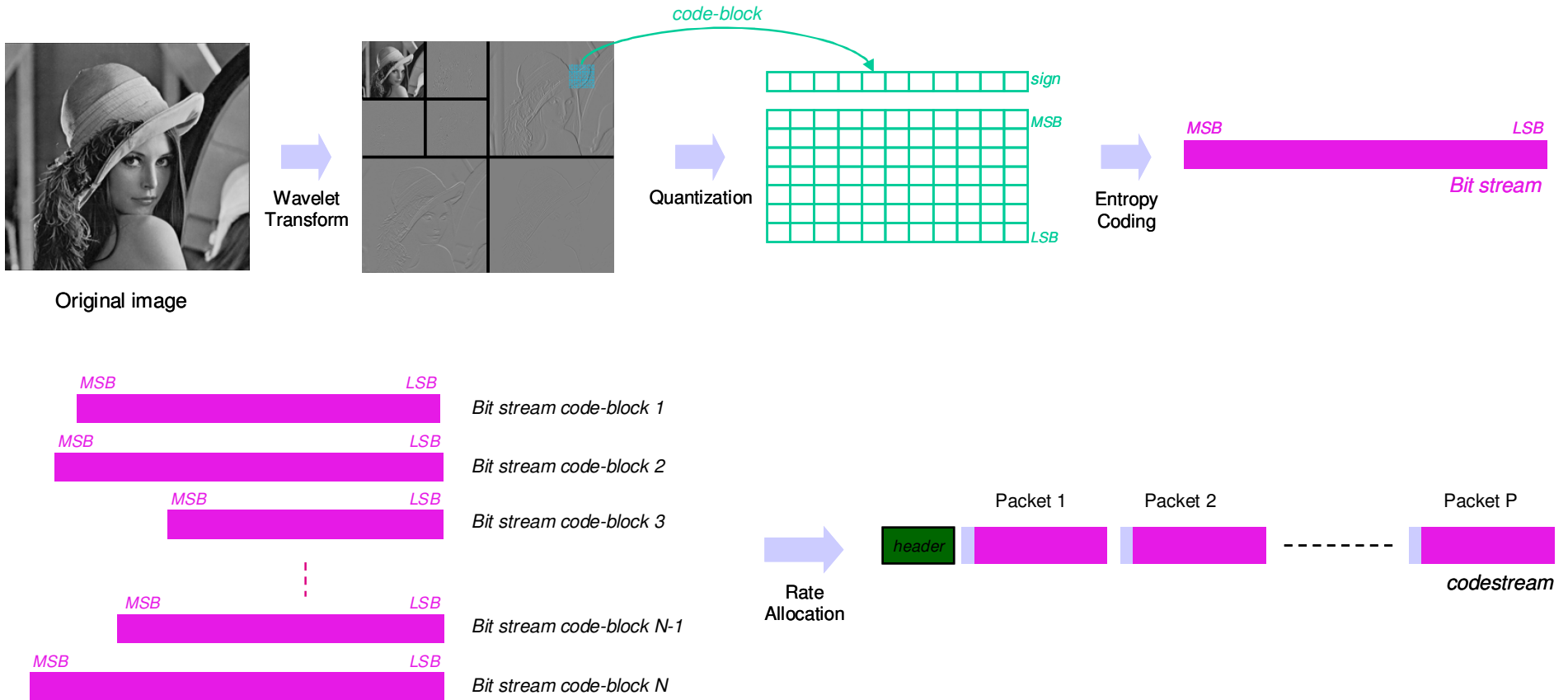
# Bitplane Pass Coded Data: Example



# JPEG 2000 Main Encoder Modules

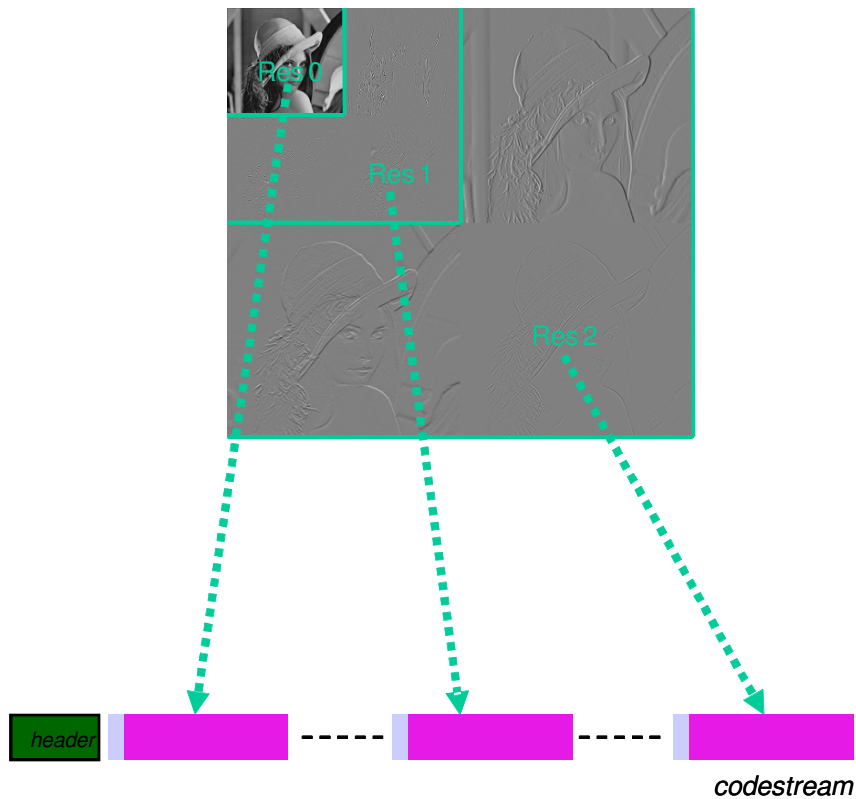


# JPEG 2000: Encoder Overview

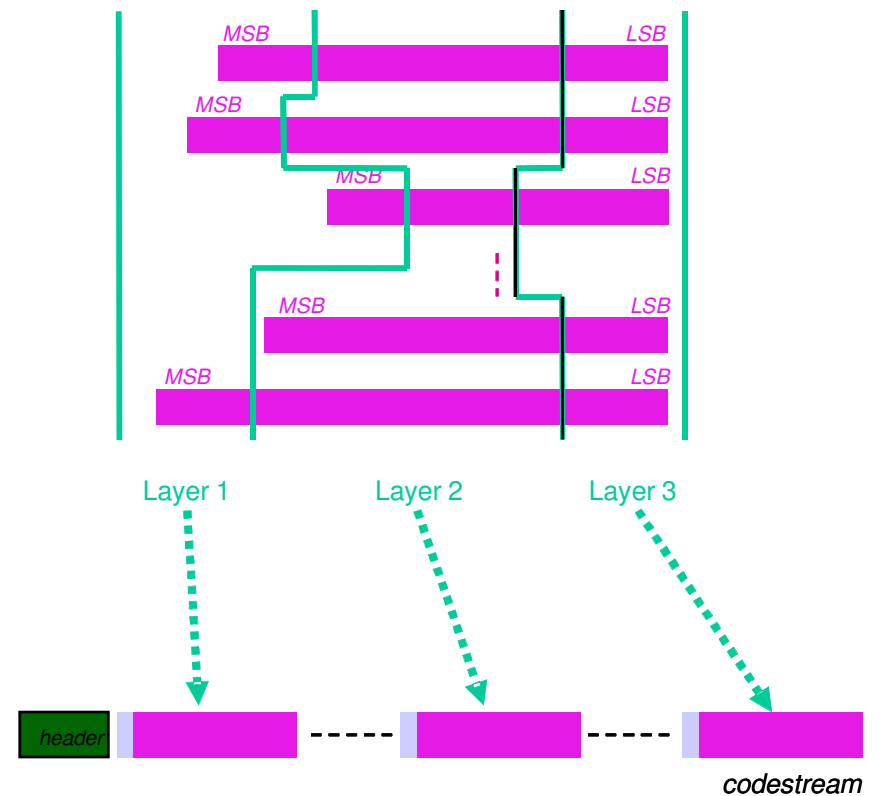


# JPEG 2000 Progressive Coding

## Progressive by resolution



## Progressive by quality





# JPEG 2000 Entropy Coding: Two Tiers

JPEG 2000 entropy coding considers two main phases:

★ **TIER-1** – Regards the entropy coding of the DWT coefficients

- **DWT Coefficients are coded through the binary values of the sample in a block of a bitplane of a subband**
- **Tier-1 has 2 main parts:**
  - **Coefficient bit modeling (EBCOT) - chooses which data to encode first for each bitplane**
  - **Arithmetic coding (MQ) - reduces redundancy inside the binary sequence**

★ **TIER-2** – Regards the organization and generation of the final codestream



# JPEG 2000 Tier-1: Coefficient Bit Modeling

**Target: To encode first the bits providing the largest distortion reduction among all the bits belonging to the code-block. This allows obtaining an optimal bitstream whatever the truncation point.**

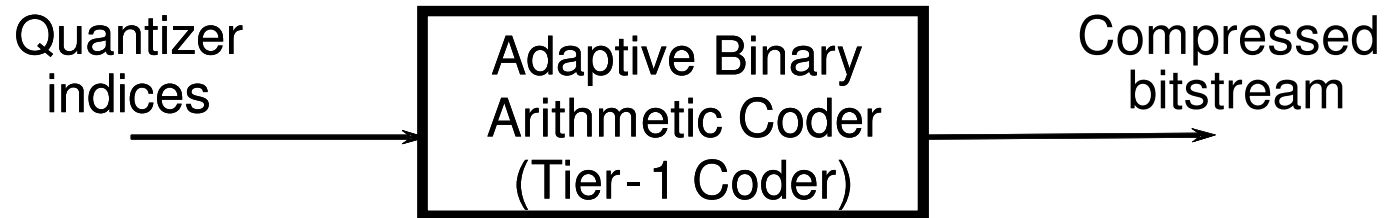
★ **Method:**

- **Encode bitplanes from MSB to LSB, and**
- **Encode each bitplane in three successive coding passes**

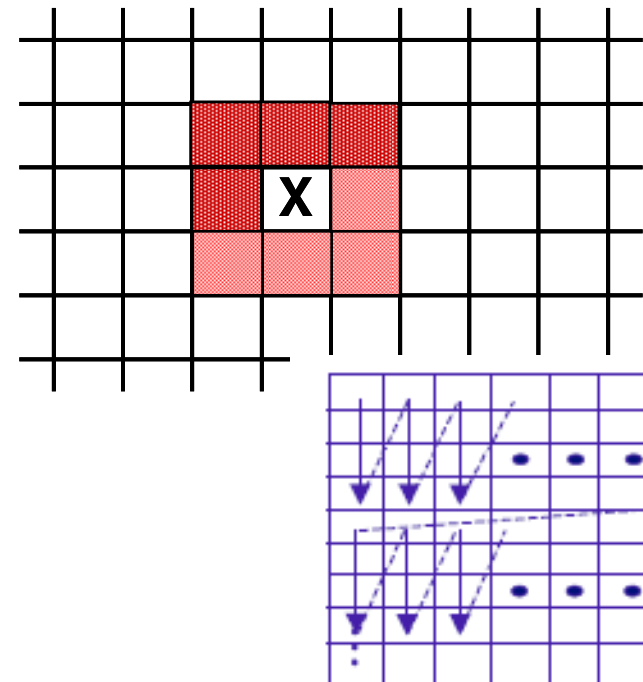
★ **Bit modeling considers three successive coding passes (*significance, refinement and clean up*)**

- **Bits from a bitplane are grouped in three categories - *significance, refinement and clean up* - depending on the distortion reduction they will bring**
- **The category providing the largest distortion reduction is encoded first**
- **The three coding passes provide two additional truncation points per bitplane**

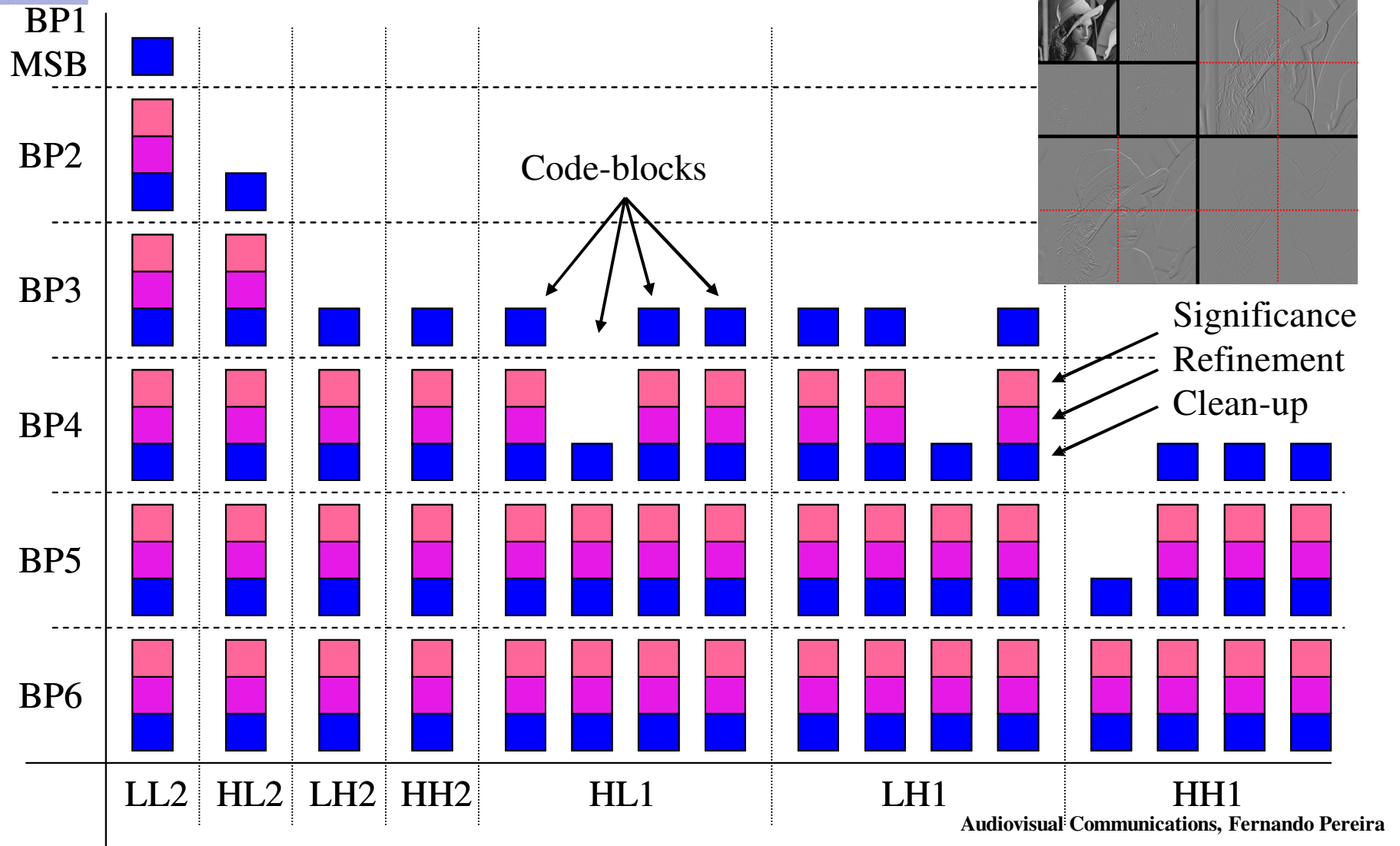
# JPEG 2000 Tier-1: Entropy Coding



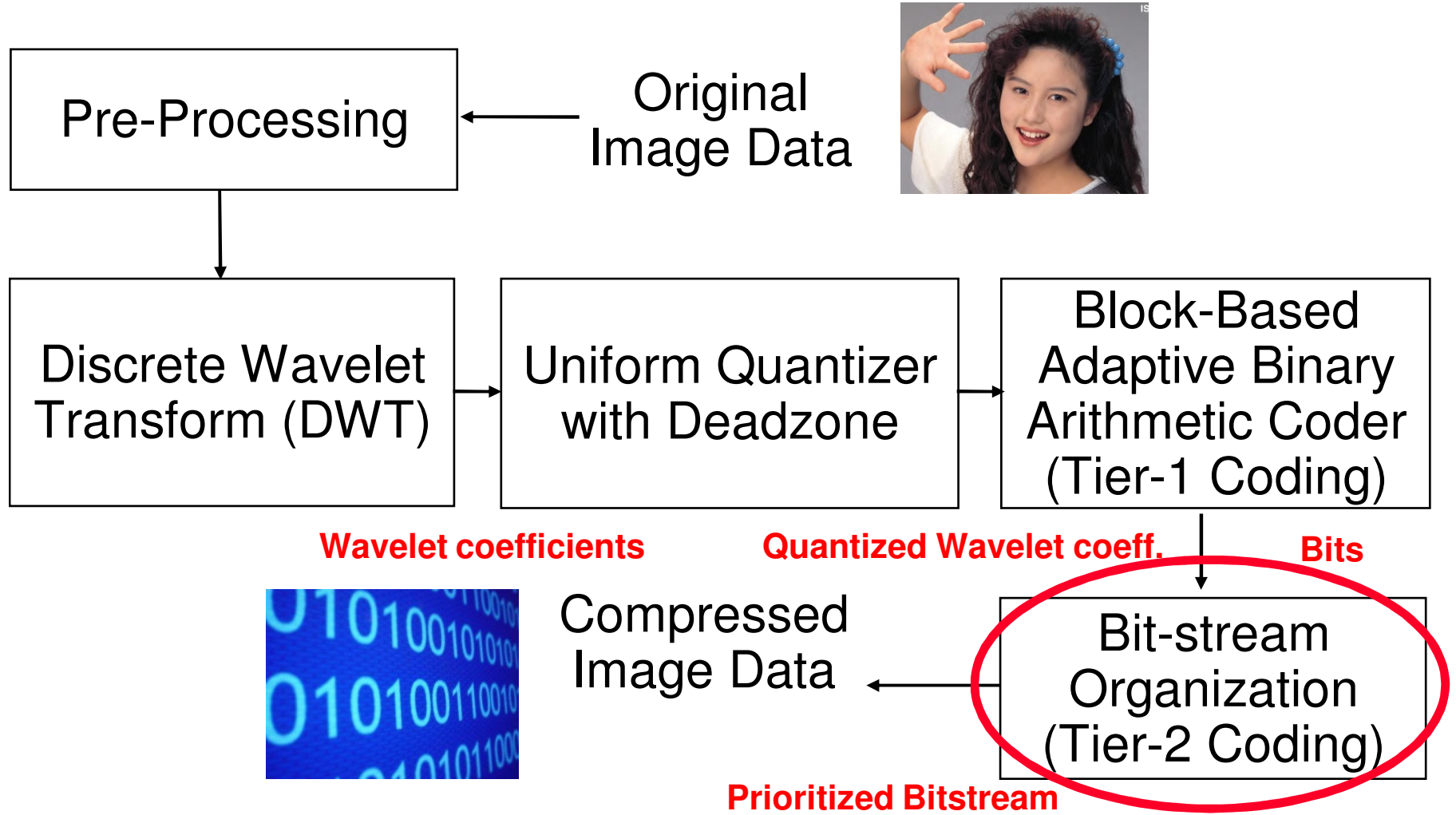
- ★ **Context-based adaptive binary arithmetic coding is used in JPEG 2000 to efficiently compress each individual bitplane.**
- ★ **The binary value of a sample in a block of a bitplane of a subband is coded as a binary symbol with the JBIG2 MQ-Coder which is an arithmetic entropy encoder.**



# Bitplane Pass Coded Data: Example



# JPEG 2000 Main Encoder Modules





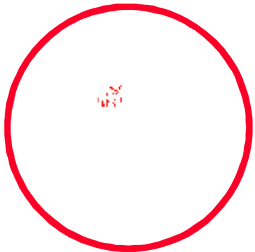
# JPEG 2000: Tier-2 Role

- ★ **Tier-1 generates a collection of bitstreams**
  - **One independent bitstream for each code block bitplane**
  - **Each bitstream is embedded following the three coding passes**
- ★ **Tier-2 multiplexes the bitstreams selected for inclusion in the codestream and signals the ordering of the resulting coded bitplane passes in an efficient manner**
- ★ **Tier-2 coded data can be rather easily parsed**
- ★ **Tier-2 enables SNR, resolution, spatial, ROI and arbitrary progression scalability**



Sig. Prop. =	0
Refine =	0
Cleanup =	21
<hr/>	
Total Bytes	21

Bit plane 1  
Compression ratio = 12483 : 1  
RMSE = 39.69 PSNR = 16.16 db  
% refined = 0 % insig. = 99.99



**Quality scalability: Specific bitplanes of all DWT coefficients**



Sig. Prop.	=	38
Refine	=	13
Cleanup	=	57
<hr/>		
Total Bytes		108

Bit plane      3  
Compression ratio = 1533 : 1  
RMSE = 21.59    PSNR = 21.45 db  
% refined = 0.05 % insig. = 99.89

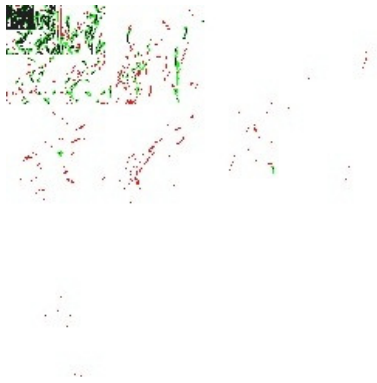


**Quality scalability: Specific bitplanes of all DWT coefficients**



Sig. Prop. =	224
Refine =	73
Cleanup =	383
<hr/> Total Bytes	680

Bit plane 5  
Compression ratio = 233 : 1  
RMSE = 12.11 PSNR = 26.47 db  
% refined = 0.23 % insig. = 99.43

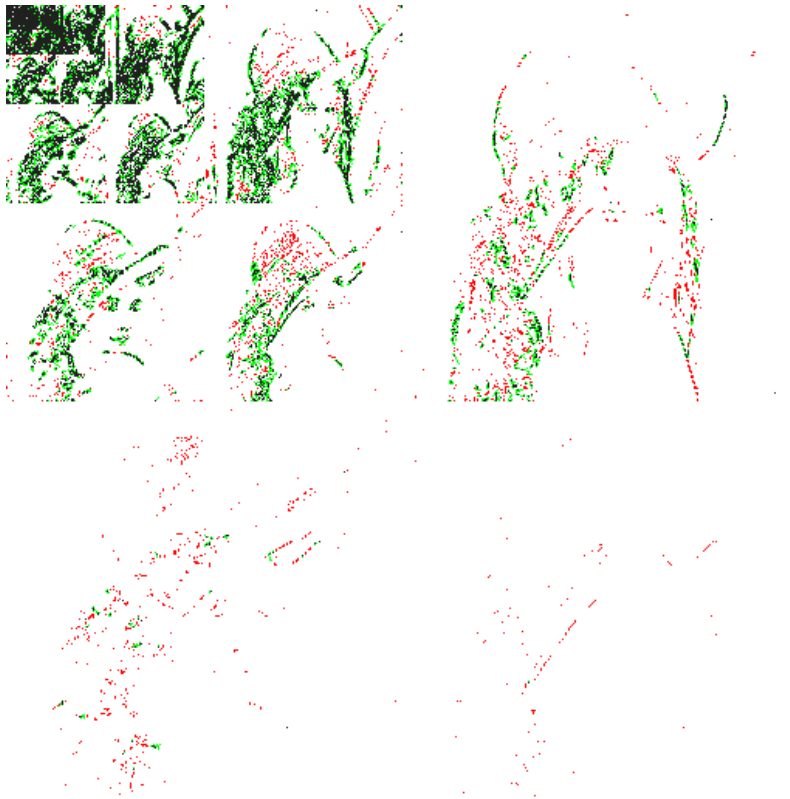


**Quality scalability: Specific bitplanes of all DWT coefficients**



Sig. Prop.	=	2315
Refine	=	932
Cleanup	=	2570
<hr/>		
Total Bytes		5817

Bit plane      8  
Compression ratio = 23 : 1  
RMSE = 4.18      PSNR = 35.70 db  
% refined = 2.91 % insig. = 93.99

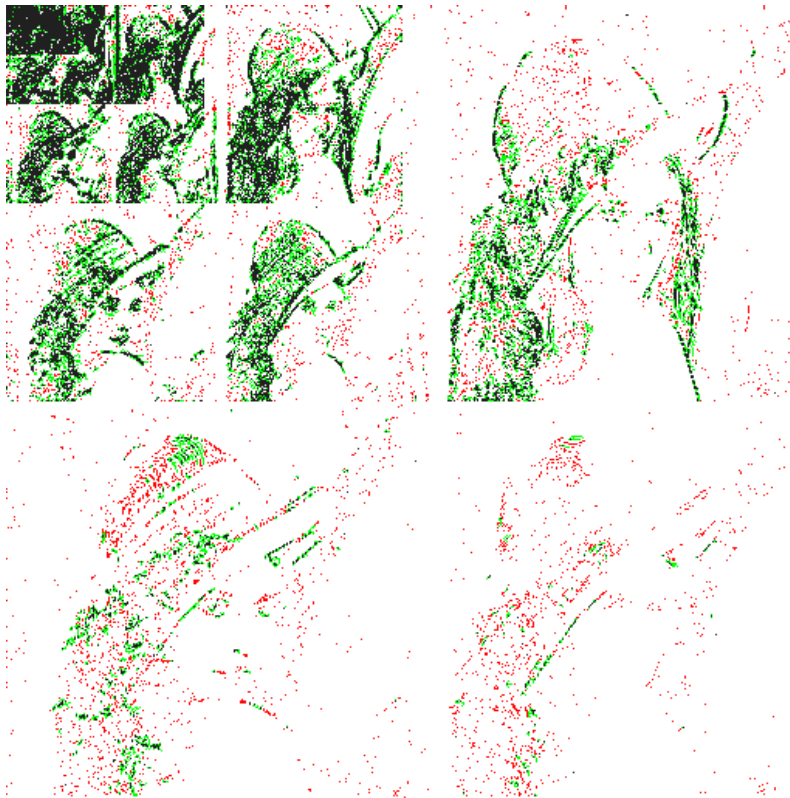


**Quality scalability: Specific bitplanes of all DWT coefficients**



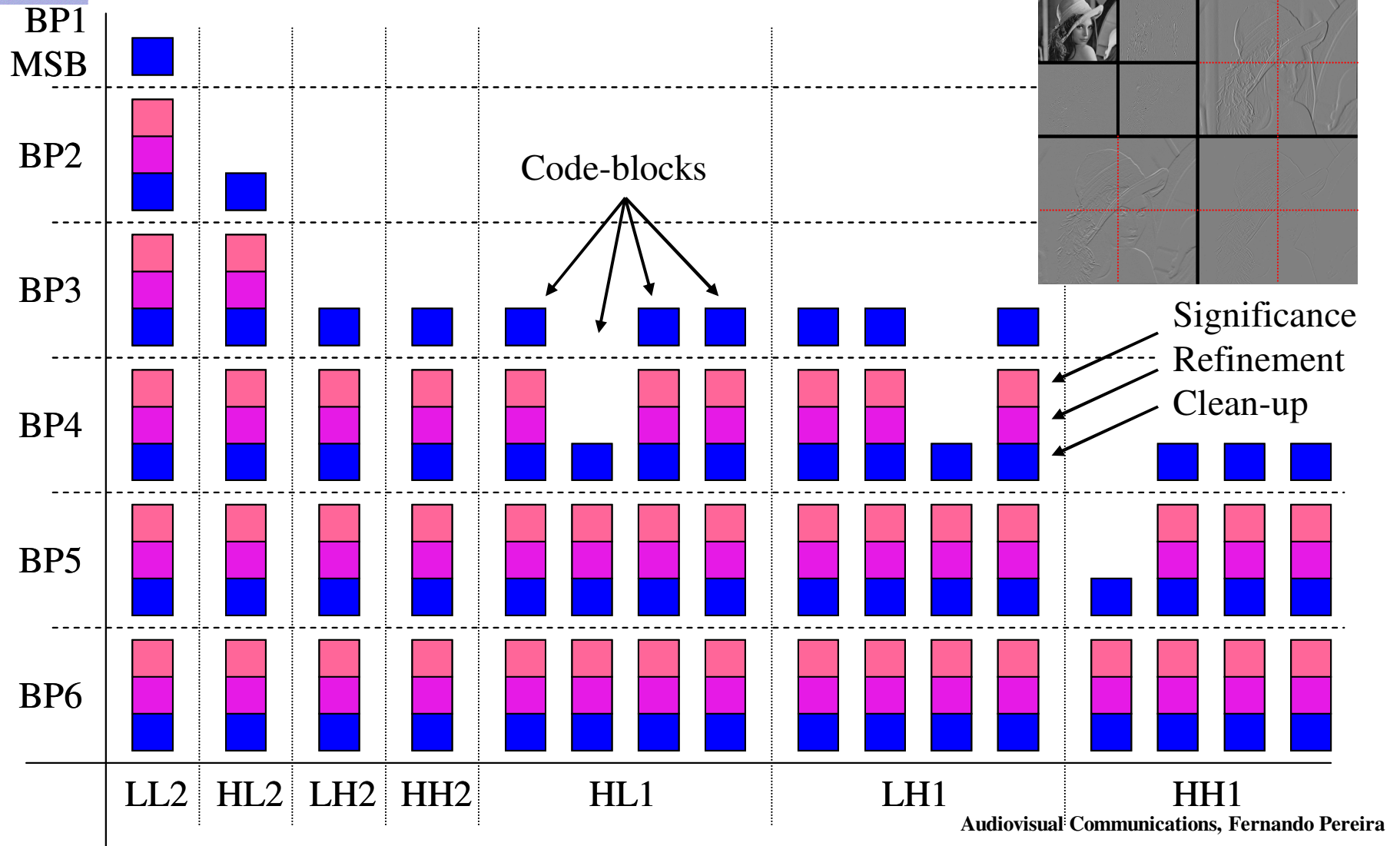
Sig. Prop.	=	4593
Refine	=	1925
Cleanup	=	5465
<hr/>		
Total Bytes		11983

Bit plane      **9**  
Compression ratio = 11.2 : 1  
RMSE = 2.90      PSNR = 38.87 db  
% refined = 6.01 % insig. = 87.66

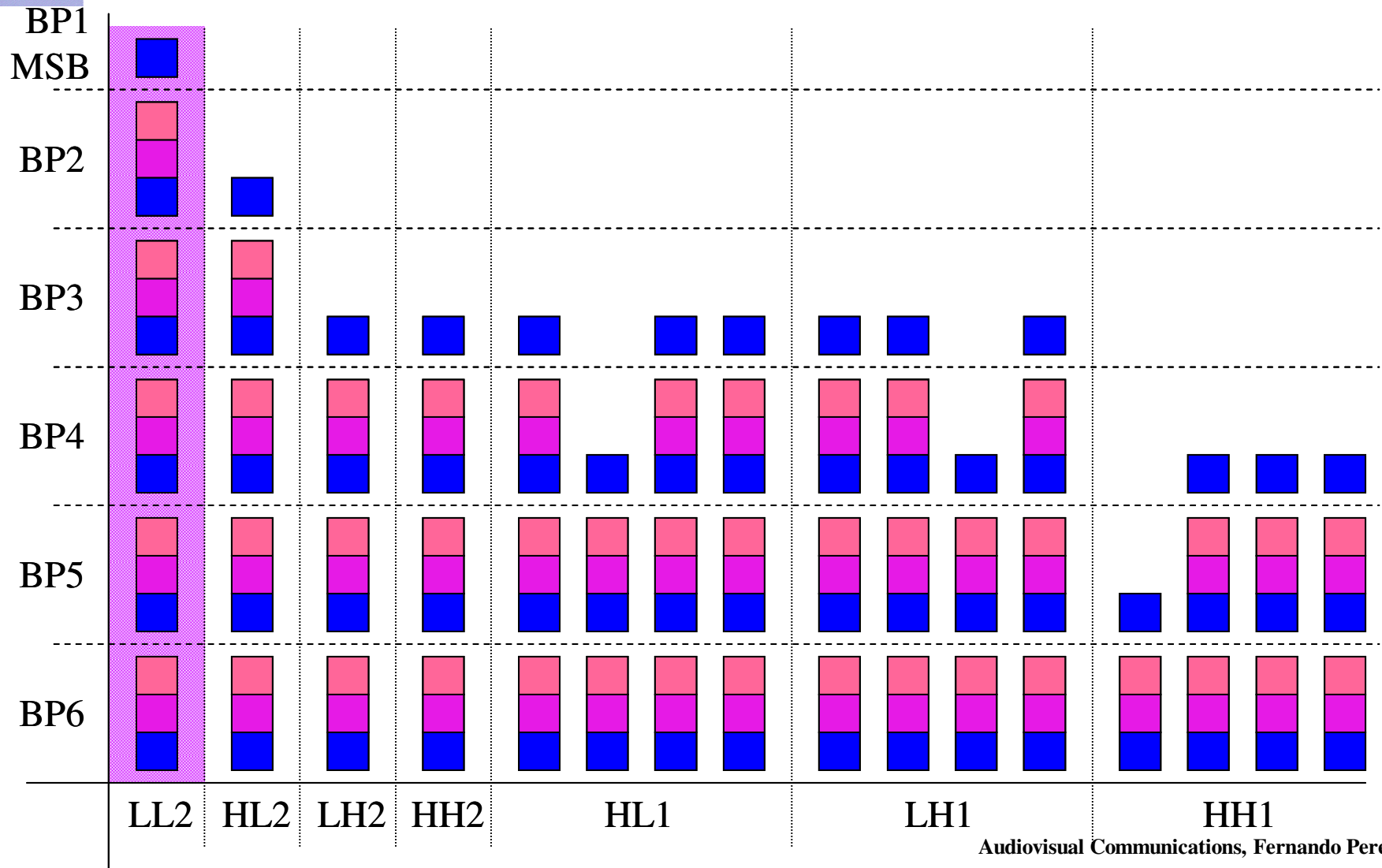


**Quality scalability: Specific bitplanes of all DWT coefficients**

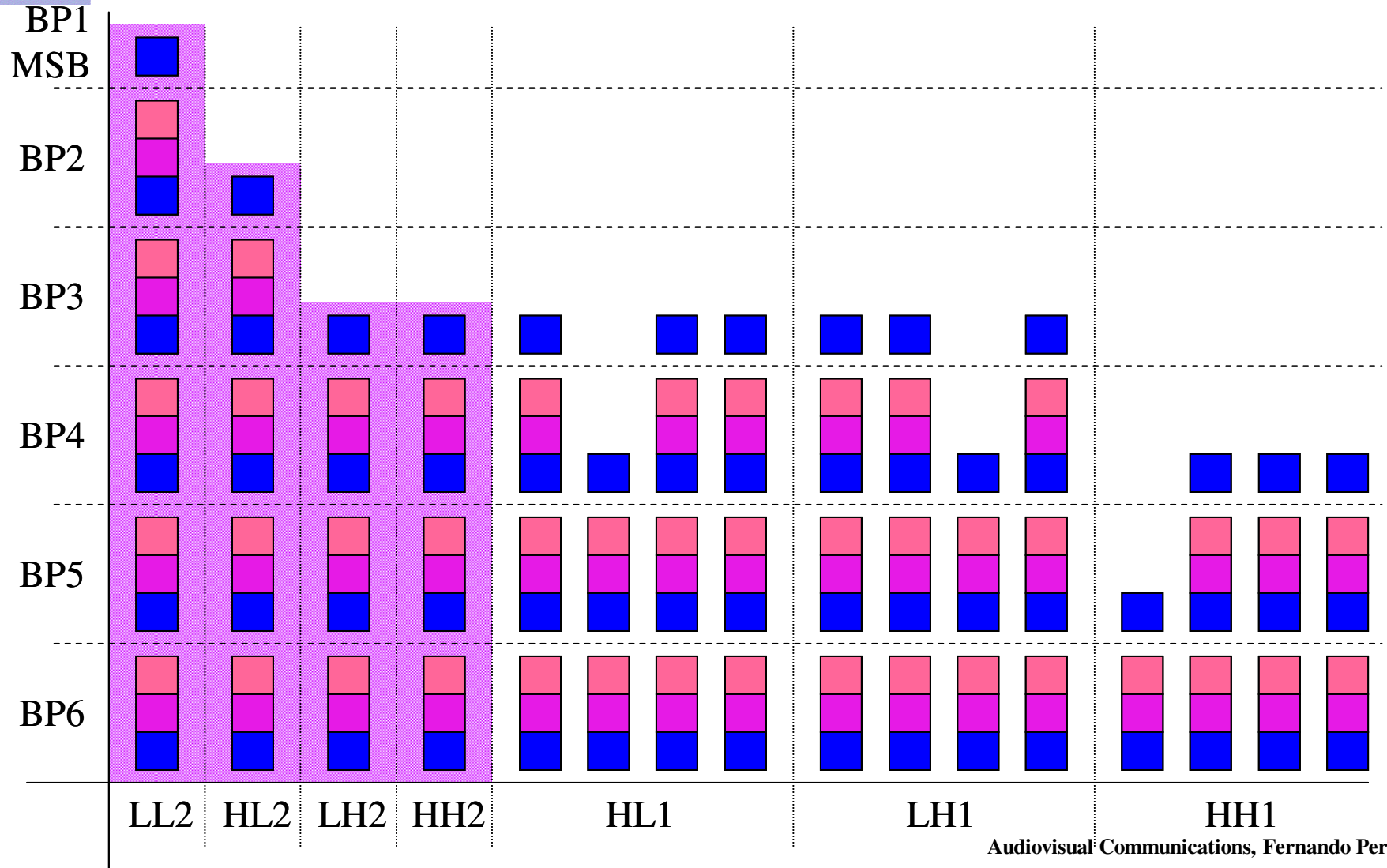
# Bitplane Pass Coded Data: Example



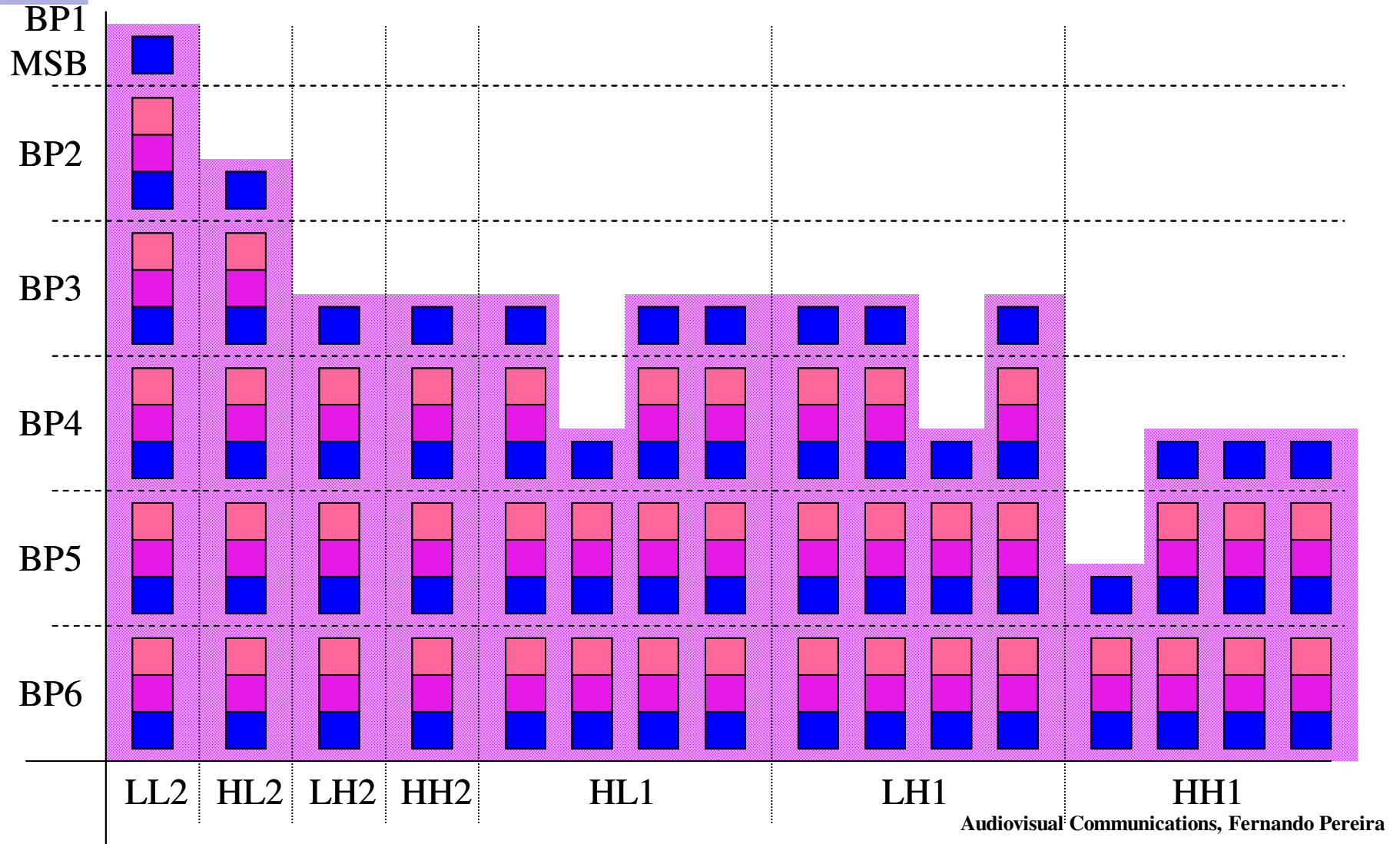
# Lowest Resolution, Highest Quality



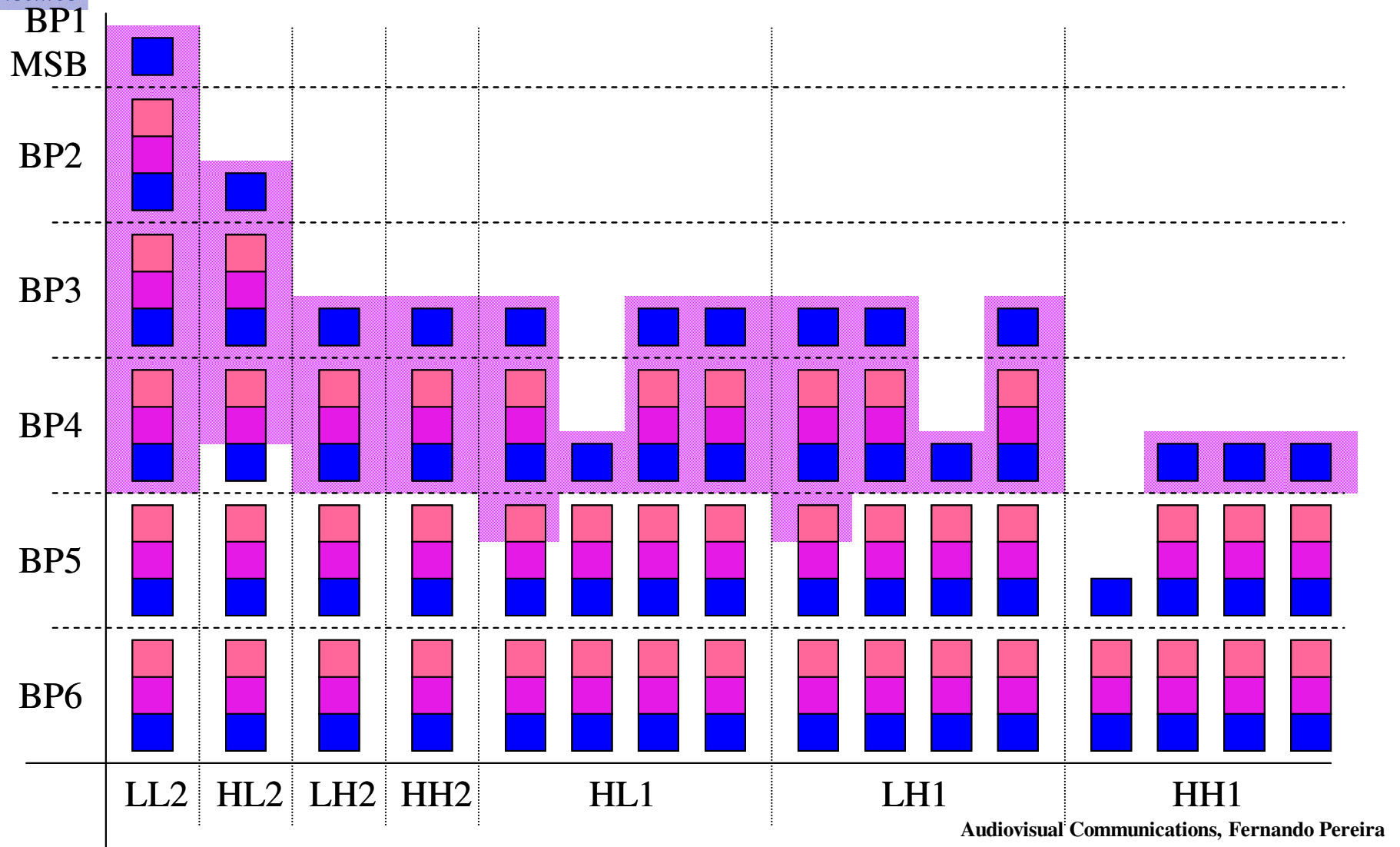
# Medium Resolution, Highest Quality



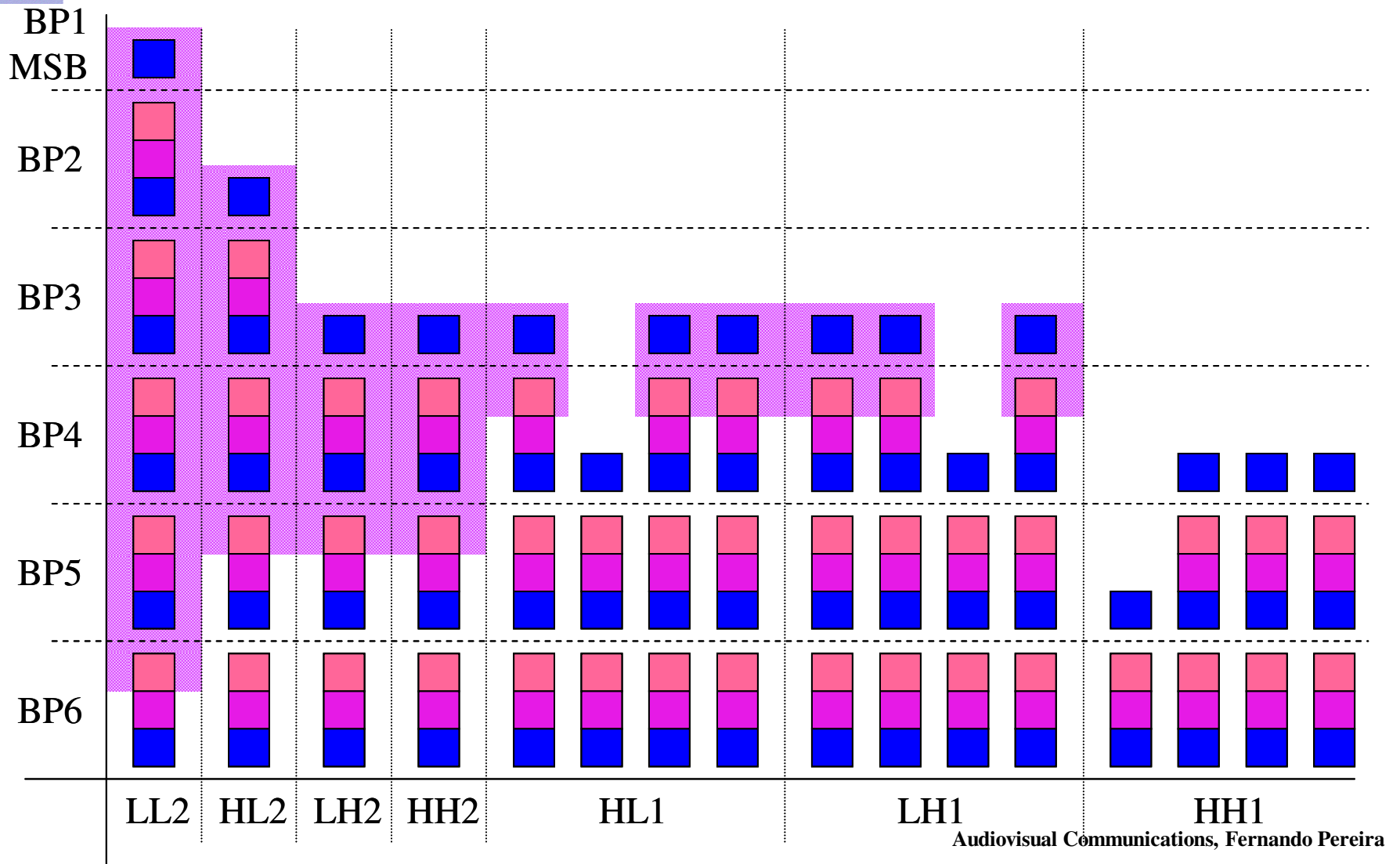
# Highest Resolution, Highest Quality



# Highest Resolution, Target SNR Quality



# Highest Resolution, Target Visual Quality

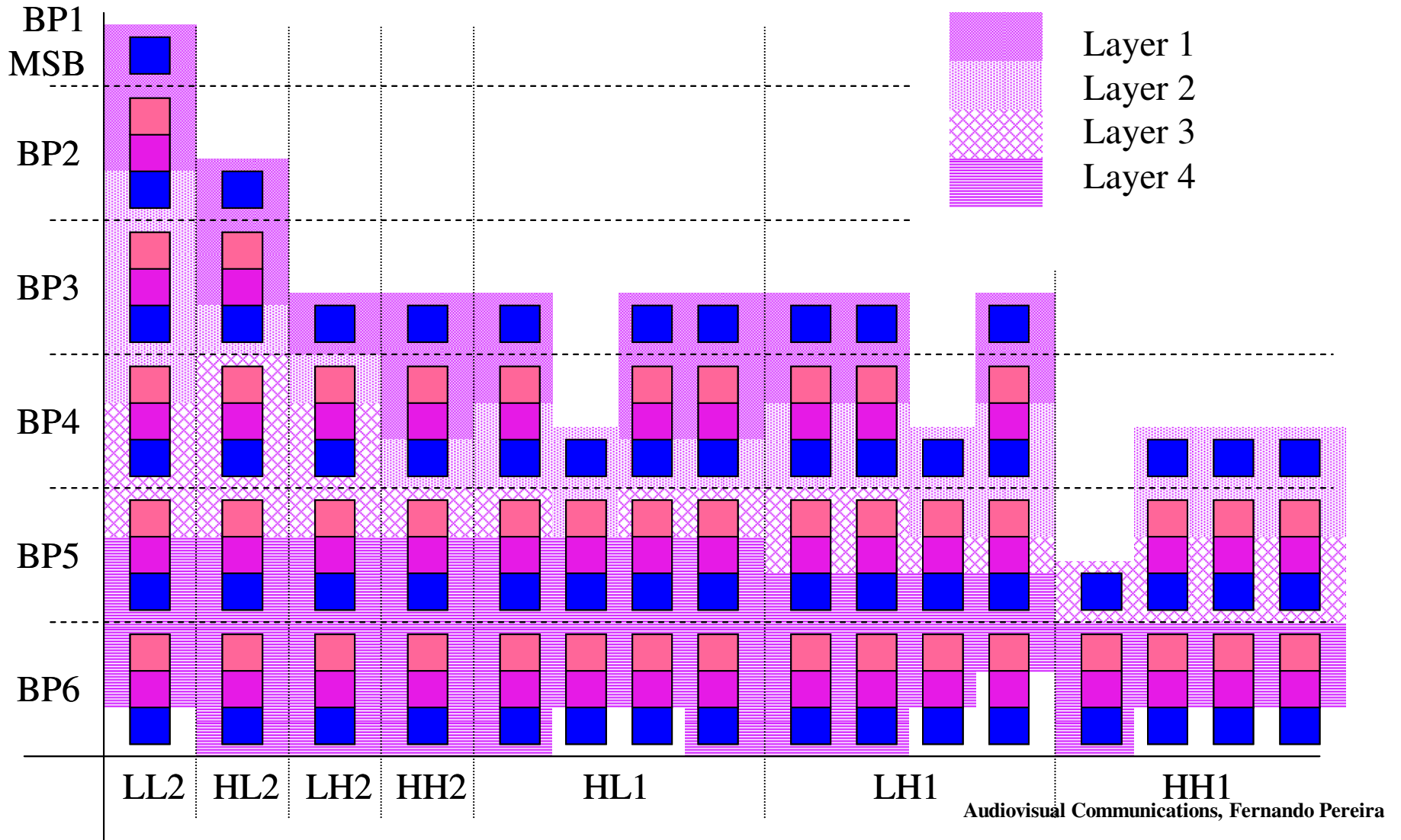




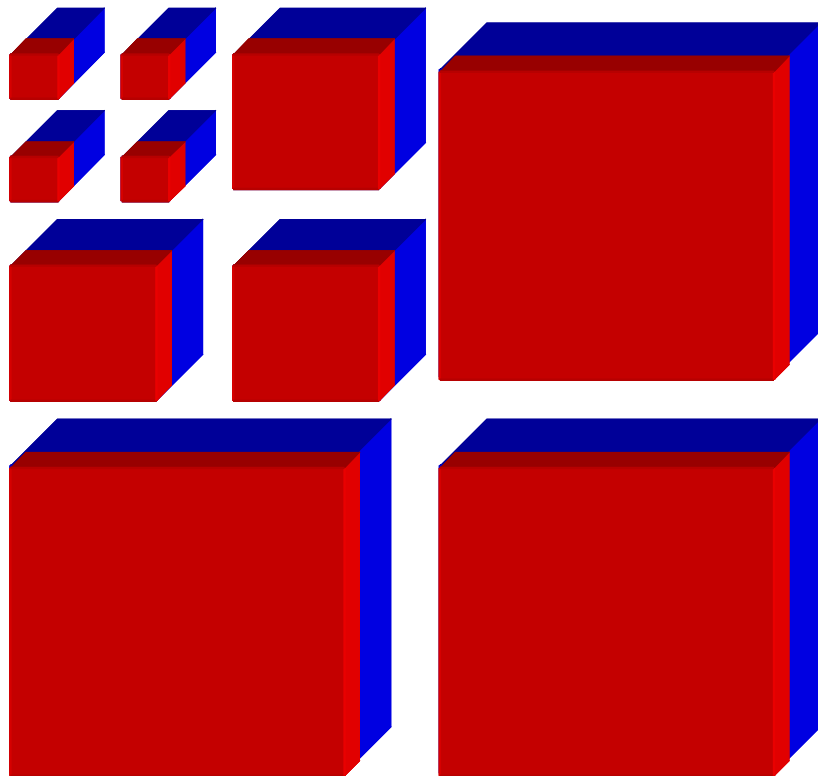
## **JPEG 2000: Layers**

- ★ **Layer: a collection of some consecutive bitplane coding passes from all code-blocks in all subbands and components. Each code-block can contribute an arbitrary number of bitplane coding passes to a layer.**
- ★ **Each layer successively increases the image quality, most often associated with SNR or visual quality levels.**
- ★ **Layers are explicitly signaled and can be arbitrarily determined by the encoder.**
- ★ **The number of layers can range from 1 to 65535, typically around 20; larger numbers are intended for interactive sessions where each layer is generated depending on user feedback.**

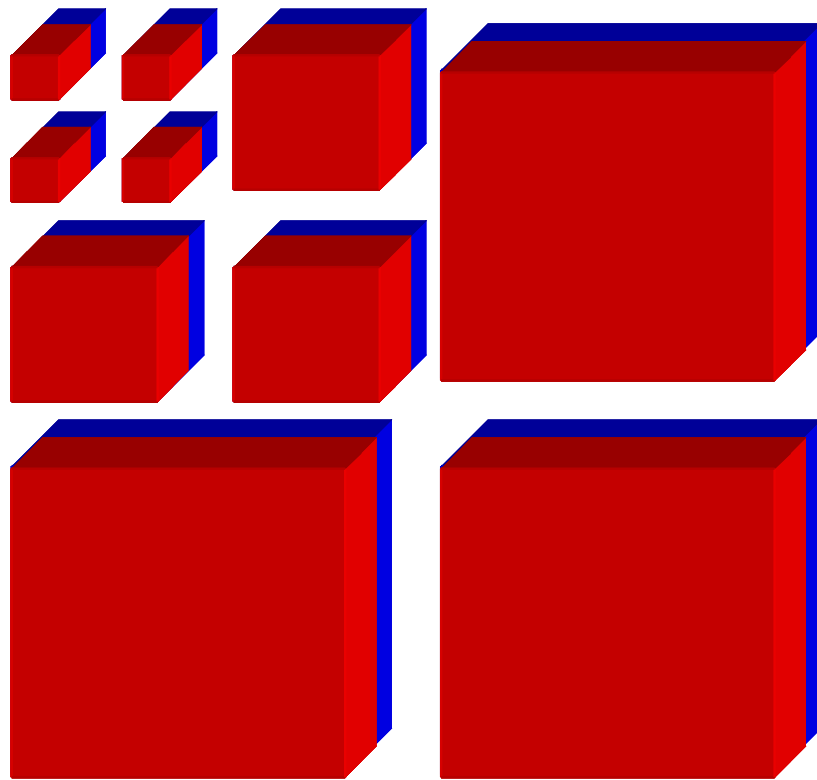
# Layer Organization: Example



# Layer (SNR) Progressive Example

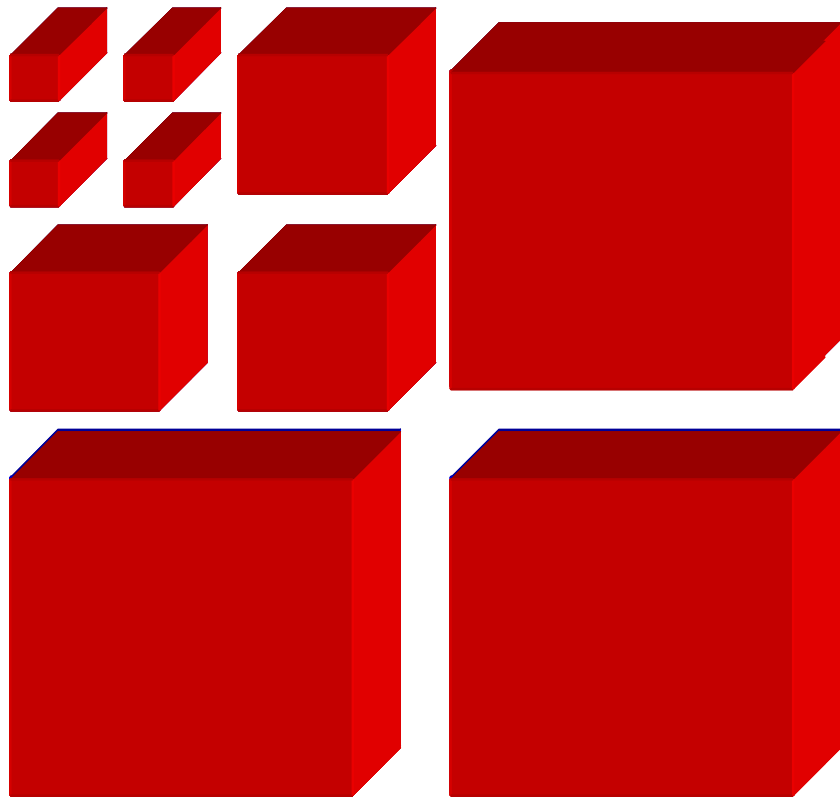


# Layer (SNR) Progressive Example





# Layer (SNR) Progressive Example

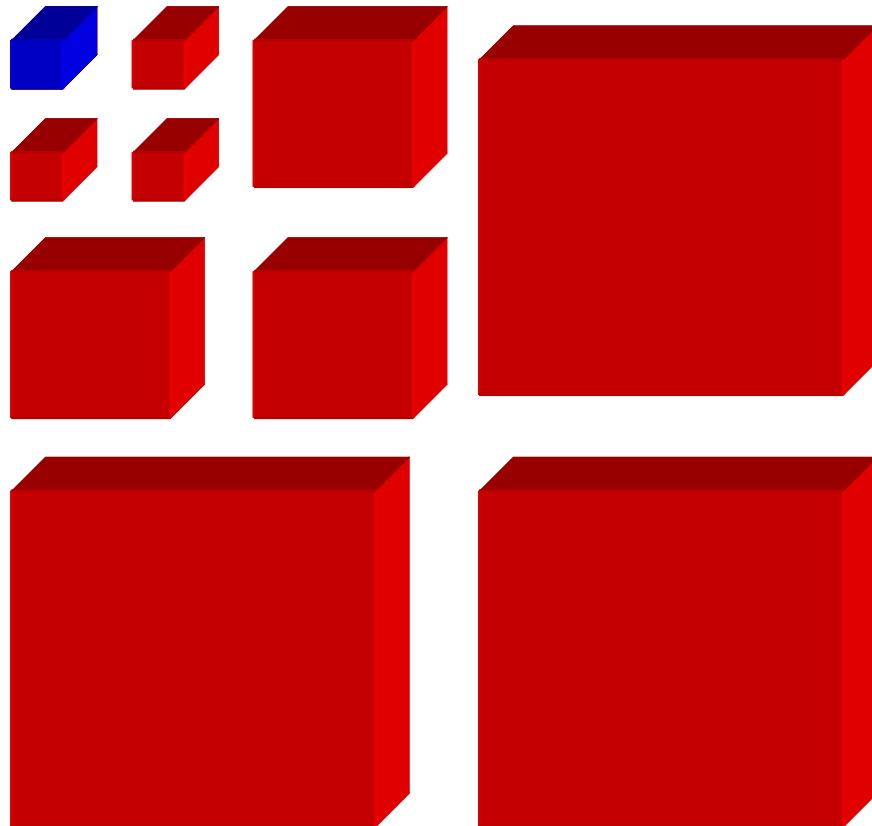


# Layer (SNR) Progressive Example

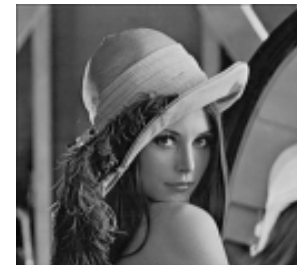
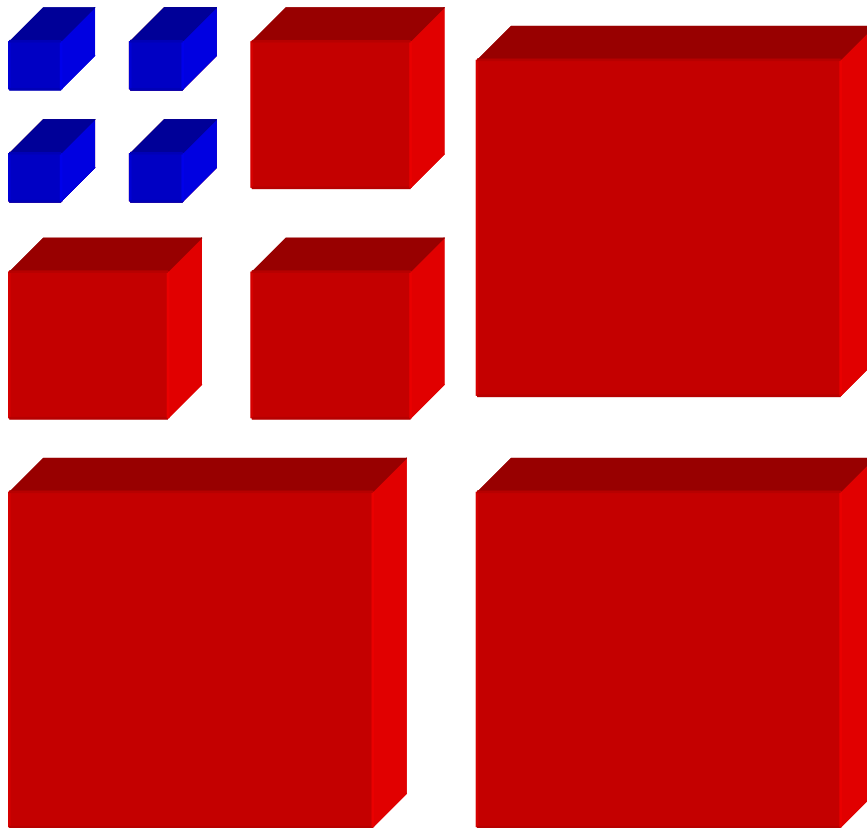




# Resolution Progressive Example

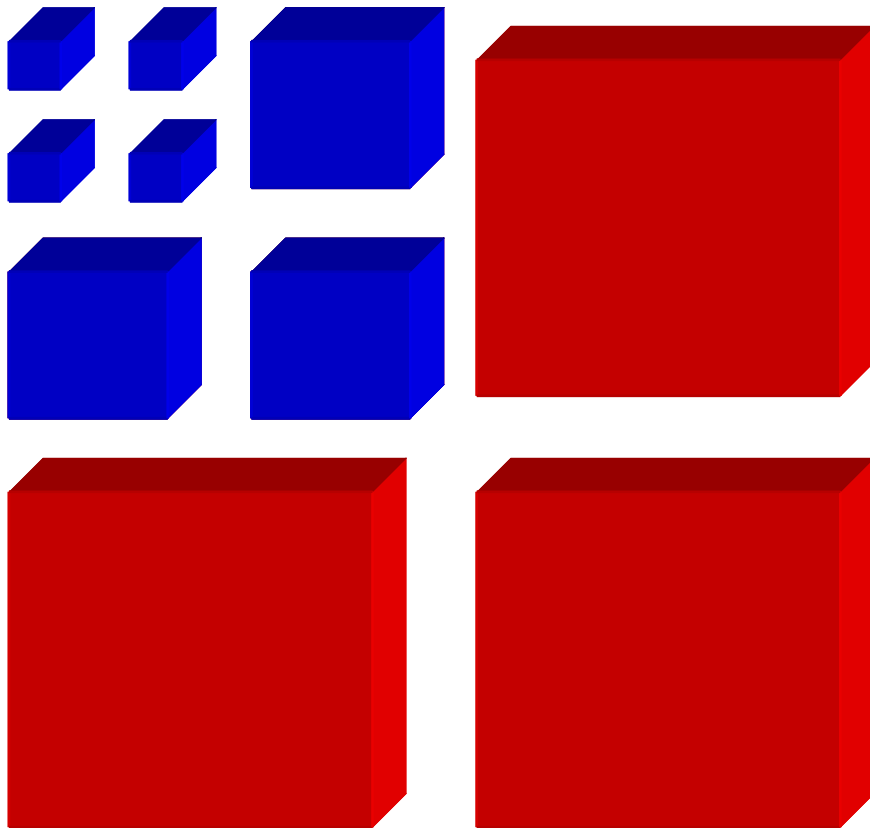


# Resolution Progressive Example

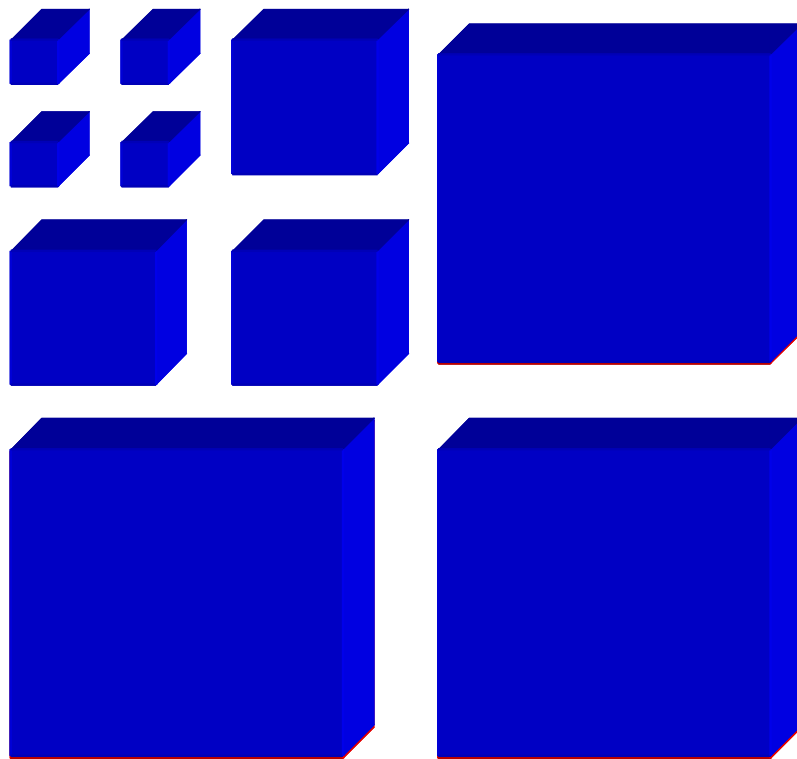




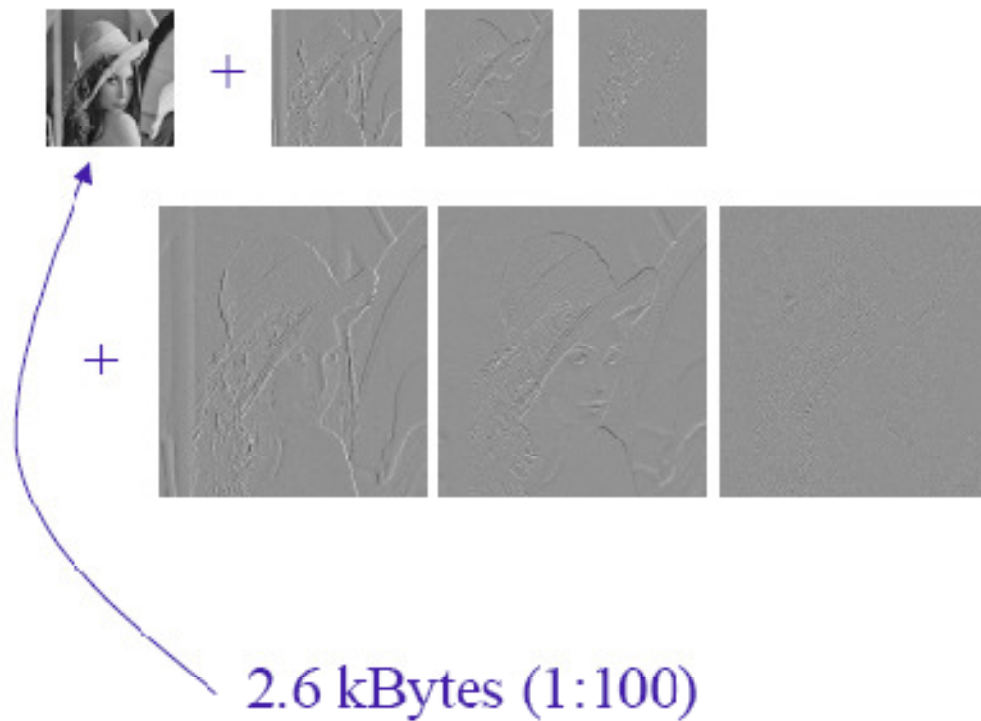
# Resolution Progressive Example



# Resolution Progressive Example



# Resolution Progressive Example

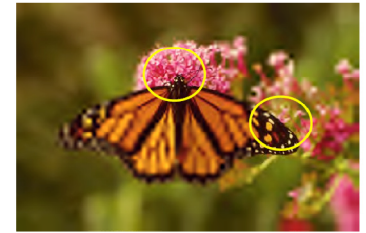


# JPEG 2000: Rate Allocation



- ★ **Rate allocation is the process allowing to target a specific compression ratio with the best possible quality (MSE, visual or other) for each layer and/or entire codestream. Possible types are:**
  - **None:** compression ratio is determined solely by the quantization step sizes and image content.
  - **Iterative:** quantization step sizes are adjusted according to obtained compression ratio and operation is repeated.
  - **Post-compression:** rate allocation is performed after the image data has been coded, in one step.
  - **Others (Lagrangian, scan-based, etc.)**
  
- ★ **Not standardized by JPEG 2000 → encoder choice.**

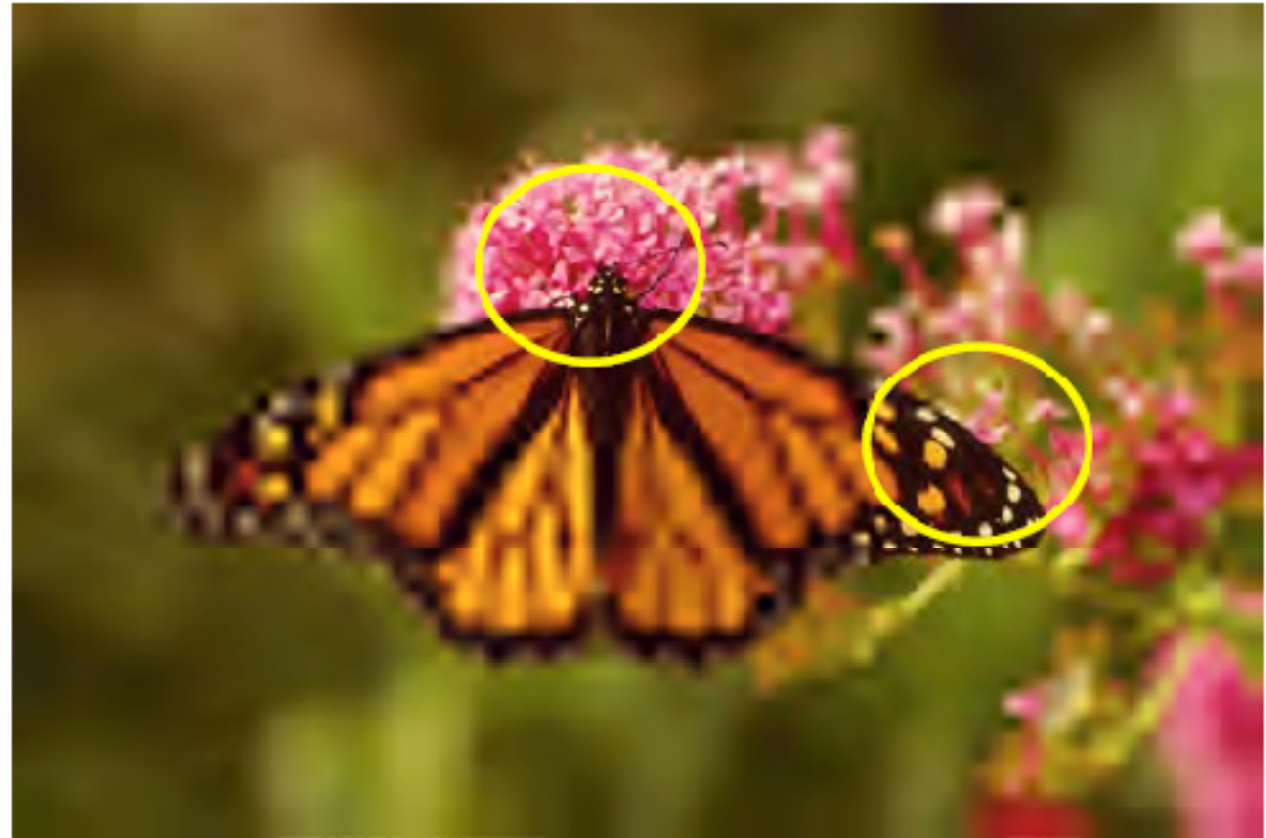
# Region of Interest Coding Principle



- ★ **Region of Interest (ROI) coding allows a non-uniform distribution of quality. The ROI is coded with a higher quality than the background (BG) region. A higher compression ratio can be achieved with the same or higher quality inside ROIs.**
- ★ **In JPEG 2000, only one ROI per image is supported; this ROI may be arbitrarily shaped and non-connected.**
- ★ **Static ROIs are defined at encoding time and are suitable for storage, fixed transmission, remote sensing, etc.**
- ★ **Dynamic ROIs are defined interactively by a user in a client/server situation during a progressive transmission; they are suitable for telemedicine, PDAs, mobile communications, etc. They can be achieved by the dynamic generation of layers matching the user's request.**



# Region of Interest Coding Example



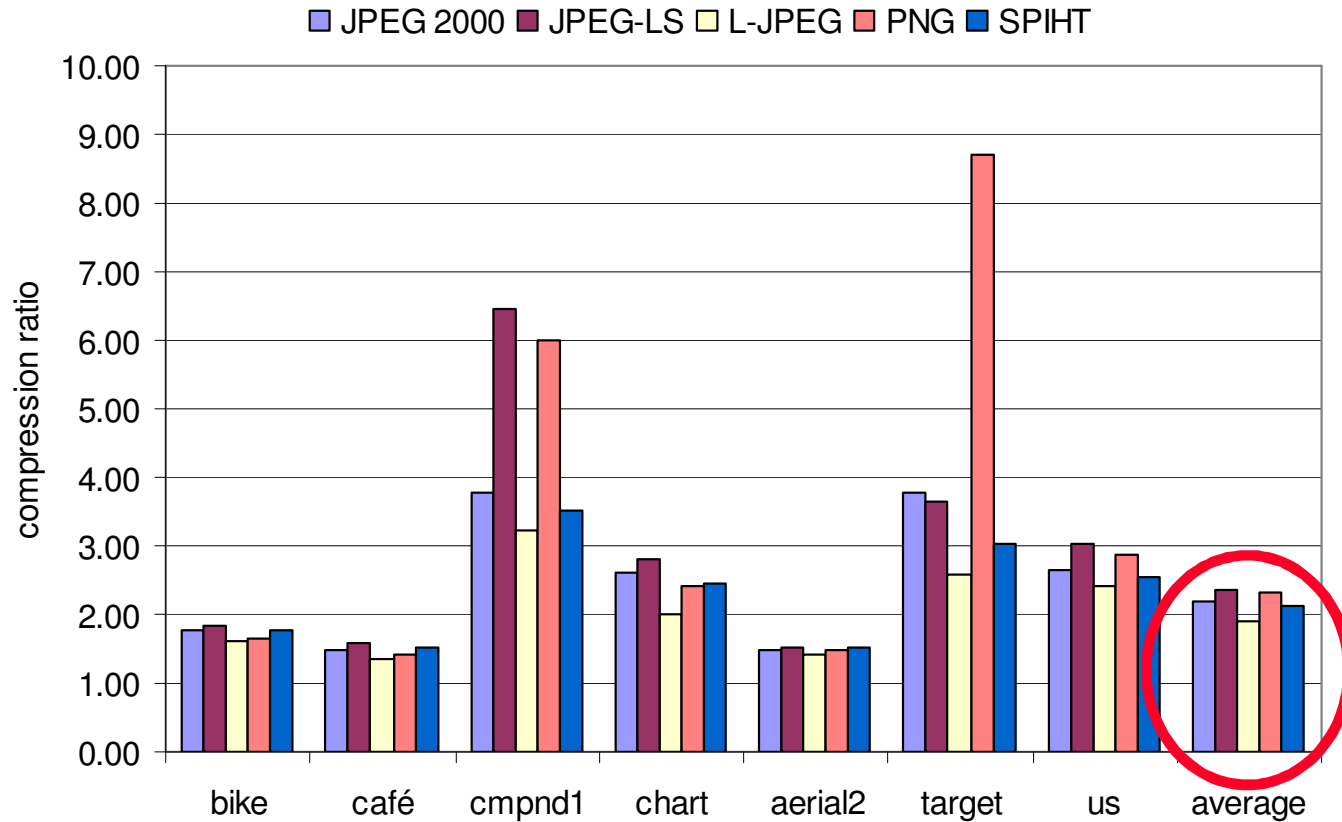


# **JPEG 2000 File Format: JP2**

**JP2 is the optional JPEG 2000 file format to encapsulate JPEG 2000 codestreams:**

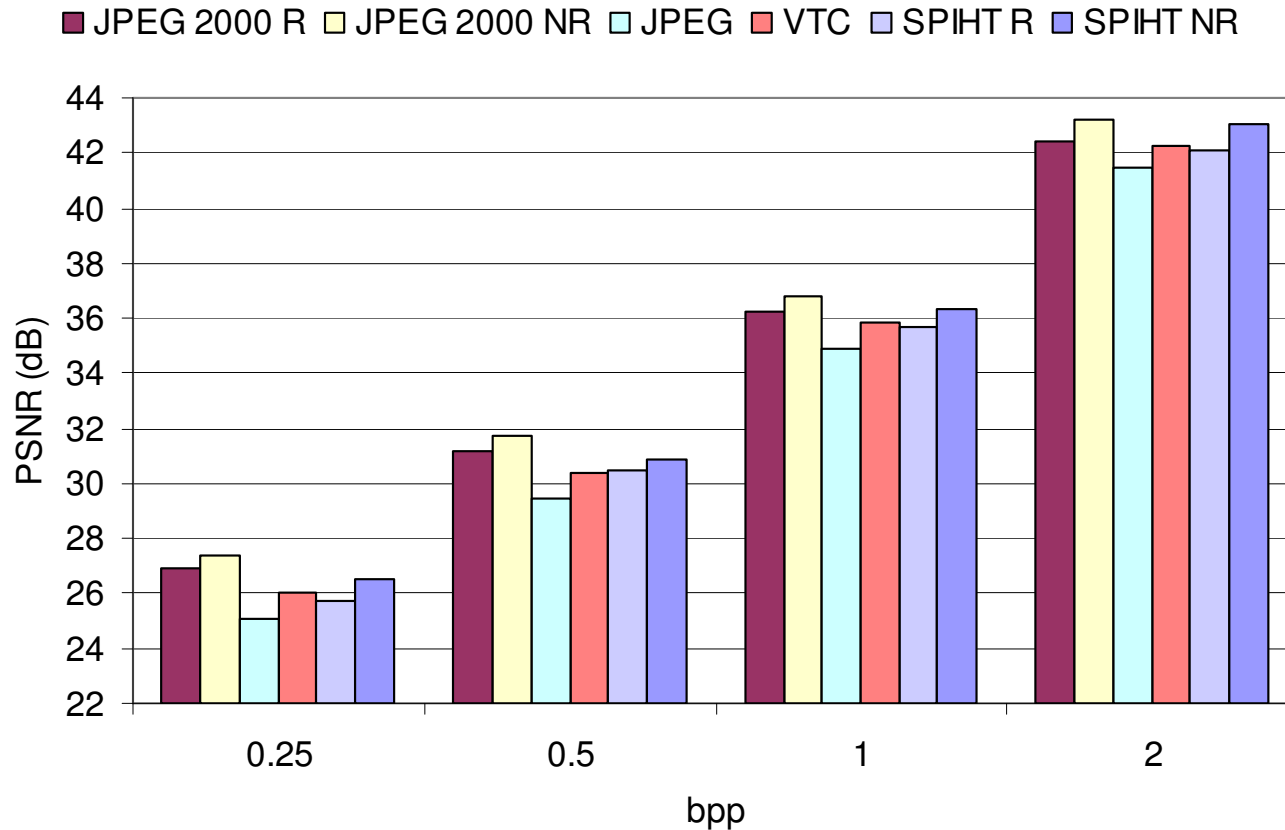
- **Extension: jp2**
- **Allows to embed XML information (e.g., metadata)**
- **Alpha channel (e.g., transparency)**
- **Accurate color interpretation**
- **“True color” and “palette color” supported**
- **Intellectual property information**
- **Capture and default display resolution**
- **File “magic number”**
- **File transfer errors (ASCII ftp, 7 bit e-mail, etc.)**

# Lossless Compression Performance



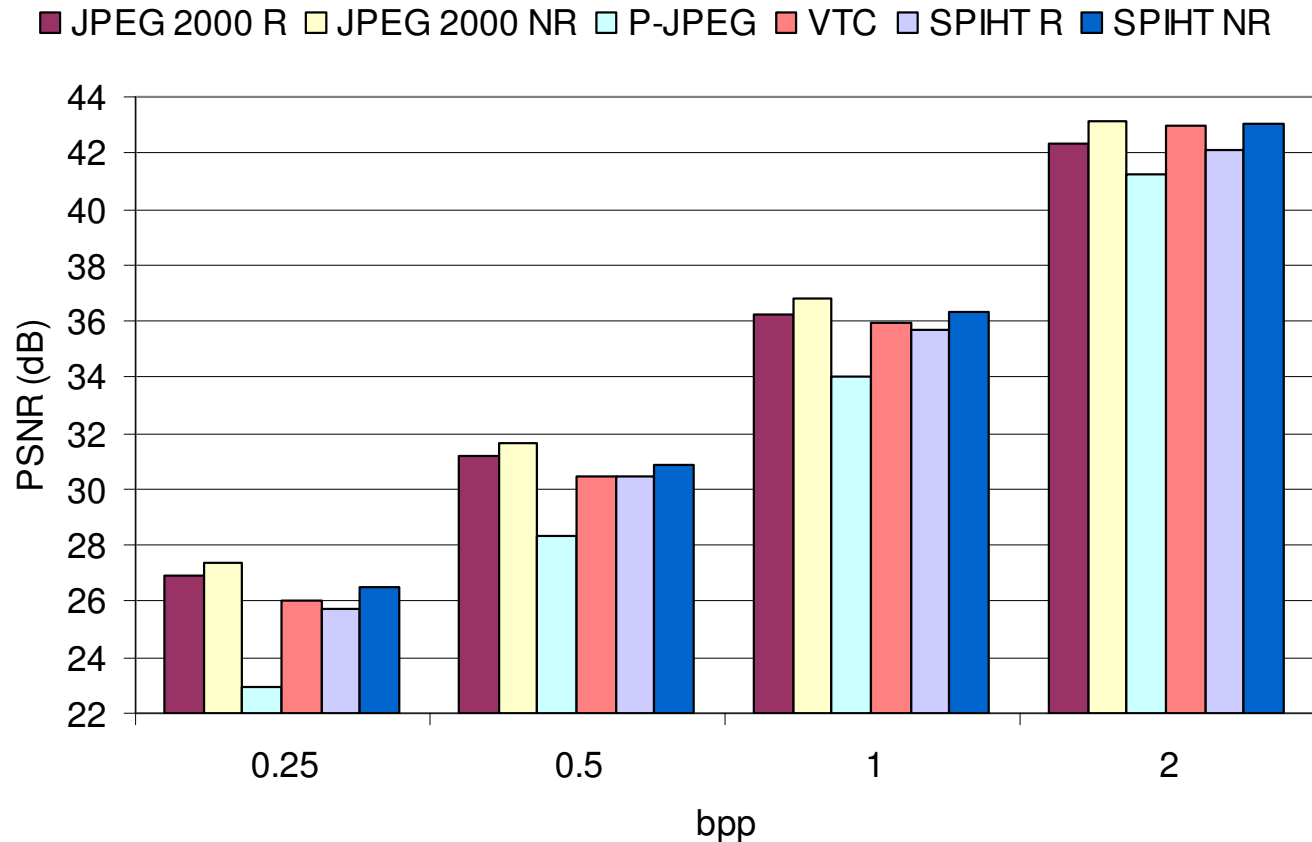
JPEG 2000: default options with (5,3) reversible filter; JPEG-LS: default options; JPEG lossless (L-JPEG): optimized Huffman tables and best predictor; PNG: maximum compression setting and best predictor; SPIHT: S+P filter with arithmetic coding.

# Non-Progressive Lossy Compression Performance



One bitstream generated for each bitrate. Average across all images. JPEG 2000: default options; JPEG: baseline with flat quantization tables and optimized Huffman tables; MPEG-4 VTC: single quantization; SPIHT: arithmetic coding.

# SNR Progressive Lossy Compression Performance



One bitstream generated at 2 bpp and decoded at 0.25, 0.5, 1 and 2 bpp. Average across all images. JPEG 2000: multiple layers; JPEG: progressive (successive refinement) and optimized Huffman tables; MPEG-4 VTC: multiple quantization; SPIHT: arithmetic coding.



# Digital Cinema: an Emerging Application Domain



- ★ **Digital Cinema Initiatives, LLC (DCI) was created in March, 2002, and is a joint venture of Disney, Fox, Paramount, Sony Pictures Entertainment, Universal and Warner Bros. Studios.**
- ★ **DCI's primary purpose is to establish and document voluntary specifications for an open architecture for digital cinema that ensures a uniform and high level of technical performance, reliability and quality control.**
- ★ **By establishing a common set of content requirements, distributors, studios, exhibitors, d-cinema manufacturers and vendors can be assured of interoperability and compatibility.**
- ★ **Because of the relationship of DCI to many of Hollywood's key studios, conformance to DCI's specifications is considered a requirement by software developers or equipment manufacturers targeting the digital cinema market.**



# DCI Adopts JPEG 2000 ...

## Image:

- ★ 2048×1080 (2K) at 24 fps or 48 fps, or 4096×2160 (4K) at 24 fps; 3×12 bits per pixel, XYZ color space (tristimulus values of a color, amounts of three primary colors in a three-component additive color model)
- ★ JPEG 2000 compression
- ★ From 0 to 5 or from 1 to 6 wavelet decomposition levels for 2K or 4K resolutions, respectively
- ★ Compression rate of 4.71 bits/pixel (2K@24 fps), 2.35 bits/pixel (2K@48 fps), 1.17 bits/pixel (4K@24 fps)
- ★ 250 Mbit/s maximum image bit rate



## Audio:

- ★ 24 bits per sample, 48 kHz or 96 kHz uncompressed PCM
- ★ Up to 16 channels



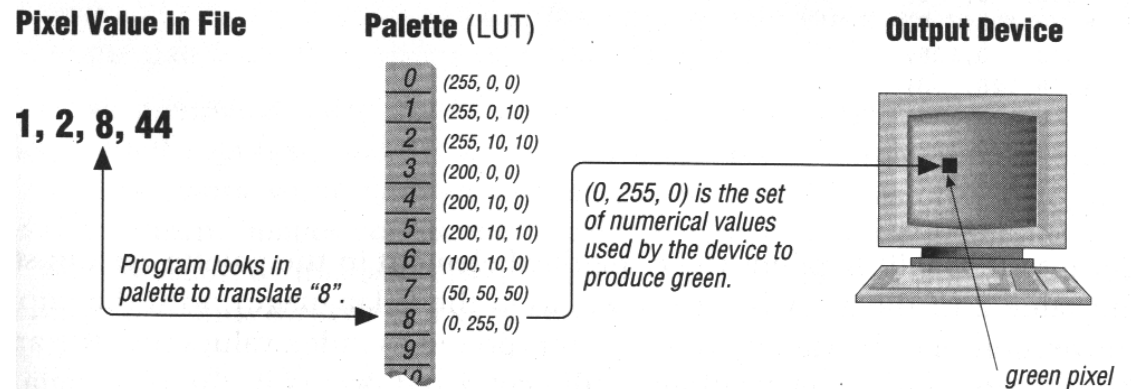
# Other Image Compression Formats

# Other Formats: Bitmap (BMP)

- ★ BMP format usually includes a header, the image data and additional information, e.g. about the colour *palette*.
- ★ The image data may correspond to PCM samples or to indices in a colour *palette*.
- ★ The image data may be structured in several ways, e.g. by samples, by components, etc.

**Plus: easy to use, to access a certain position, to change a pixel**

**Minus: low efficiency (no compression)**





## **Other Formats: Graphics Interchange Format (GIF)**

- ★ Allows to store several BMP images in a single file but always in RGB**
- ★ Image data always coded with the Lempel-Ziv-Welch (LZW) algorithm; 40% or more compression for 8 bit/sample images**
- ★ Image data structured as a sequence of packets**
- ★ Maximum image size: 64K × 64K**
- ★ Number of bit/sample: 1 to 8**



## Other Formats: Portable Network Graphics (PNG)

- ★ PNG is a bitmapped image format that employs lossless data compression.
- ★ PNG uses a lossless data compression method known as DEFLATE, which is the same algorithm used in the *zlib* compression library.
- ★ This method is combined with prediction, where for each image line, a filter method is chosen that predicts the color of each pixel based on the colors of previous pixels and subtracts the predicted color of the pixel from the actual color.
- ★ An image line filtered in this way is often more compressible than the raw image line would be, especially if it is similar to the line above.



# **Other Formats: Portable Network Graphics (PNG) versus Other Formats**

- ★ **PNG was created to improve upon and replace the GIF format, as an image-file format not requiring a patent license.**
- ★ **On most images, PNG can achieve greater compression than GIF.**
- ★ **JPEG can produce a smaller file than PNG for photographic images, since JPEG uses a lossy encoding method.**
- ★ **Using PNG for such images would result in a large increase in file size (often 5–10 times) with negligible gain in quality. PNG is a better choice than JPEG for storing images that contain text, line art, or other images with sharp transitions.**
- ★ **JPEG is a poor choice for storing images that require further editing as it suffers from generation loss, whereas lossless formats do not.**



## **Other Formats: Tag Image File Format (TIFF)**

- ★ **Allows to store several BMP images in a single file**
- ★ **Image data may be coded or not; allowed coding algorithms are LZW, RLE, Group 3 Fax, Group 4 Fax, JPEG**
- ★ **Maximum image size:  $2^{32} - 1$  pixels**
- ★ **Number of bit/sample: 1 a 24**
- ★ **Plus: Very flexible and varied**



# Image Compression Solutions Functional Comparison

	JPEG 2000	JPEG-LS	JPEG	MPEG-4 VTC	PNG
lossless compression performance	+++	++++	(+)	-	+++
lossy compression performance	+++++	+	+++	++++	-
progressive bitstreams	+++++	-	++	+++	+
Region of Interest (ROI) coding	+++	-	-	(+)	-
arbitrary shaped objects	-	-	-	++	-
random access	++	-	-	-	-
low complexity	++	+++++	+++++	+	+++
error resilience	+++	++	++	? (++++)	+
non-iterative rate control	+++	-	-	+	-
genericity	+++	+++	++	++	+++

+ : supported, the more marks the better

- : not supported

( ) : separate mode required

# What Makes a Compression Technology Successful ?

- ★ Adoption in a standard
- ★ Compression performance
- ★ Encoder and decoder complexity
- ★ Error resilience
- ★ Random access
- ★ Scalability
- ★ Added value regarding alternative solutions/standards
- ★ Patents and licensing issues
- ★ Adoption companies
- ★ ...





# Bibliography

- ★ **JPEG: Still Image Data Compression Standard**, William Pennebaker, Joan Mitchell, Kluwer Academic Publishers, 1993
- ★ **Image and Video Compression Standards: Algorithms and Architectures**, Vasudev Bhaskaran and Konstantinos Konstantinides, Kluwer Academic Publishers, 1995
- ★ **Digital Image Compression Techniques**, Majid Rabbani, Paul W. Jones, SPIE Press, Tutorial texts on Optical Engineering, 1991
- ★ **JPEG2000: Image Compression Fundamentals, Standards and Practice**, D.S. Taubman and M.W. Marcellin, Kluwer Academic Publishers, 2002