# DIGITAL IMAGE COMPRESSION


*Fernando Pereira*

*Instituto Superior Técnico*




# Multilevel Photographic Image Coding
## (gray and colour)

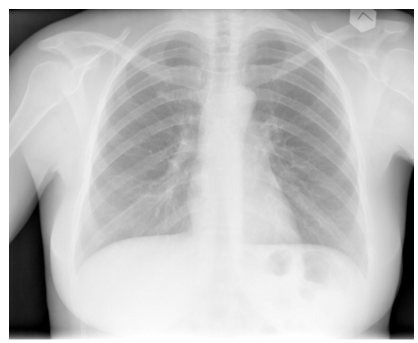
**OBJECTIVE**

**Efficient representation of multilevel photographic images (still pictures) for storage and transmission.**

# 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×N luminance and chrominance samples with a certain number of bits per sample.**

**Thus the total number of bits**

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

# Image (Source) Coding Objective

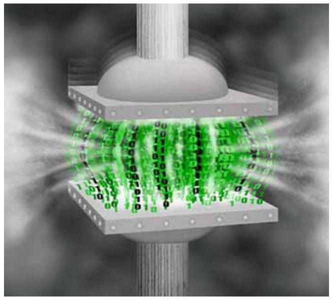


**Image coding/compression deals with the efficient representation of images, satisfying the relevant requirements.**

**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
- Fractal-based coding
- Vector quantization coding
- Wavelet-based coding, e.g. JPEG 2000 standard
- 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.

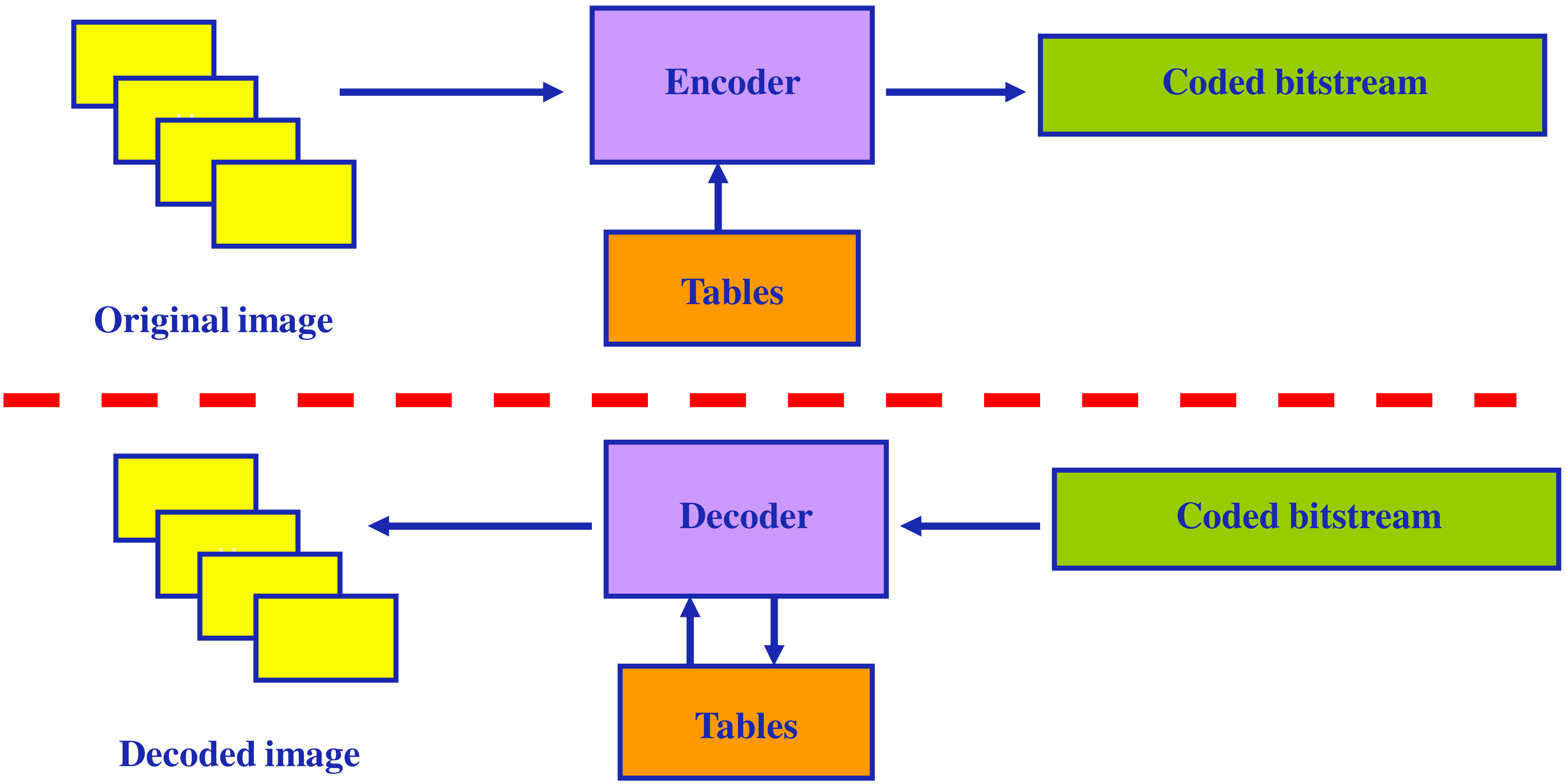
# JPEG Standard Major Requirements

- ✶ **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.

# Objective

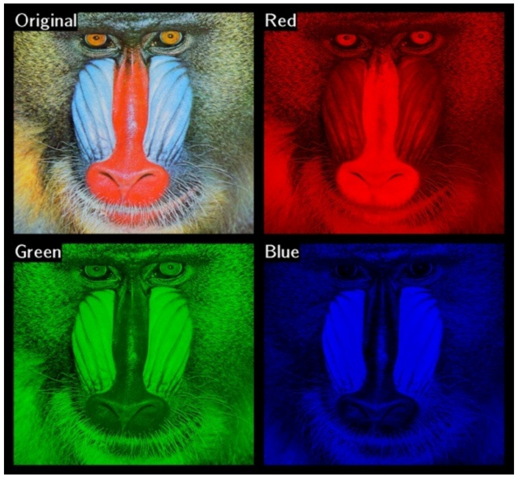
**Definition of a generic compression standard for multilevel photographic images considering the requirements of most applications using.**

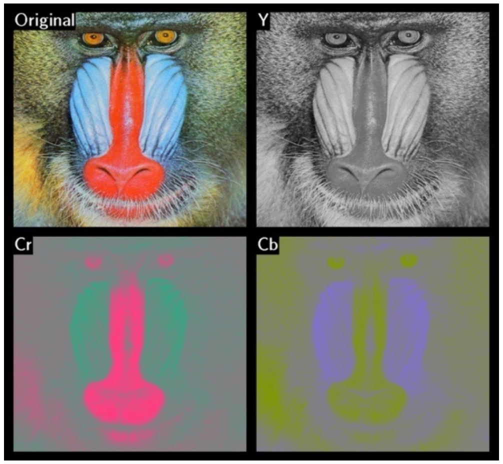
# JPEG Elements



# What Images can JPEG Encode ?

- ✶ Size between 1×1 and 65535×65535
- ✶ 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



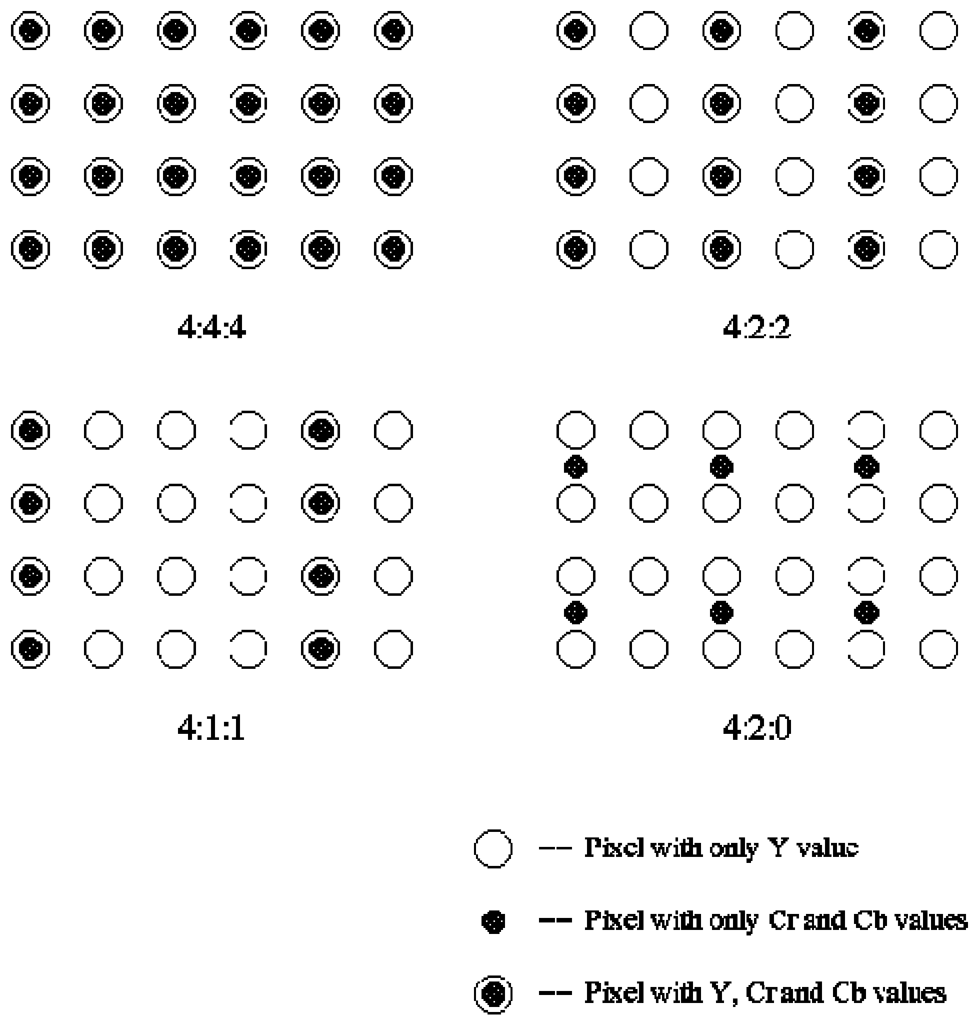



# ITU-R 601 Recommendation: a Typical Resolution

- **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 | ResolutionU/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



# Types of JPEG 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).

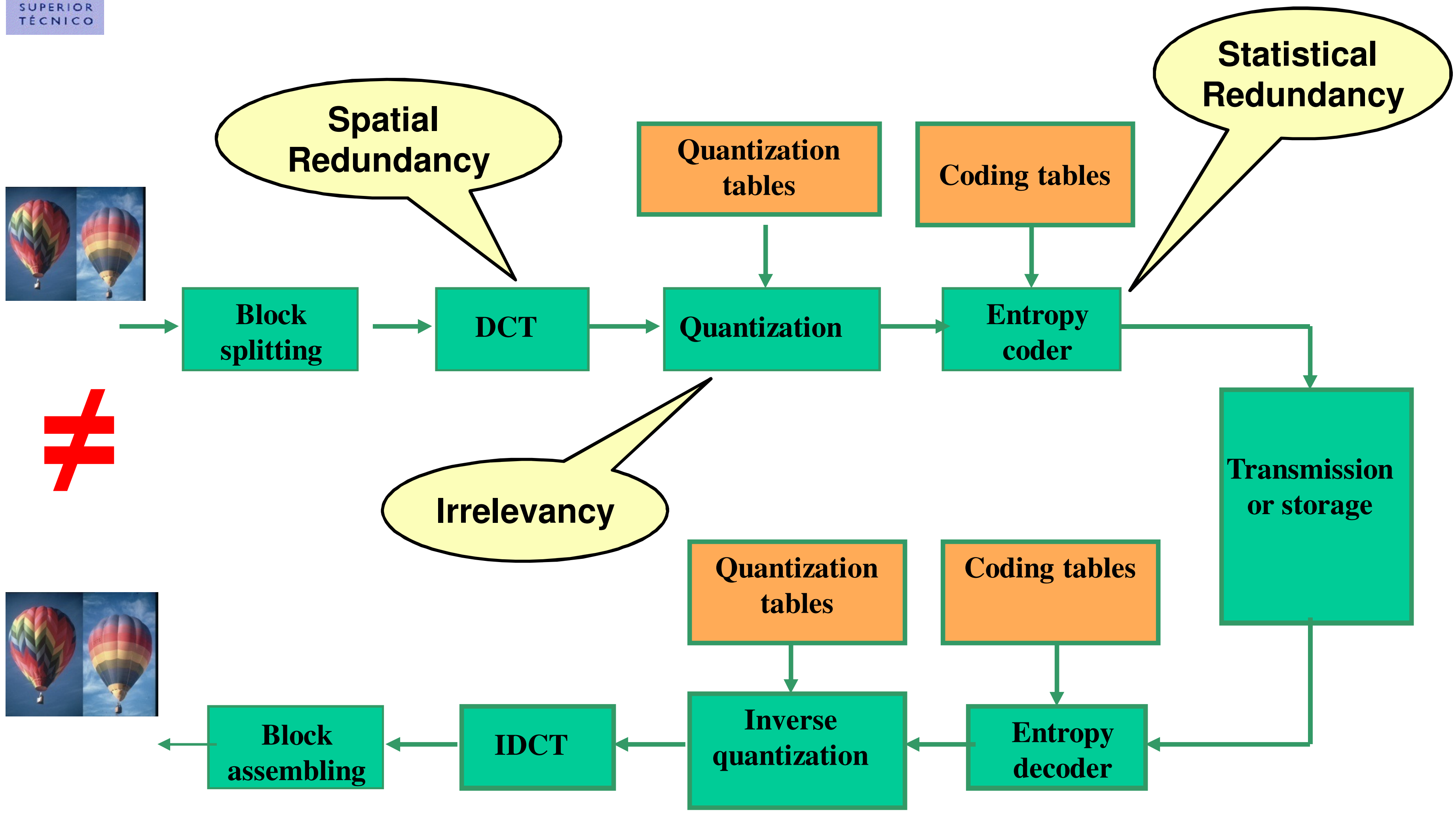
# JPEG Baseline Process

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

**BASELINE SEQUENTIAL PROCESS**

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

# DCT Based Image Coding



# Transform Coding

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

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

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

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

Transform coefficients

Image block

*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*=In where In is the identiy 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×n matrix if the original is a n×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 incorrelated 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 efficency.**
- **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**

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

Weights

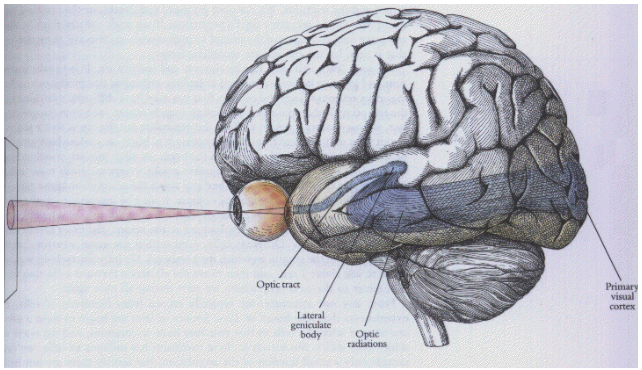
Basic image blocks

**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 meanning 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.



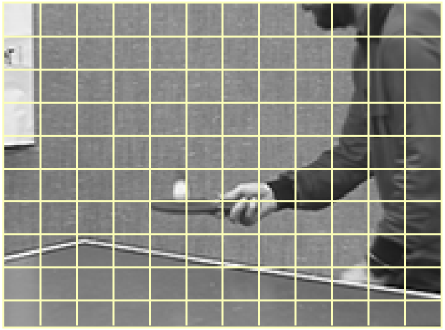

- ✶ The human visual system is less sensitive to the high spatial frequencies

->> 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 ?



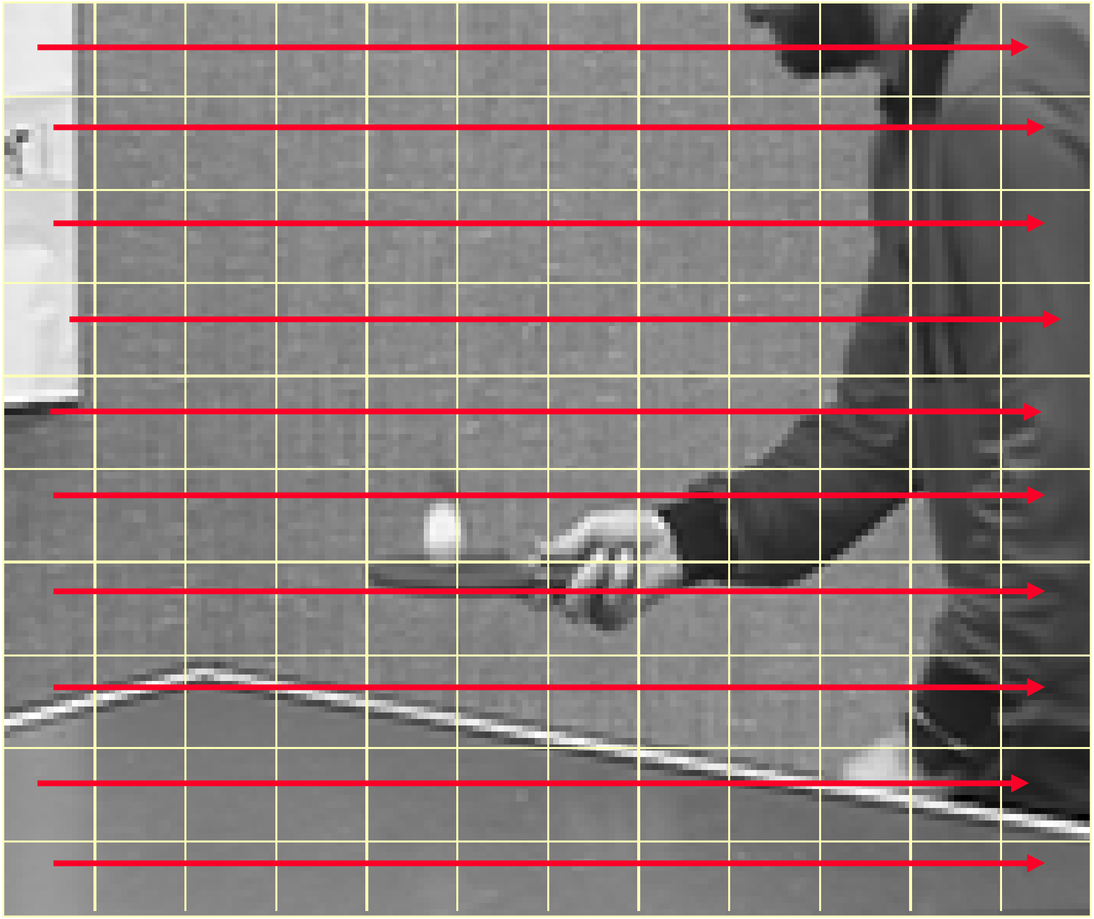
**Basically, the transform represents the original signal in another domain where it can be more efficiently coded by exploiting the spatial redundancy.**

- **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.**

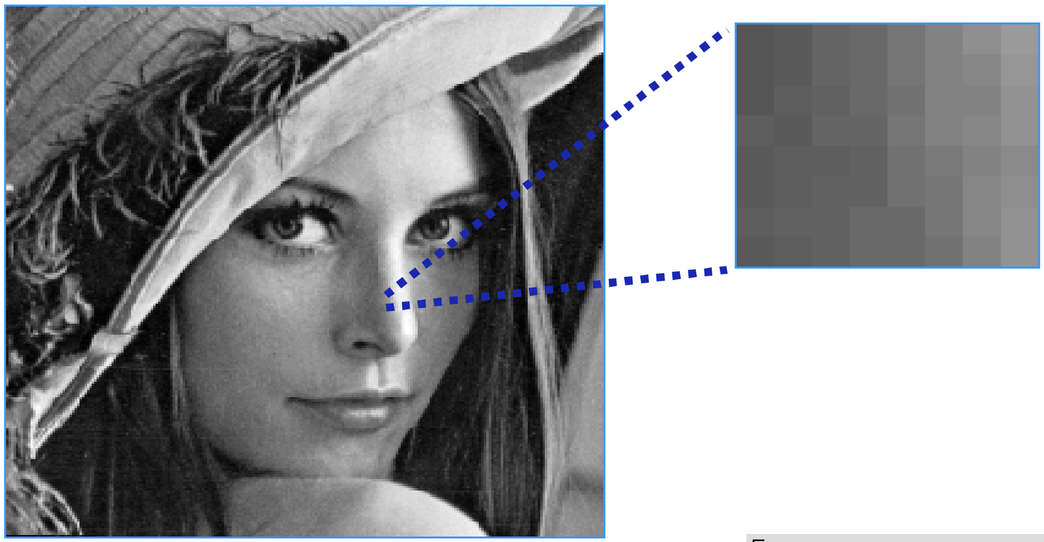
- **However, the computational effort associated to the transform grows quickly with the size of the block used … and the added spatial redundancy decreases …**

**Applying the transform to blocks, typically of 8×8 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 =

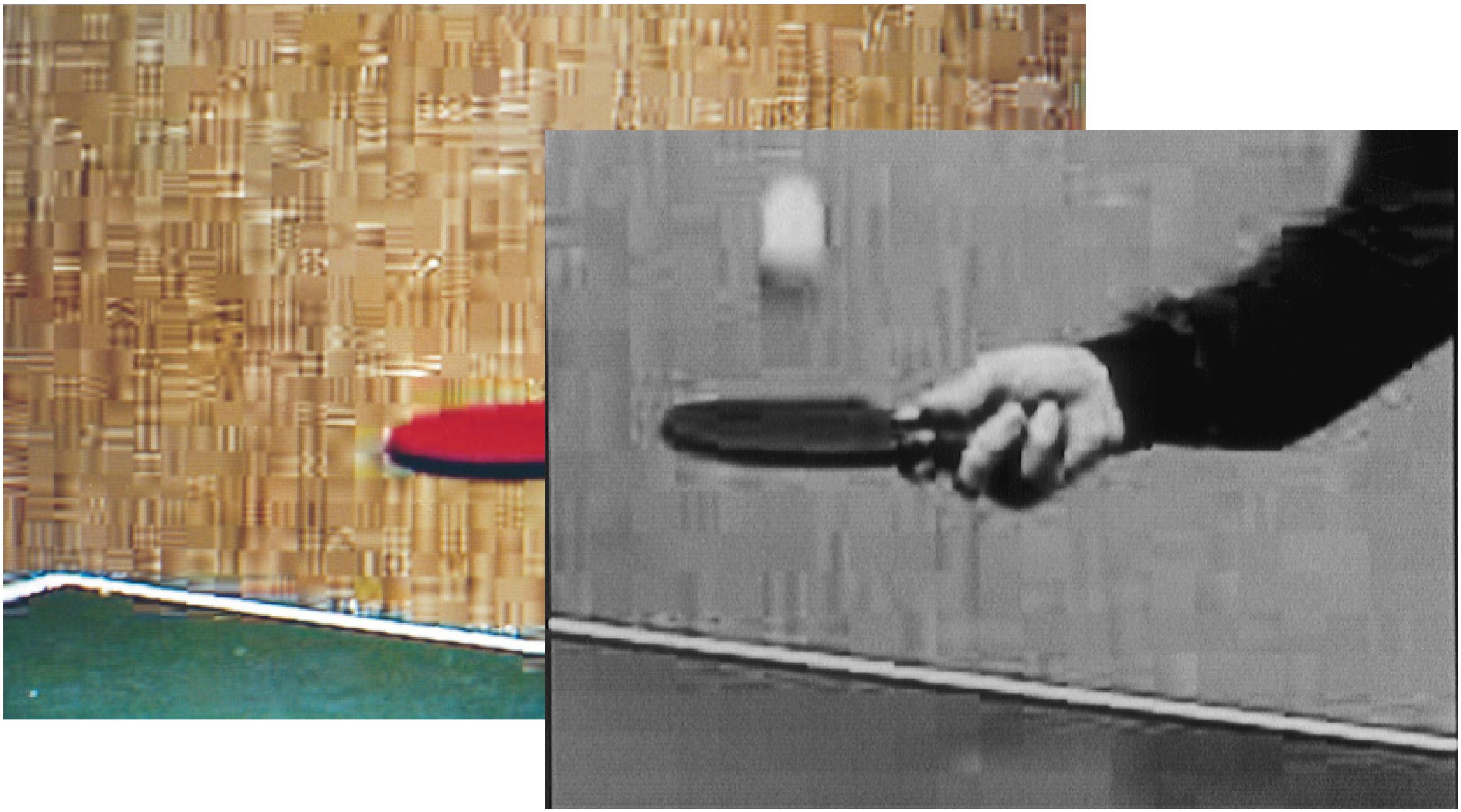
$$
\begin{bmatrix}
87 & 89 & 101 & 106 & 118 & 130 & 142 & 155 \\
85 & 91 & 101 & 105 & 116 & 129 & 135 & 149 \\
86 & 92 & 96 & 105 & 112 & 128 & 131 & 144 \\
92 & 88 & 102 & 101 & 116 & 129 & 135 & 147 \\
88 & 94 & 94 & 98 & 113 & 122 & 130 & 139 \\
88 & 95 & 98 & 97 & 113 & 119 & 133 & 141 \\
92 &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

↓ Transform ↓

Transform Coefficients =

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

# The Block Effect …



# Karhunen-Loéve Transform (KLT)

**The Karhunen-Loéve 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 KLT.

# Discrete Cosine Transform (DCT)

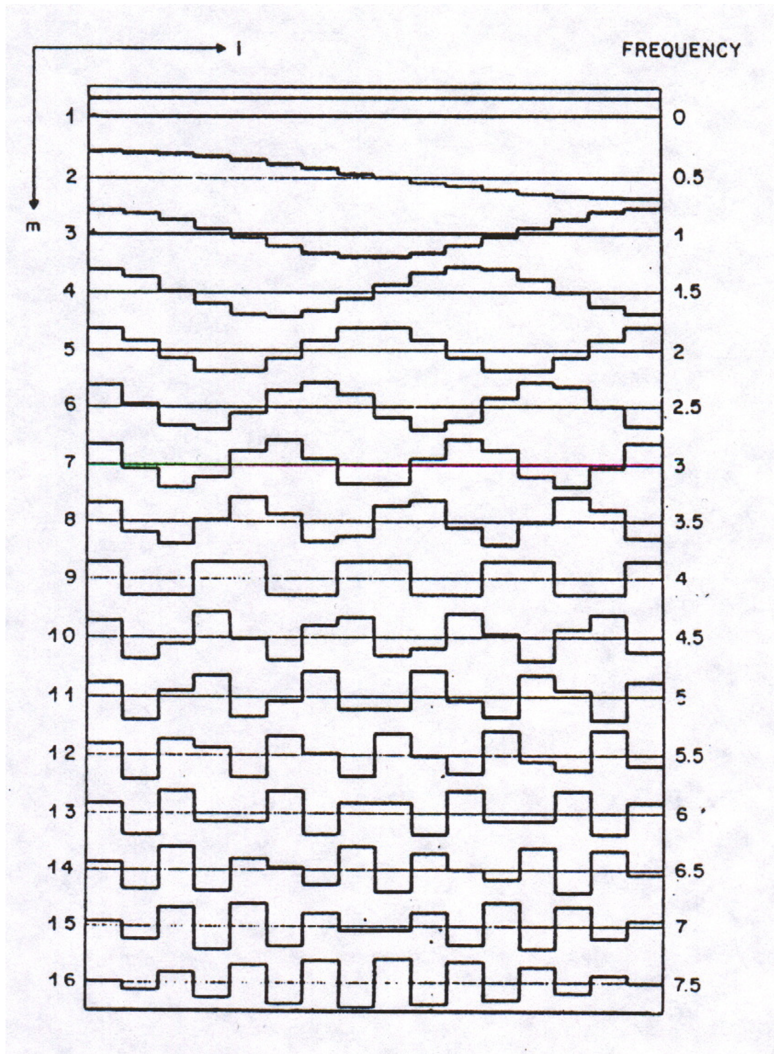
**The DCT is one of the several sinusoidal transforms available; its basis functions correspond to discretized sinusoisal functions.**

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

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

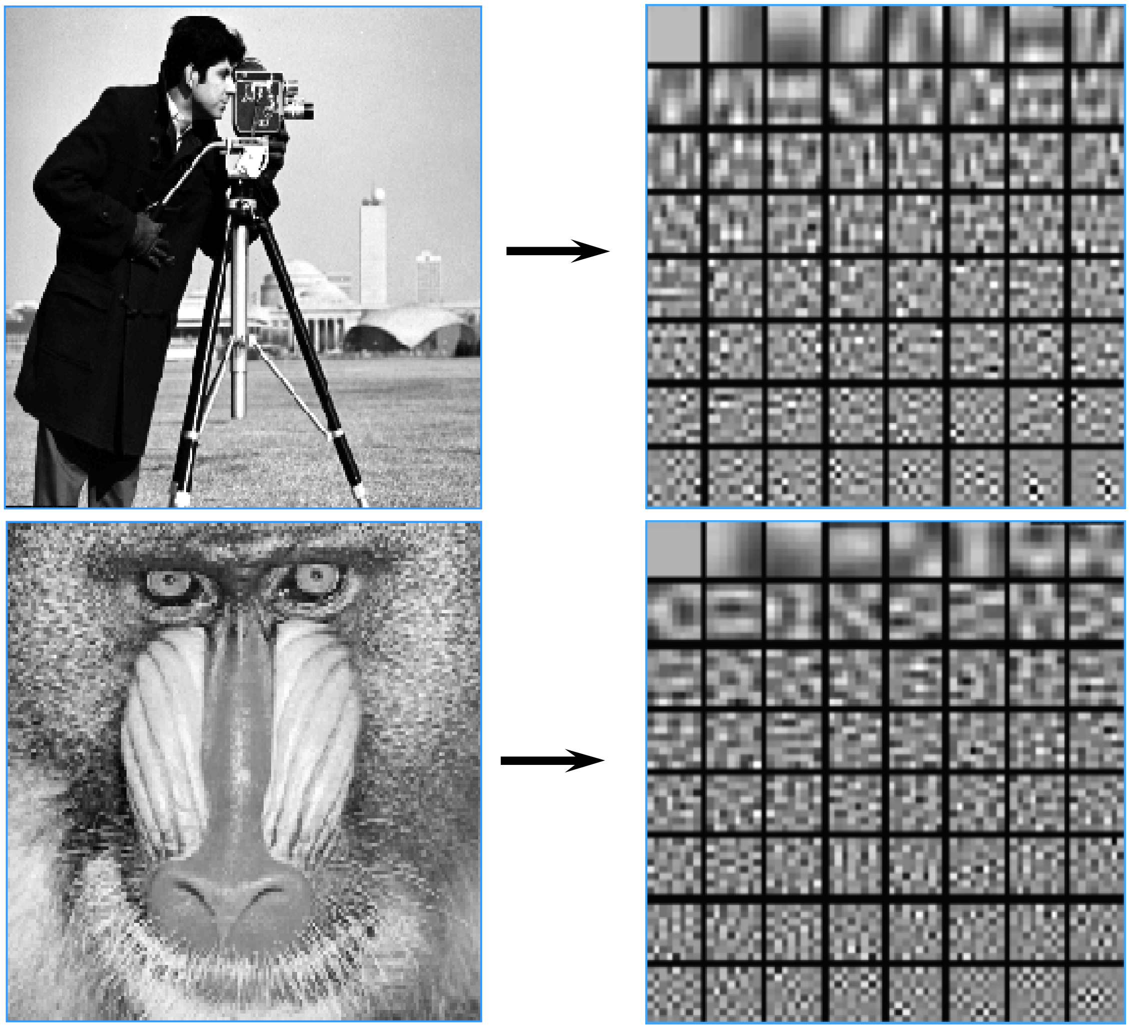
**The DCT is the most used transform for image and video compression since its performance is close to the KLT performance for highly correlated signals; moreover, there are fast implementation algorithms available.**

# DCT Unidimensional Basis Functions (N=8)

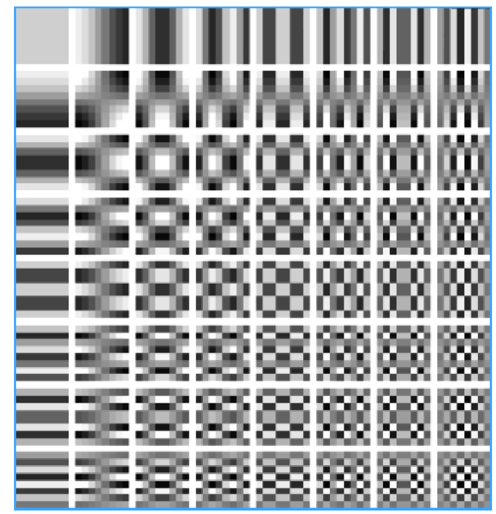


# DCT Bidimensional Basis Functions (N=8)

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



KLT



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**

↓

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 |

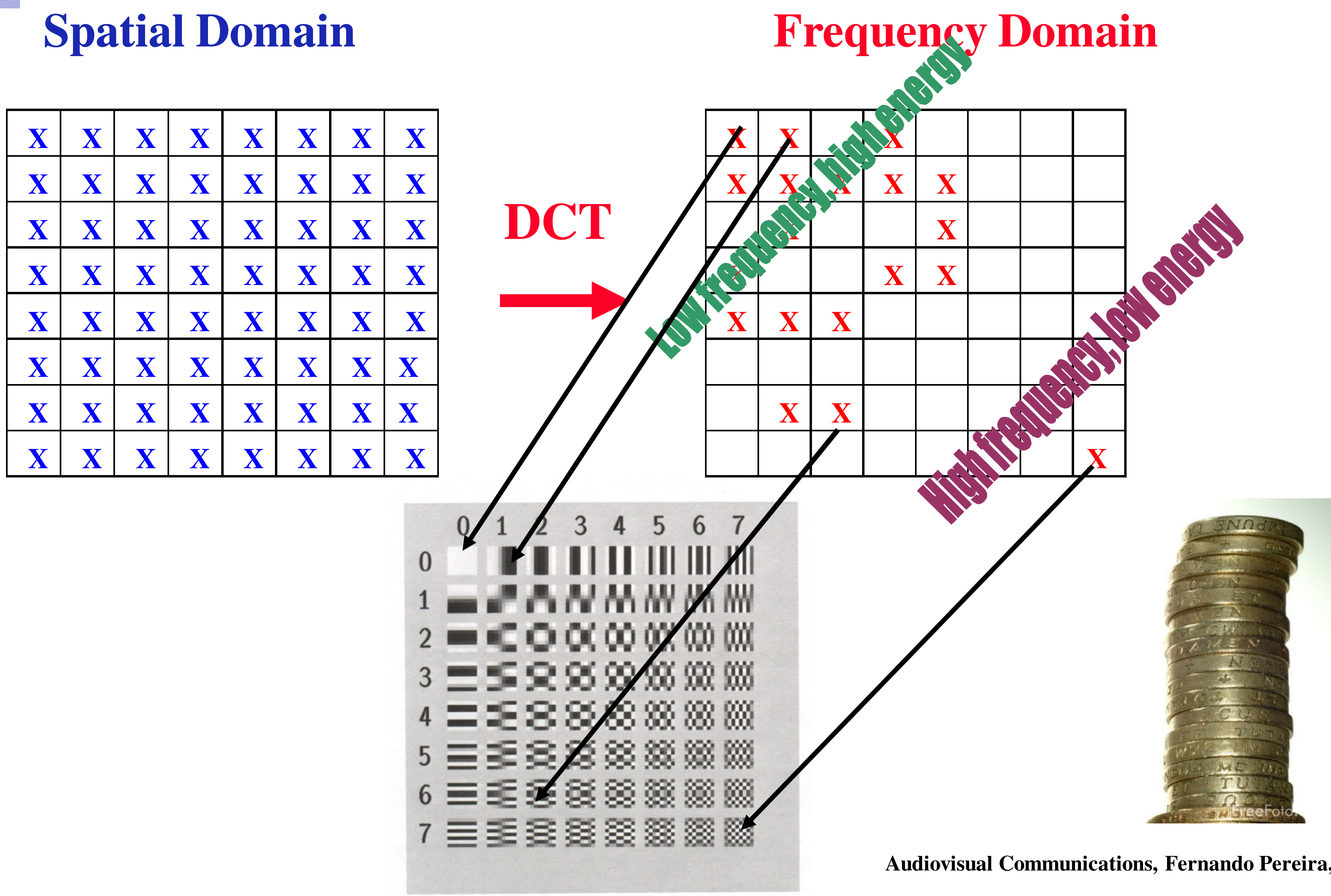
# DCT in JPEG

**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).**

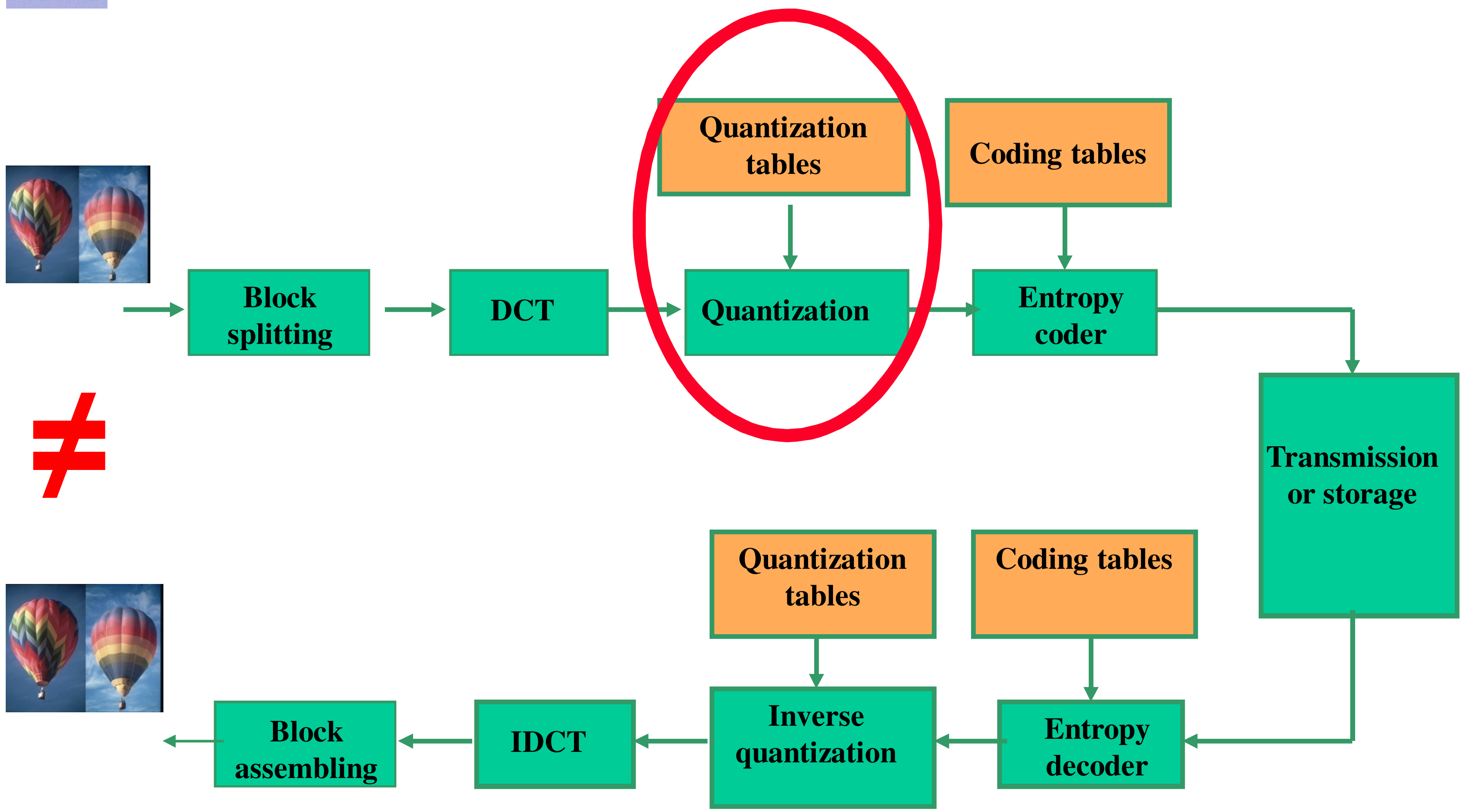
- ✶ **In order to accomodate future implementation developments, the JPEG recommendation does not specify any specific DCT or IDCT implementation.**
- ✶ **The JPEG recommendation specifies a fidelity/accuracy test in order to limit the differences caused by the freedom in terms of DCT and IDCT implementation.**

***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 ?



# DCT Based Image Coding



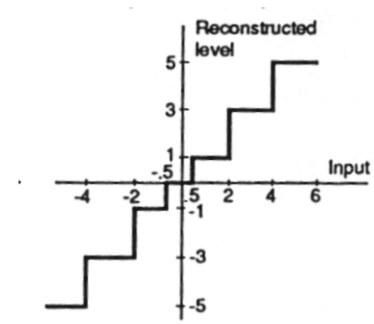
# Quantization

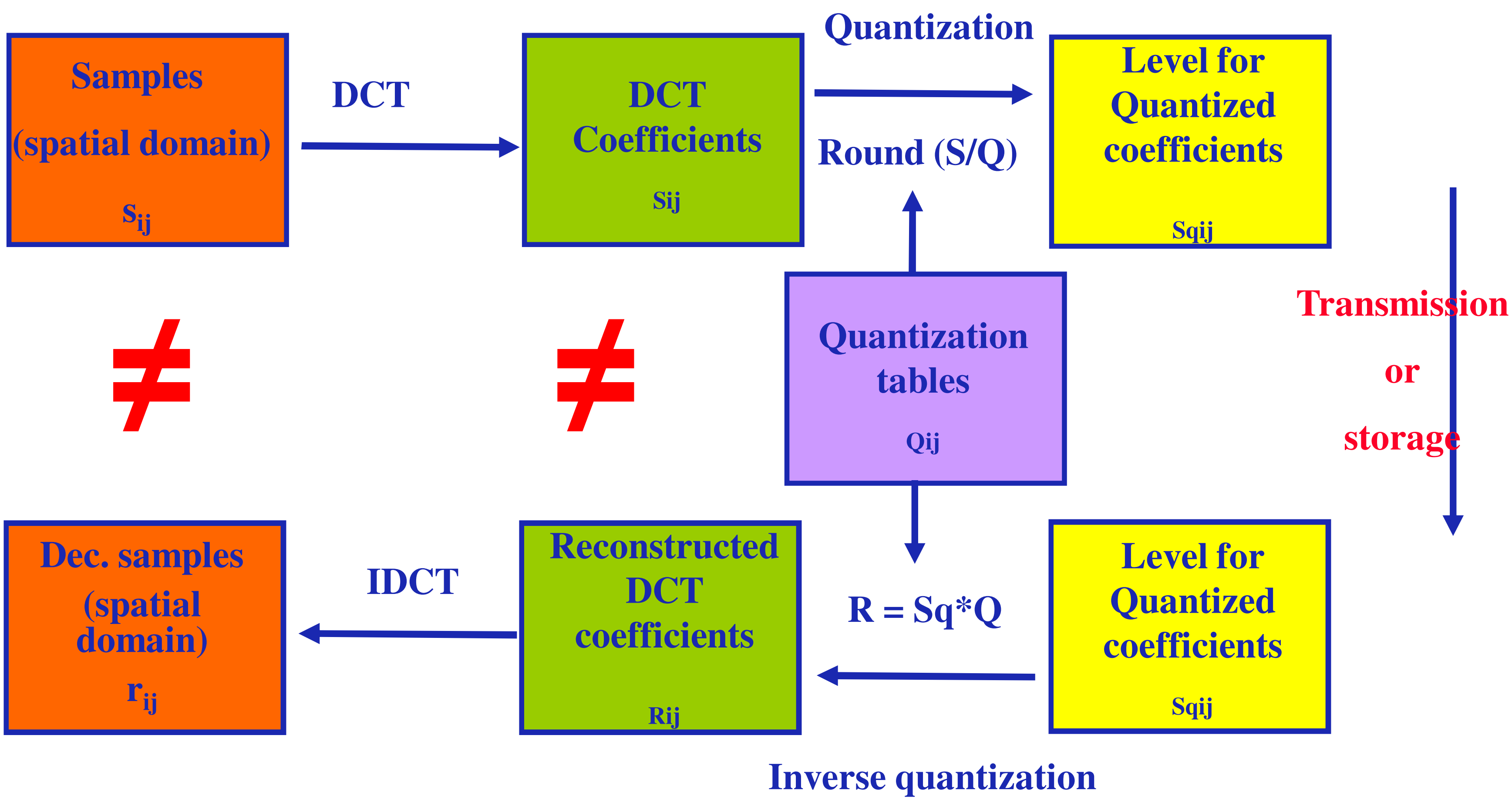
Quantization is the process by which irrelevancy or perceptual redundancy is reduced. This process is the major responsible for the quality losses in DCT based codecs (which may be transparent ;-).

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).

# How Does DCT Coding Work ?



# 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.

**Luminance**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 16 | 11 | 10 | 16 | 24 | 40 | 51 | 61 |
| 12 | 12 | 14 | 19 | 26 | 58 | [illegible] | [illegible] |
| 14 | 13 | 16 | [illegible] | [illegible] | [illegible] | 69 | [illegible] |
| [illegible] | 17 | [illegible] | [illegible] | [illegible] | [illegible] | 80 | 62 |
| [illegible] | [illegible] | [illegible] | [illegible] | 68 | 109 | 103 | 77 |
| [illegible] | [illegible] | 55 | 64 | 81 | 104 | 113 | 92 |
| 49 | 64 | 78 | 87 | 103 | 121 | 120 | 101 |
| 72 | 92 | 95 | 98 | 112 | 100 | 103 | 99 |

**Chrominances**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 17 | 18 | 24 | 47 | 99 | 99 | 99 | 99 |
| 18 | 21 | 26 | 66 | 99 | 99 | 99 | 99 |
| 24 | 26 | 56 | 99 | 99 | [illegible] | [illegible] | 99 |
| 47 | 66 | 99 | [illegible] | [illegible] | [illegible] | [illegible] | 99 |
| [illegible] | [illegible] | [illegible] | [illegible] | [illegible] | 99 | 99 | 99 |
| [illegible] | [illegible] | [illegible] | 99 | 99 | 99 | 99 | 99 |
| 99 | 99 | 99 | 99 | 99 | 99 | 99 | 99 |
| 99 | 99 | 99 | 99 | 99 | 99 | 99 | 99 |

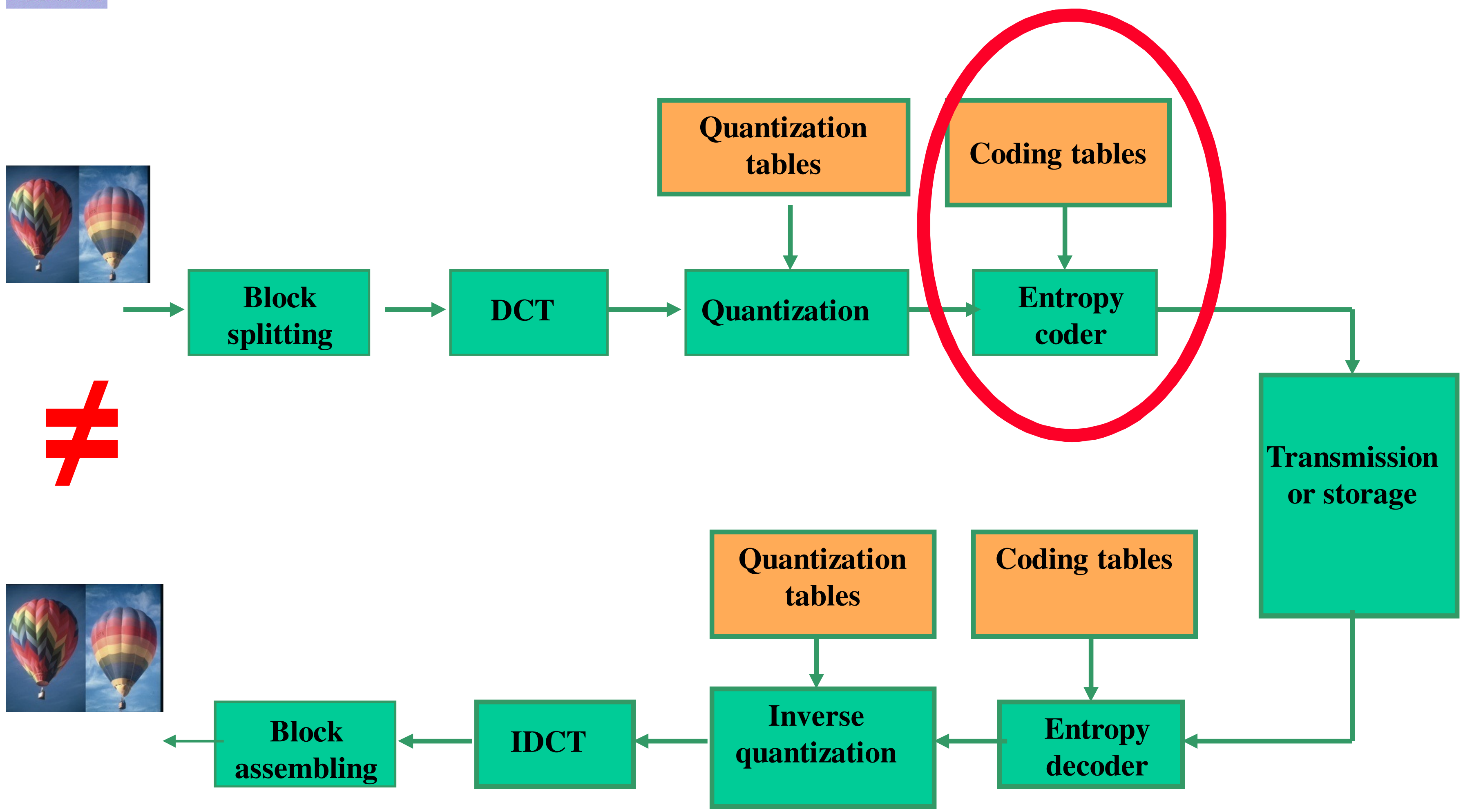
**Situation: Luminance and crominance with 2:1 horizontal subsampling; samples with 8 bits (*Lohscheller*)**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 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 |

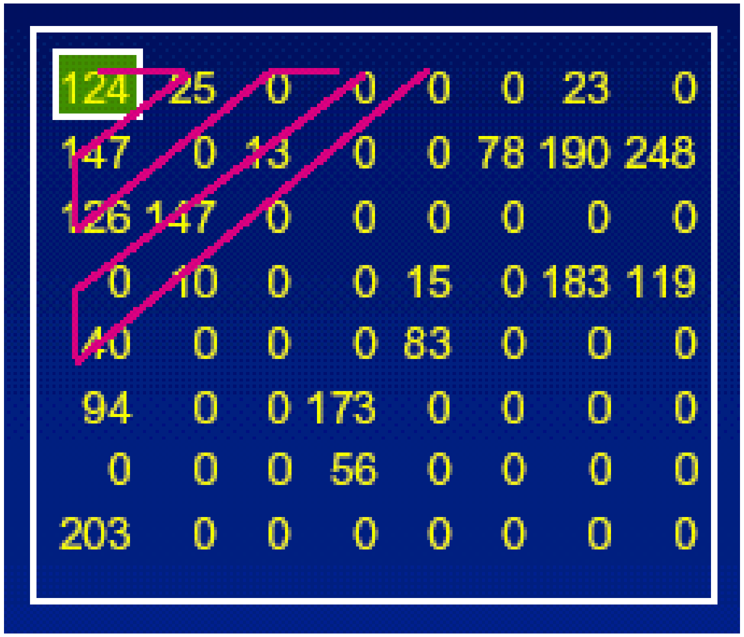
**Quantizing …**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 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 |

# DCT Based Image Coding



# Zig-Zag Serializing the Quantized Coefficients



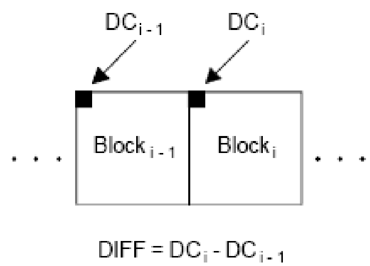
| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 124 | 25 | 0 | 0 | 0 | 0 | 23 | 0 |
| 147 | 0 | 13 | 0 | 0 | 78 | 190 | 248 |
| 126 | 147 | 0 | 0 | 0 | 0 | 0 | 0 |
| 0 | 10 | 0 | 0 | 15 | 0 | 183 | 119 |
| 40 | 0 | 0 | 0 | 83 | 0 | 0 | 0 |
| 94 | 0 | 0 | 173 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 56 | 0 | 0 | 0 | 0 |
| 203 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

**Each DCT block is represented as a sequence of <u>(run, level) pairs</u>, e.g. (0,124), (0, 25), (0,147), (0, 126), (3,13), (0, 147), (1,40) ...**

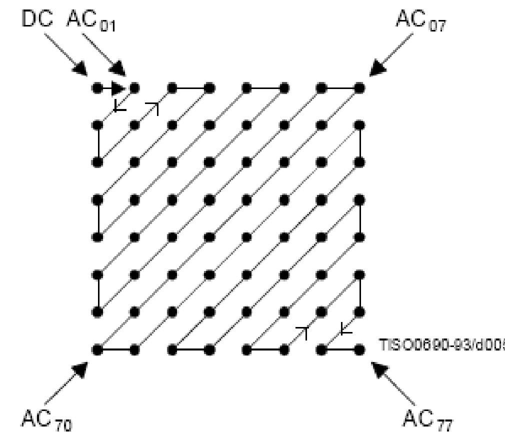
- **For the decoder to reconstruct the matrix with the quantized DCT coefficients, the position and amplitude of the non-null coefficients has to be sent, one after another.**
- **The position of each quantized DCT coefficient may be sent in a relative or absolute way.**
- **The JPEG solution is to send the position of each non-null quantized DCT coefficient through a *run* indicating <u>the number of null DCT coefficients existing between the current and the previous non-null coefficients.</u>**

# Generating the Symbols

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




Differential DC encoding

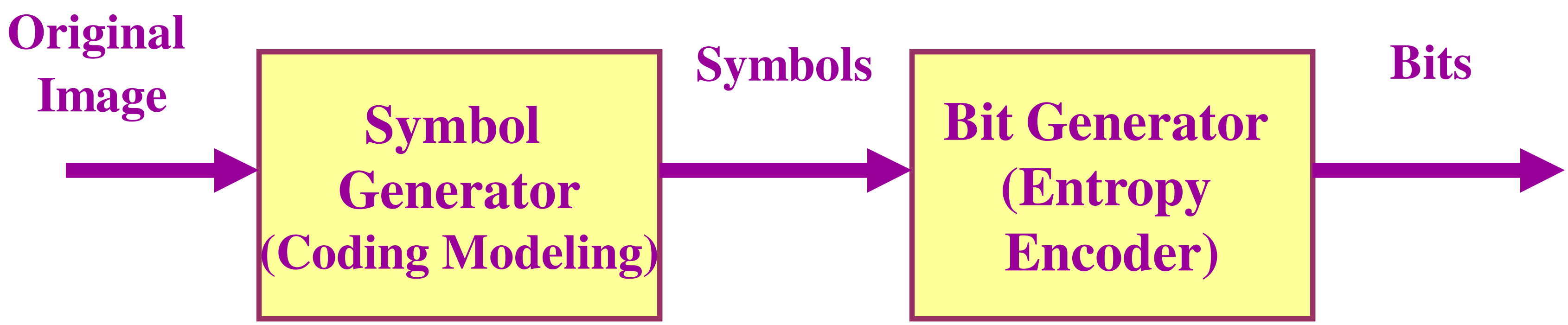



Zig-zag order

- The DC coefficient is treated differently (using differential prediction) because of the high correlation between the DC coefficients of adjacent 8×8 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.**

# JPEG Symbolic Model




**JPEG Model: An image is represented as a sequence of (almost) independent 8×8 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*.**

# 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 lenght depends on the level being coded.**
- **VLI codes are VLC codes where the codeword lenght is previously indicated; they are based on <u>a complement to 2 notation</u>.**
- **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 |
|---|---|---|---|---|---|---|---|
| Runlength | 0 | EOB | Run-size values | | | | |
| | . | 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

| | |
|---|---|
| 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 |

| *Run* | *Size* | *Level* |
|---|---|---|
| Symbol 1 - Huffman (bidimensional) | | Symbol 2 - VLI |

**0011 (-12) → 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) → 1100**

**+12 em binário**

# Summary: How Does JPEG Compress ?

✶ **Spatial Redundancy - DCT**

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

✶ **Irrelevancy**

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

✶ **Statistical Redundancy**

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

# JPEG Extensions

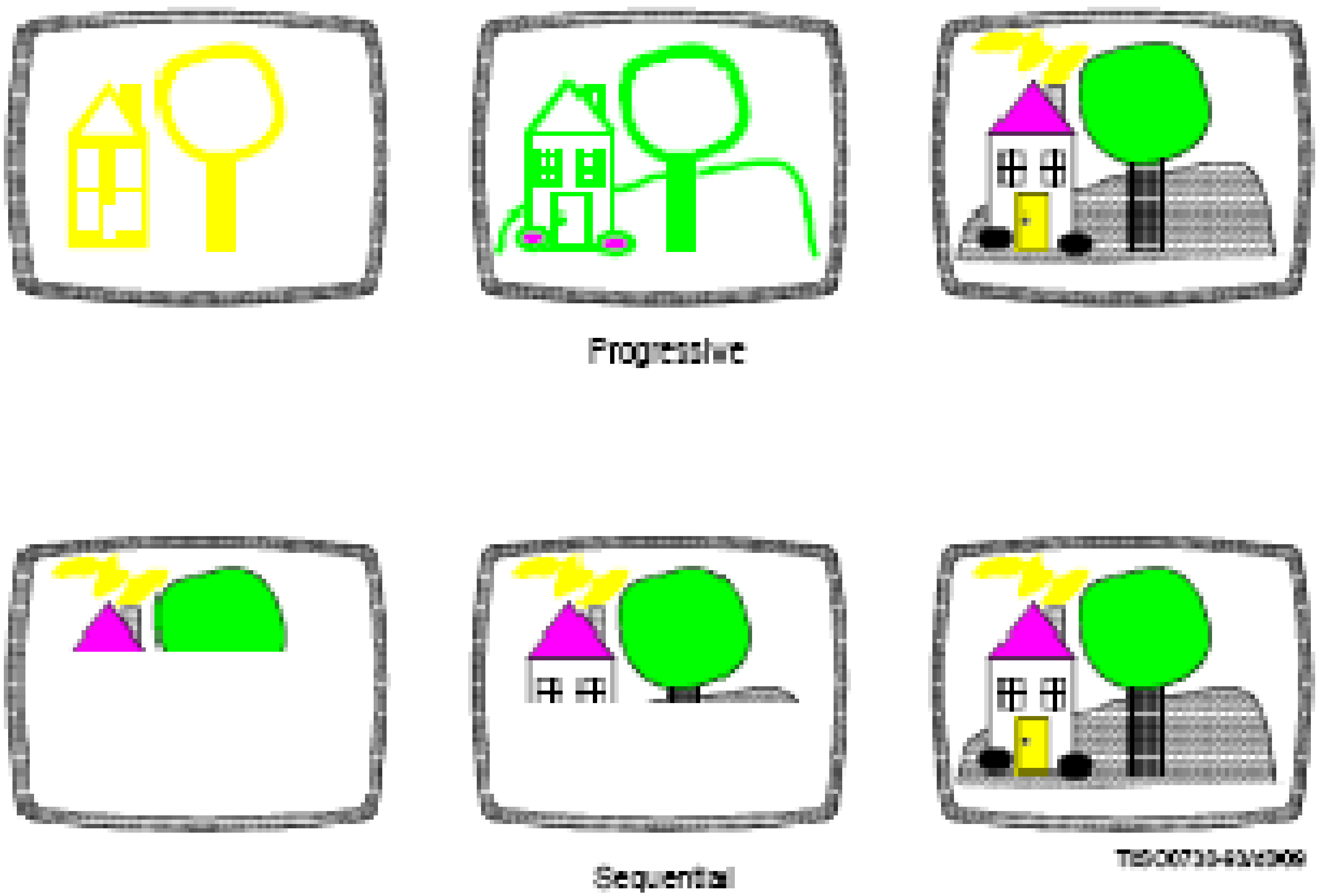
# JPEG Operation Modes

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

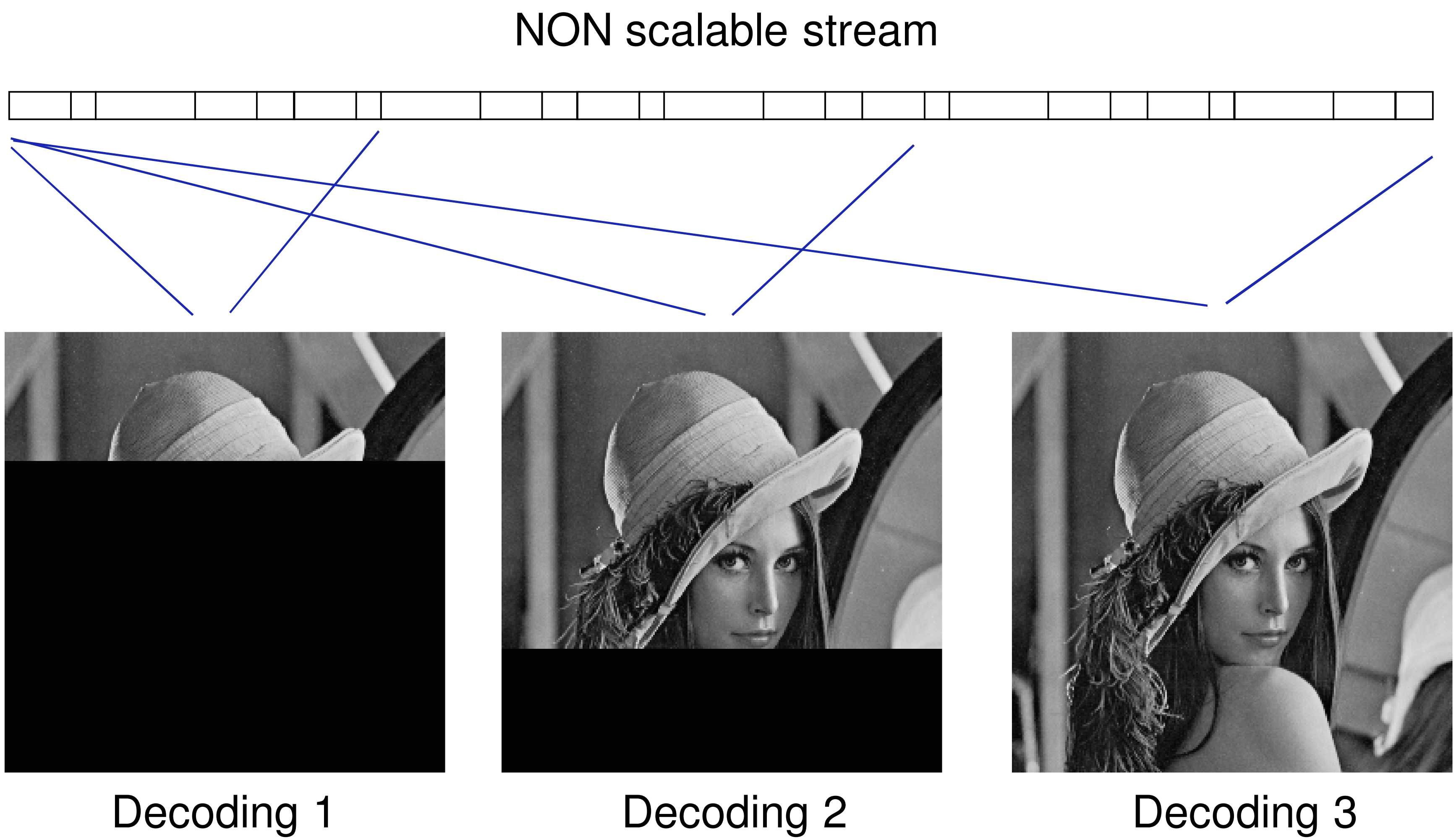
- **SEQUENTIAL MODE – Each image component is coded in a single scan (from top to bottom and left to right).**
- **PROGRESSIVE MODE - The image is coded with several scans which offer a successively better quality.**
- **HIERARCHICAL MODE - The image is coded in several resolutions exploiting mutual 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

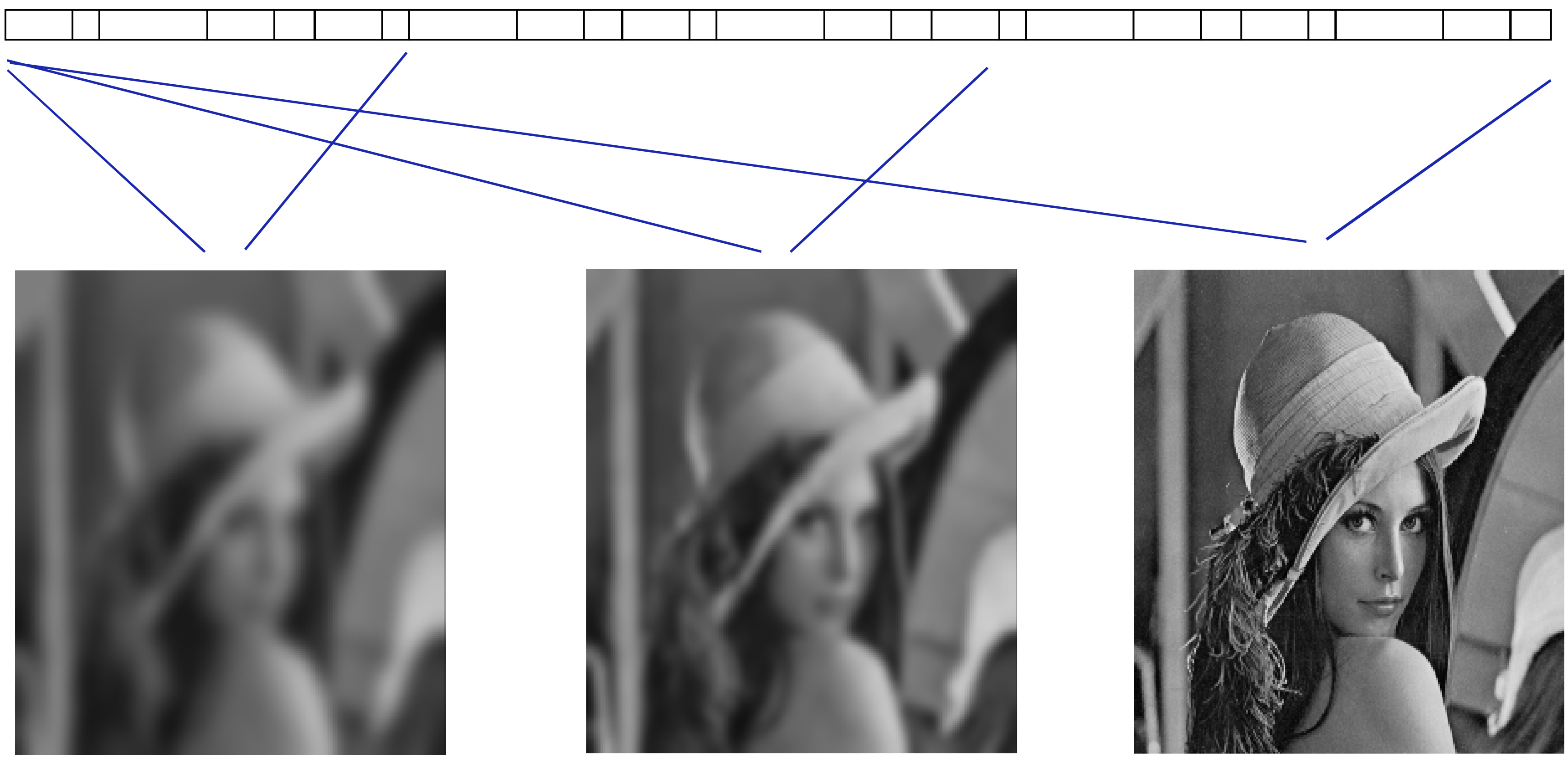


# Sequential Mode or No Scalability ...



# Progressively More Quality: Quality or SNR Scalability

Scalable stream



Decoding 1 Decoding 2 Decoding 3

# 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.

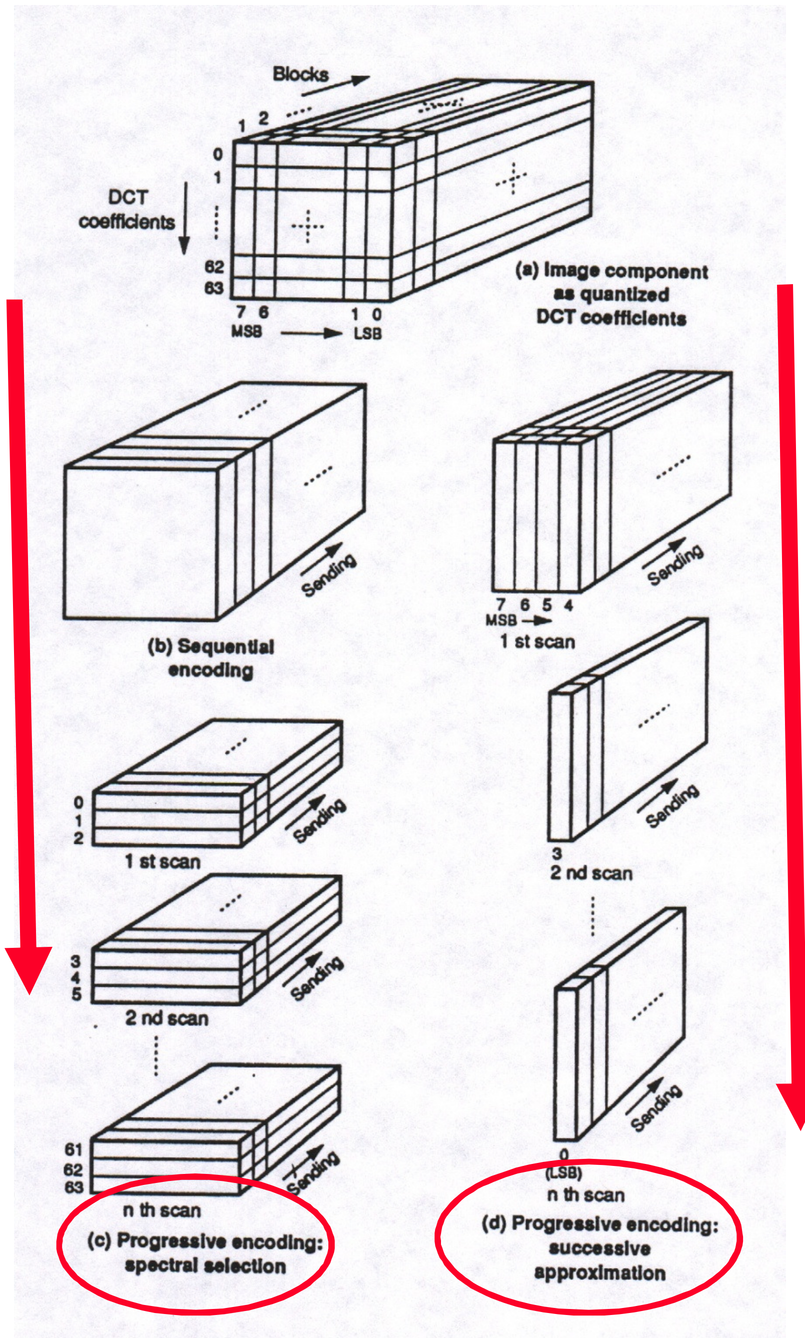
There are methods of implementing the progressive mode:

- ✶ **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.

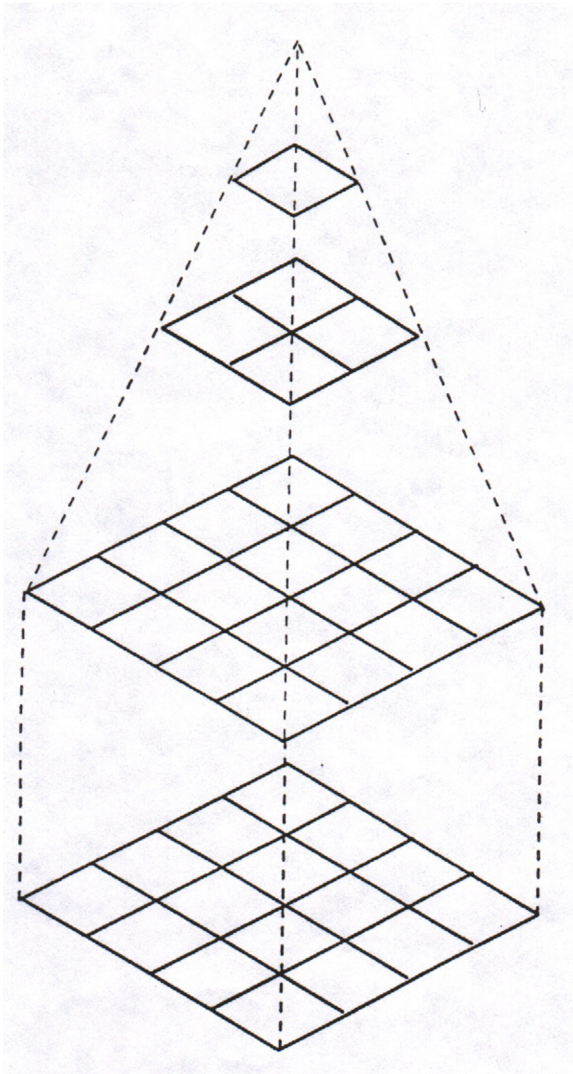
# Progressive Modes: Spectral Selection and Growing Precision

Increasing number of DCT coefficients

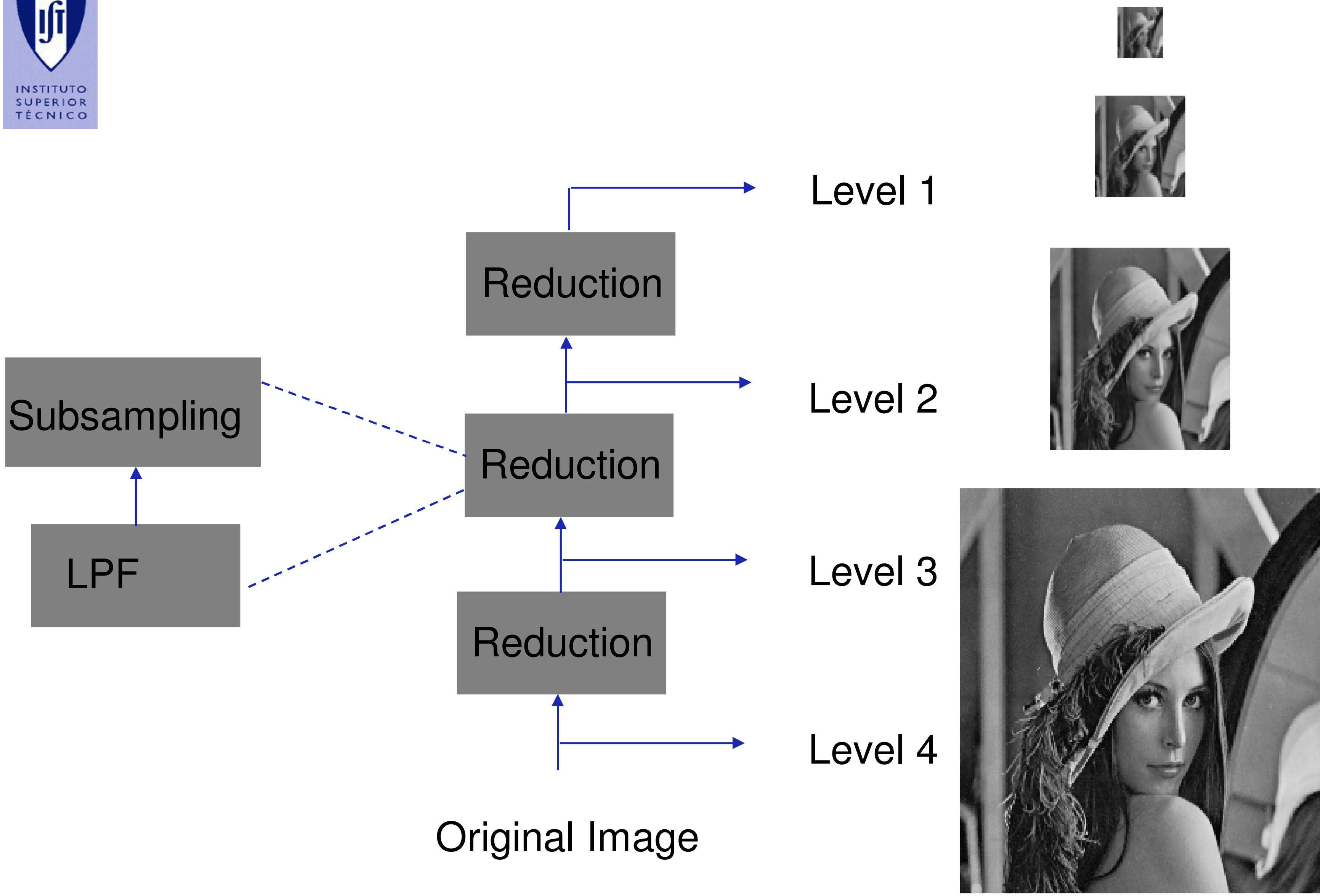



Increasing precision for each coefficient

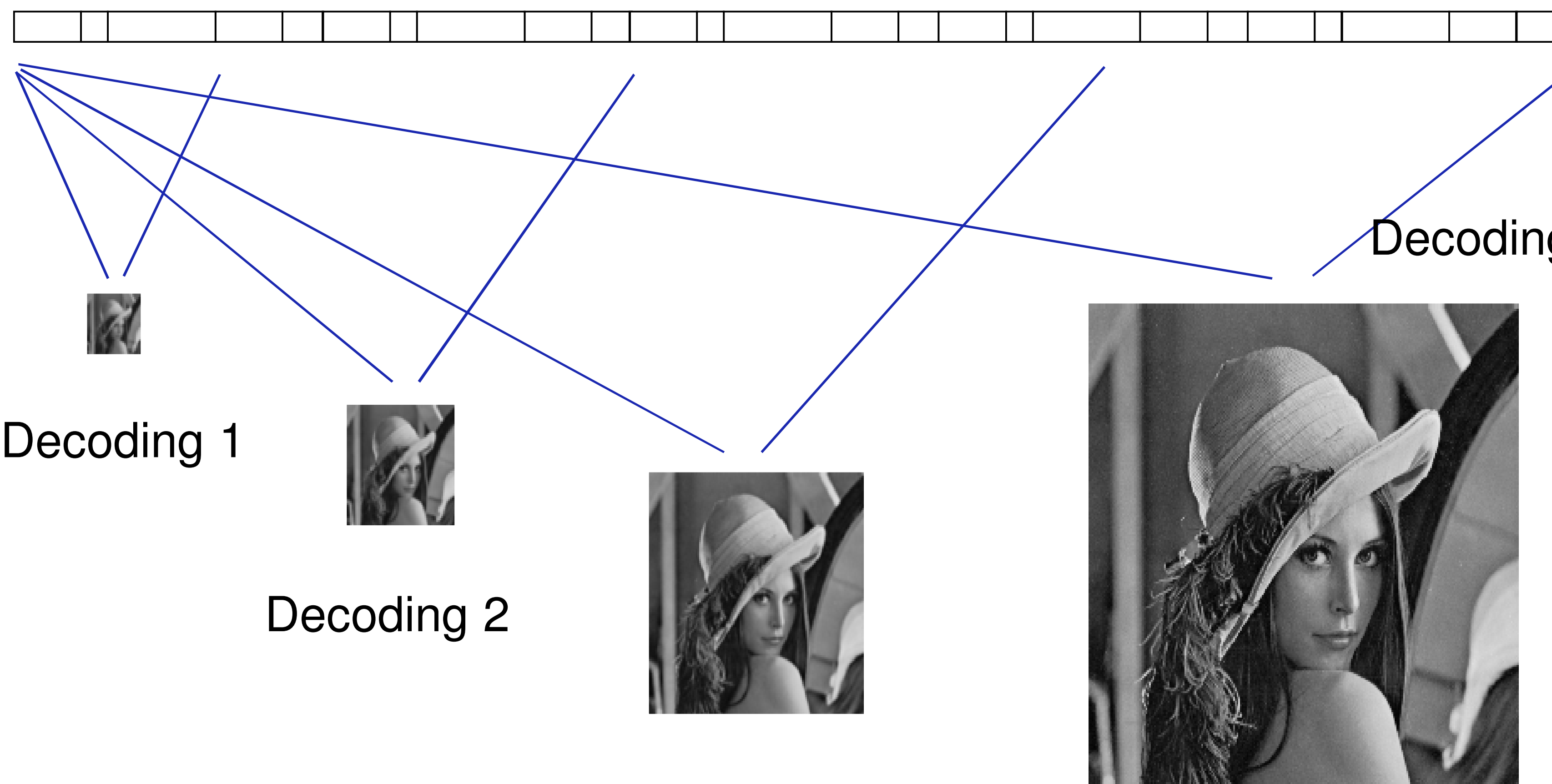
# Hierarchical Mode

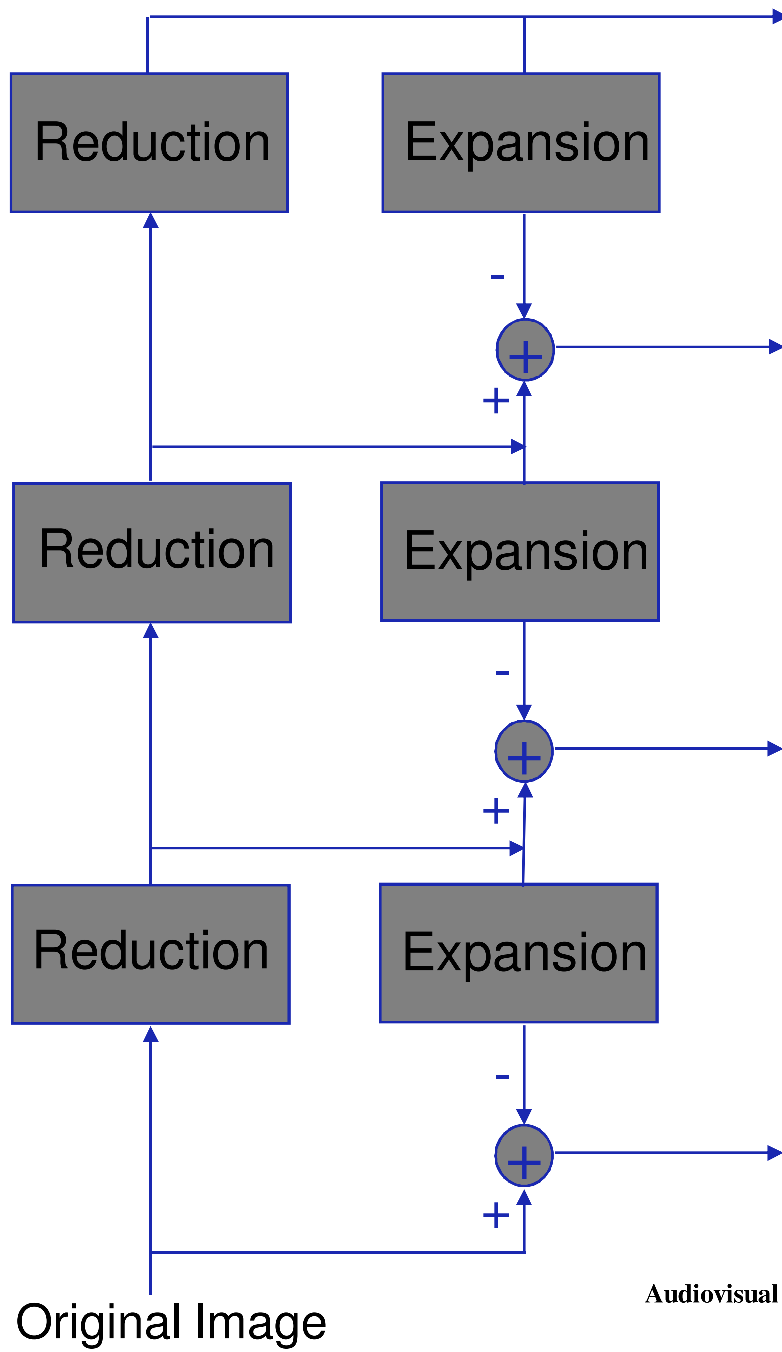


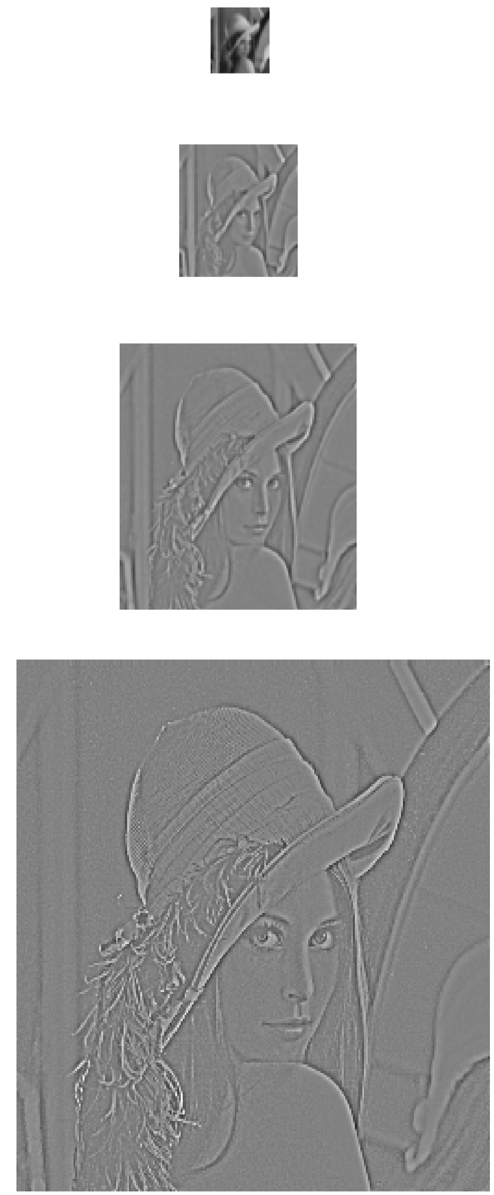
- The hierarchical mode implements a piramidal coding of the image with several resolutions. Each (higher) resolution multiplies by 2 the number of vertical and horizontal samples.
- JPEG hierarchical coding may integrate in the various layers, lossless coding as well as DCT based coding.


Level 1
Reduction
Level 2
Subsampling
Reduction
LPF
Level 3
Reduction
Level 4
Original Image

# Hierarchical Mode or Spatial Scalability …




Reduction
Expansion
-
+
+
Reduction
Expansion
-
+
+
Reduction
Expansion
-
+
+
Original Image



# JPEG Lossless Mode

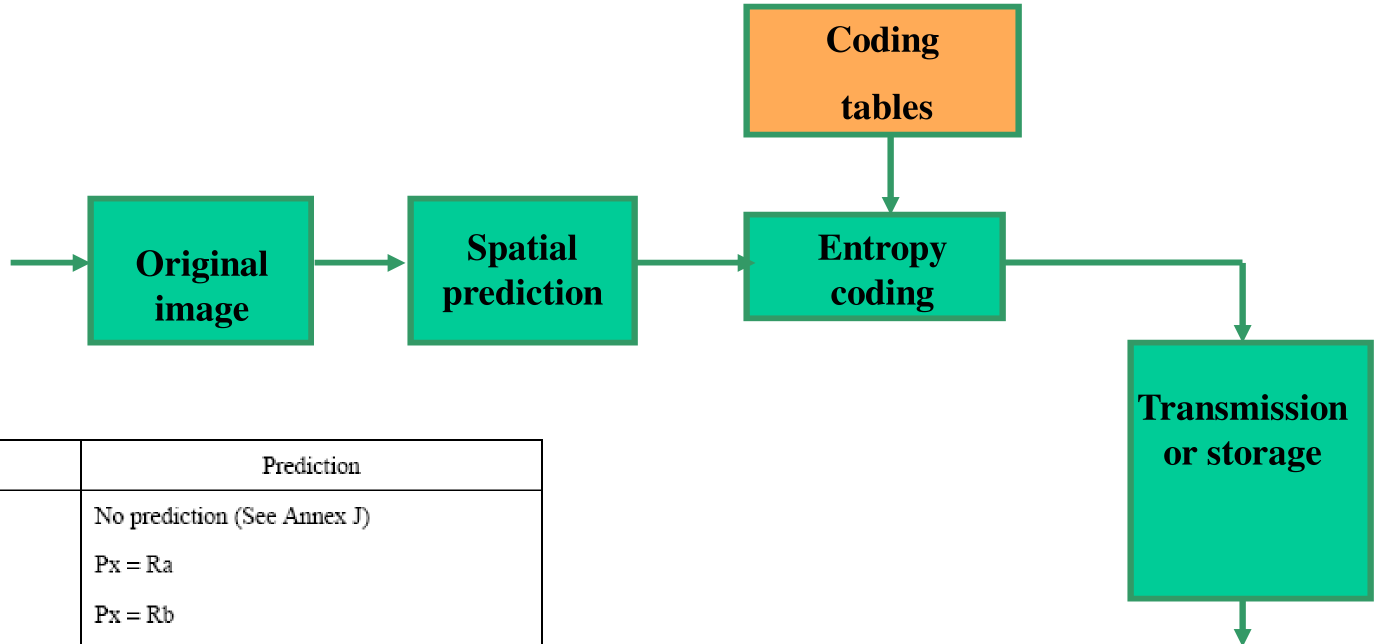
The JPEG lossless mode is based on a spatial predictive scheme. The prediction combines the values of, at most, 3 adjacent pixels. Finally, the prediction mode and the prediction error are coded.

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

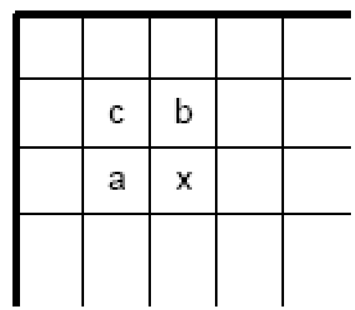
Two codecs are specified for the lossless mode: one using Huffman coding and another using arithmetic coding.

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

# Lossless Coding




| Selection-value | Prediction |
|---|---|
| 0 | No prediction (See Annex J) |
| 1 | Px = Ra |
| 2 | Px = Rb |
| 3 | Px = Rc |
| 4 | Px = Ra + Rb – Rc |
| 5 | Px = Ra + ((Rb – Rc)/2)[a] |
| 6 | Px = Rb + ((Ra – Rc)/2)[a] |
| 7 | Px = (Ra + Rb)/2 |
| [a] Shift right arithmetic operation | |




***Px*** **is the prediction and** ***Ra, Rb*****, and** ***Rc*** **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 <u>some</u> applications
- 0.5 - 0.75 bit/pixel – good to very good quality; enough for <u>many</u> applications
- 0.75 - 1.5 bit/pixel – excellent quality; enough for <u>most</u> applications
- 1.5 - 2.0 bit/pixel – transparent quality; enough for the <u>most demanding</u> 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 coefficents.
- *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 | 0.948 | 34.55 | 37.95 | 36.99 |

# The JPEG 2000 Standard

(not for examination in 2011)

# 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

# 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