# Information and Coding

Armando J. Pinho

Departamento de Electrónica, Telecomunicações e Informática

Universidade de Aveiro

ap@ua.pt

# Contents

# Principles

- Let $x^n = x_1 x_2 \ldots x_n$ be the sequence of values (scalars or vectors) produced by an information source until time $n$.
- Predictive coding is based on encoding sequence $r^n = r_1 r_2 \ldots r_n$, instead of the original sequence $x^n$, where

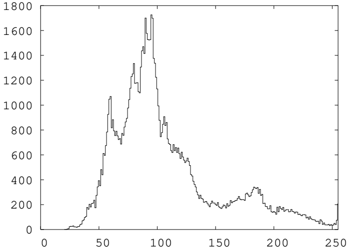
$$r_n = x_n - \hat{x}_n$$

and

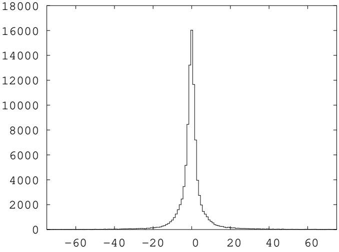
$$\hat{x}_n = p(x^{n-1}) = p(x_1 x_2 \ldots x_{n-1})$$

- The $\hat{x}_n$ are the estimates and the values of the sequence $r^n$ are the residuals.
- Function $p()$ is the estimator or predictor.
- The aim of predictive coding is to have $H(r^n) < H(x^n)$.

# Example




Original
$H = 7.26$ bits/symbol




Predictor 1 JPEG
$H = 4.49$ bits/symbol

# Simple 1D prediction

- Simple polynomial predictors used in some audio encoders:

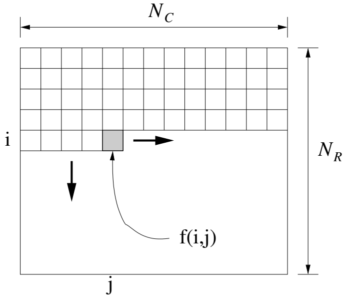
$$\begin{cases} \hat{x}_n^{(0)} = 0 \\ \hat{x}_n^{(1)} = x_{n-1} \\ \hat{x}_n^{(2)} = 2x_{n-1} - x_{n-2} \\ \hat{x}_n^{(3)} = 3x_{n-1} - 3x_{n-2} + x_{n-3} \end{cases}$$

and the corresponding residuals, computed efficiently:

$$\begin{cases} \hat{r}_n^{(0)} = x_n \\ \hat{r}_n^{(1)} = r_n^{(0)} - r_{n-1}^{(0)} \\ \hat{r}_n^{(2)} = r_n^{(1)} - r_{n-1}^{(1)} \\ \hat{r}_n^{(3)} = r_n^{(2)} - r_{n-1}^{(2)} \end{cases}$$

# Predictive image coding techniques

- Typically, images are encoded from left to right, top to bottom, i.e., in raster-scan order:



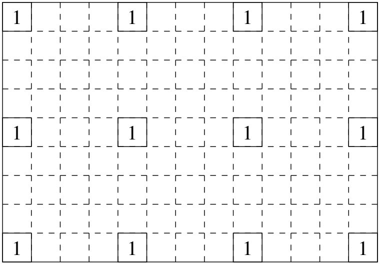

- In this case, the sequence $x^n$ is obtained by concatenating the first $\lfloor n/N_c \rfloor$ image rows, plus the $n \bmod N_c$ pixels from row number $\lfloor n/N_c \rfloor + 1$.

# Predictive image coding techniques

- Other approaches use hierarchical decompositions (or multi-resolution).
- This is the case of the HINT method (Hierarchical INTerpolation):

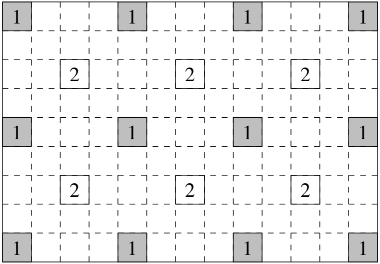
# Predictive image coding techniques

- Other approaches use hierarchical decompositions (or multi-resolution).
- This is the case of the HINT method (Hierarchical INTerpolation):



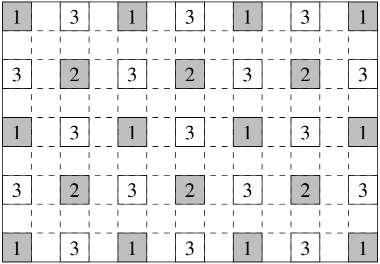
# Predictive image coding techniques

- Other approaches use hierarchical decompositions (or multi-resolution).
- This is the case of the HINT method (Hierarchical INTerpolation):



# Predictive image coding techniques

- Other approaches use hierarchical decompositions (or multi-resolution).
- This is the case of the HINT method (Hierarchical INTerpolation):



# Predictive image coding techniques

- Other approaches use hierarchical decompositions (or multi-resolution).
- This is the case of the HINT method (Hierarchical INTerpolation):

| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | 3 | | 1 | | 3 | | 1 | | 3 | | 1 |
| | 4 | | 4 | | 4 | | 4 | | 4 | | 4 | |
| 3 | | 2 | | 3 | | 2 | | 3 | | 2 | | 3 |
| | 4 | | 4 | | 4 | | 4 | | 4 | | 4 | |
| 1 | | 3 | | 1 | | 3 | | 1 | | 3 | | 1 |
| | 4 | | 4 | | 4 | | 4 | | 4 | | 4 | |
| 3 | | 2 | | 3 | | 2 | | 3 | | 2 | | 3 |
| | 4 | | 4 | | 4 | | 4 | | 4 | | 4 | |
| 1 | | 3 | | 1 | | 3 | | 1 | | 3 | | 1 |

# Predictive image coding techniques

- Other approaches use hierarchical decompositions (or multi-resolution).
- This is the case of the HINT method (Hierarchical INTerpolation):

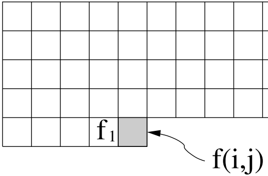
| | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 5 | 3 | 5 | 1 | 5 | 3 | 5 | 1 | 5 | 3 | 5 | 1 |
| 5 | 4 | 5 | 4 | 5 | 4 | 5 | 4 | 5 | 4 | 5 | 4 | 5 |
| 3 | 5 | 2 | 5 | 3 | 5 | 2 | 5 | 3 | 5 | 2 | 5 | 3 |
| 5 | 4 | 5 | 4 | 5 | 4 | 5 | 4 | 5 | 4 | 5 | 4 | 5 |
| 1 | 5 | 3 | 5 | 1 | 5 | 3 | 5 | 1 | 5 | 3 | 5 | 1 |
| 5 | 4 | 5 | 4 | 5 | 4 | 5 | 4 | 5 | 4 | 5 | 4 | 5 |
| 3 | 5 | 2 | 5 | 3 | 5 | 2 | 5 | 3 | 5 | 2 | 5 | 3 |
| 5 | 4 | 5 | 4 | 5 | 4 | 5 | 4 | 5 | 4 | 5 | 4 | 5 |
| 1 | 5 | 3 | 5 | 1 | 5 | 3 | 5 | 1 | 5 | 3 | 5 | 1 |

# Predictors

- For efficient encoding, the estimated values should be as close as possible to the real values, i.e., the $r_k$ values should be small.
- The decoder must be able to generate the same sequence, $\hat{x}^n$, of estimated values, i.e., the predictor cannot introduce any error during encoding / decoding.
- Therefore, the predictor must be causal, and, in lossy coding, the predictor at the encoder must use the reconstructed values, $\tilde{x}^{n-1}$, instead of the original values, $x^{n-1}$.

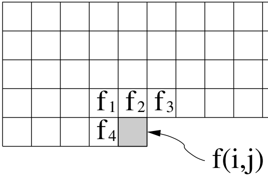
# Predictors

- Generally, the complexity of the predictor depends on two aspects:
  - The number of values used for calculating the estimates (the order of the predictor).
  - The spatial (or temporal) configuration of these values.
- Consider the example of a spatial predictor of order 1, where the estimated value is given by the immediately preceding value:



# Predictors

- This type of predictor can be easily extended to higher orders, using the last *k* processed pixels of the image.
- However, for orders higher than 3 or 4, the efficiency does not increase significantly.
- This happens because images are 2D signals, not 1D sequences of data.
- Therefore, generally, the spatial configurations used for predictive image coding have a 2D shape:

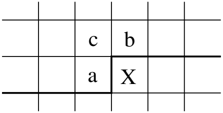


# Lossless predictive coding

- One of the main advantages of predictive coding is allowing a simple design of lossless encoders.
- In fact, most lossless encoders for audio and image rely on predictive coding techniques.
- However, for lossless coding, there is an additional constraint regarding the predictor: the estimates generated must be platform independent.
- Generally, this constraint implies that the predictor can use only integer arithmetic.

# Linear prediction: the lossless mode of JPEG

- The lossless mode of JPEG (ISO/IEC 10918-1, ITU-T T.81, 1992) provides seven linear predictors:




| Mode | Predictor |
|---|---|
| 1 | $a$ |
| 2 | $b$ |
| 3 | $c$ |
| 4 | $a+b-c$ |
| 5 | $a+(b-c)/2$ |
| 6 | $b+(a-c)/2$ |
| 7 | $(a+b)/2$ |

- Generally, the performance of the several predictors may vary considerably from image to image.
- If encoding time is not a problem, then all of them can be tested and the one with the best compression rate chosen.

# Linear prediction: the lossless mode of JPEG

Example:

| Predictor | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| Entropy | 4.49 | 4.21 | 4.74 | 4.17 | 4.16 | **4.04** | 4.10 |

# The nonlinear predictor of JPEG-LS

- JPEG-LS (ISO/IEC 14495-1, ITU-T T.87, 1999) uses a predictor based on the same spatial configuration as that of JPEG:

| | | | | | |
|---|---|---|---|---|---|
| | | c | b | | |
| | | a | X | | |

- However, instead of a linear predictor, it uses the nonlinear predictor

$$\hat{x} = \begin{cases} \min(a,b) & \text{if} \quad c \geq \max(a,b) \\ \max(a,b) & \text{if} \quad c \leq \min(a,b) \\ a+b-c & \text{otherwise} \end{cases}$$

- Note that the linear part of this predictor ($a+b-c$) is the same as predictor number 4 of JPEG.

# The nonlinear predictor of JPEG-LS

Example:

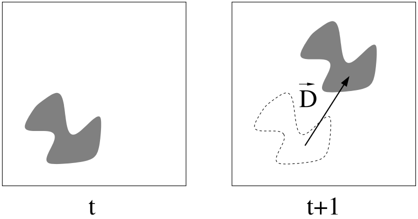
(a)

(b)

| Predictor | 1 | 2 | 3 | 4 | 5 | 6 | 7 | JLS |
|---|---|---|---|---|---|---|---|---|
| Entropy (a) | 4.49 | 4.21 | 4.74 | 4.18 | 4.16 | 4.04 | 4.10 | **3.98** |
| Entropy (b) | 5.60 | 5.05 | 5.82 | 5.19 | 5.23 | 4.97 | 5.15 | **4.93** |

# Motion compensation

- Typically, the differences between consecutive frames of a video sequence are due to motion of the scene objects.
- Exceptions occur when there are scene changes, zoom-in / zoom-out operations and camera translation.




- To explore this redundancy, it is frequent to use temporal prediction (interframe compression), which relies on motion compensation.

# Motion compensation

- Conditional replenishment video coding:
  - Finds zones in the video frame where there were changes with respect to the previous frame.
  - Only those zones are encoded.
  - This technique does not use motion compensation. It just performs a detection of temporal activity.
- Video coding based on motion compensation involves the following steps:
  - Estimation of the motion vectors.
  - Compensation, i.e., temporal prediction.
  - Encoding of the motion vectors.
  - Encoding of the prediction residuals.

# Motion compensation

- There are a large number of techniques for motion detection, but one of them is clearly the most common approach for video coding.
- For each frame block (for example, of $N \times N$ pixels), it seeks the position where it minimizes some measure in relation to the previous frame (the reference frame).
- Note that this approach tries to find the position that minimizes a measure of interest, which might not correspond to the true motion in the scene…

# Motion compensation

- Typically, we want to minimize some measure $C(i, j)$, such as

$$C(i,j) = \sum_{r=1}^{N} \sum_{c=1}^{N} d\Big(g(r,c,t) - g(i+r, j+c, t+1)\Big)$$

where $d(\cdot)$ is, for example, $(\cdot)^2$ or $|\cdot|$.

- Due to complexity constraints, searching is limited to a neighborhood of $(N + 2\Delta) \times (N + 2\Delta)$ pixels around the block, i.e., $-\Delta \leq i, j \leq \Delta$.
- If exhaustive search is used, it is guaranteed to find the minimum of $C(i, j)$...
- This approach is generally computationally too demanding, hence other sub-optimal techniques have been proposed.

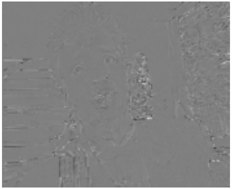
# Motion compensation

- Example:

Frame 200

Frame 201

Direct difference
$H = 5.23$ bpp

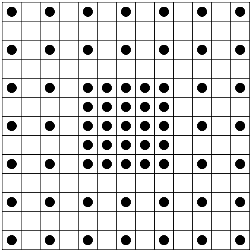


Motion compensation
$H = 4.38$ bpp

# Motion compensation

- Several of the sup-optimal approaches for finding the best reference block rely on spatial sub-sampling.
- For example, considering that the most probable zone for finding the reference block is in the near neighborhood of the block, then we may use the following scheme:

Total: 169 blocks

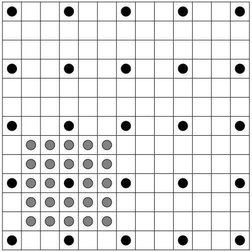


Sub-optimal: 65 blocks

# Motion compensation

- If we consider that after finding a reasonably good reference block it is probable that others better than itself can be found in the near neighborhood, then we can use a greedy search:

Total: 169 blocks

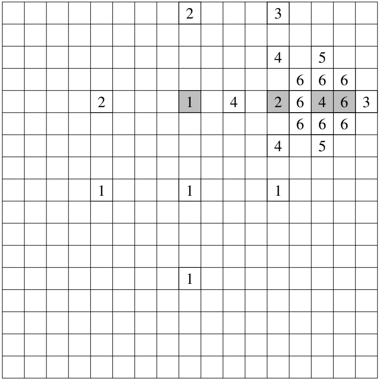


Sub-optimal: 49 blocks

- A number of other variants of local search have been proposed…

# Motion compensation

- For example, the logarithmic search:



# Information and Coding

Armando J. Pinho

Departamento de Electrónica, Telecomunicações e Informática

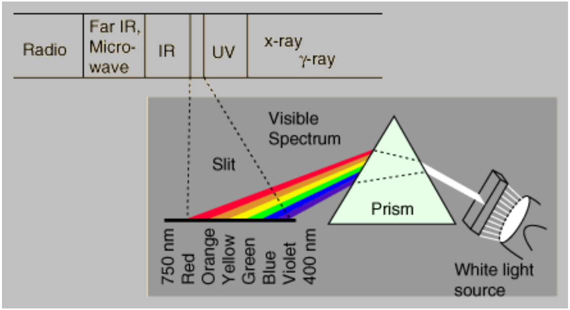
Universidade de Aveiro

ap@ua.pt

# Contents

# The visible spectrum

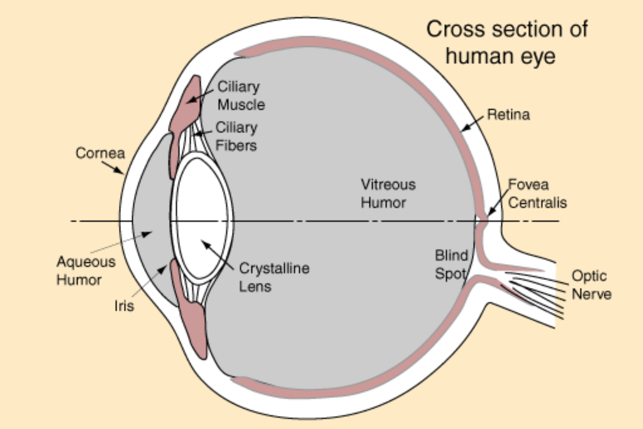
- The typical human eye senses electromagnetic wavelengths between 400 and 700 nm, and has maximum sensitivity around the 555 nm (green zone).



# The human perception of color

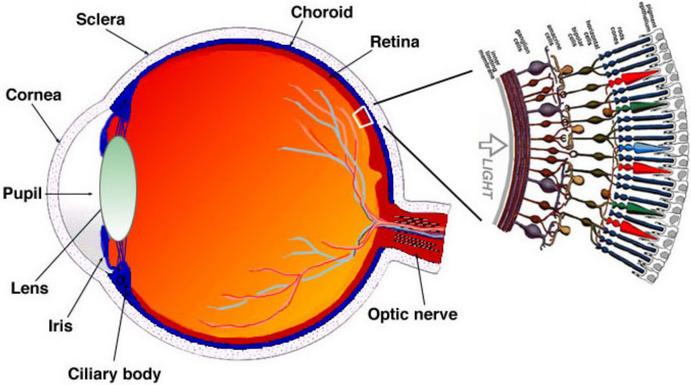
- Normally, the characteristics that allow colors to be distinguished are:
  - The **brightness** (how bright is the color).
  - The **hue** (the dominant color).
  - The **saturation** (how pure is the color).
- Together, the hue and the saturation define the **chromaticity**.
- Therefore, a color can be characterized by the brightness and the chromaticity.

# The human eye



# The human perception of color

- The human eye has **photoreceptors** that are sensitive to short wavelengths ($S$), medium wavelengths ($M$) and long wavelengths ($L$), also known as the blue, green and red photoreceptors.



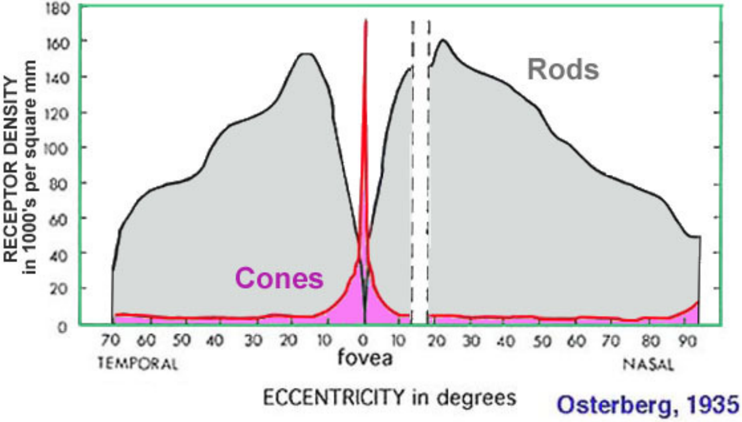
# The photoreceptors: cones and rods

- The **cones**:
  - They provide the photopic vision.
  - They are between 6 and 7 million.
  - They are responsible for the perception of color.
  - There are three types:
    - Sensitive to the blue ($\approx$ 2%)
    - Sensitive to the green ($\approx$ 33%)
    - Sensitive to the red ($\approx$ 65%)
  - They are positioned mainly in the central part of the retina (fovea $\approx$ 0.3 mm diameter).

# The photoreceptors: cones and rods

- The **rods**:
  - They provide the scotopic vision (under low light conditions).
  - They are between 75 and 150 million.
  - They are much more sensitive than the cones, but they are unable to distinguish colors.
  - They allow vision at low levels of light.
  - Because several rods are connected to the same nerve, they provide less spatial resolution.

# Spatial distribution of the photoreceptors



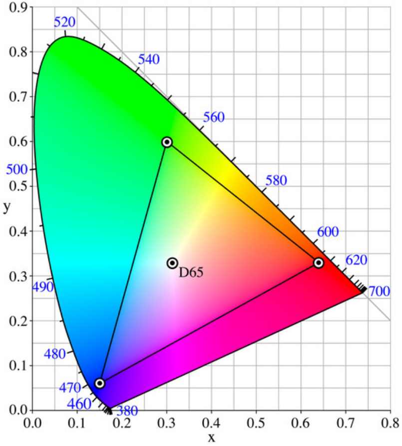
# Additive primaries

- The red, green and blue are the three additive primary colors.

- Adding these three colors produces white.

# The standard *RGB* (*sRGB*) color space

- Chromaticity diagram and corresponding color gamut of *sRGB* (proposed by HP and Microsoft):



# The *sRGB* color space

*R* component

*G* component

*B* component

# The *CMY* color space

- *CMY* is based on the subtractive properties of inks.
- The cyan, magenta and yellow are the **subtractive primaries**. They are the complements, respectively, of the red, green and blue. For example, the cyan subtracts the red from the white.

- Conversion from *RGB* to *CMY*: $C = 1 - R$, $M = 1 - G$, $Y = 1 - B$.

# The *CMY* color space

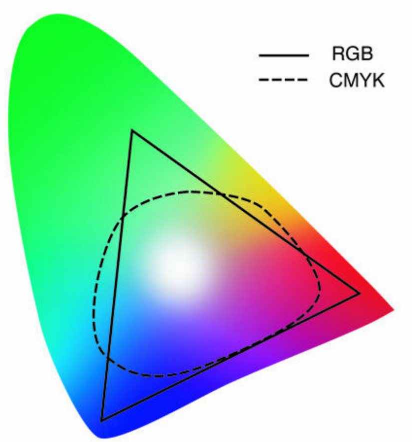
*C* component

*M* component

*Y* component

# The *CMYK* color space

- Due to technological difficulties regarding the reproduction of black, it is generally used the *CMYK* color space for printing.



# The *CMYK* color space

*C* component

*M* component

*Y* component

*K* component

# The *YUV* color space

- The *YUV* color space is used by the PAL television standard.

- $Y$ is the luminance component:

$$Y = 0.299R + 0.587G + 0.114B$$

- Components $U$ and $V$ represent the chrominance:

$$\begin{aligned} U &= -0.147R - 0.289G + 0.436B = 0.492(B - Y) \\ V &= 0.615R - 0.515G - 0.100B = 0.877(R - Y) \end{aligned}$$

- For $R, G, B \in [0, 1]$, we have $Y \in [0, 1]$, $U \in [-0.436, 0.436]$ and $V \in [-0.615, 0.615]$.

# Some advantages of the *YUV* color space

- The *YUV* color space allowed to maintain the compatibility with the old “black and white” television receivers.
- The human eye is more sensitive in the green zone, which is represented mainly by the $Y$ component (the $U$ and $V$ components are related to the blue and red.
- Because the human eye is less sensitive to the blue and red, it is possible to reduce the bandwidth used to represent the $U$ and $V$ components, without introducing significant perceptual degradation.

# The $YC_bC_r$ color space

- $YC_bC_r$ is a family of color spaces used mainly in **digital video** systems. Before being converted to digital format, they are referred to as $YP_bP_r$.
- The $YP_bP_r$ signals are obtained from the $RGB$ signals:

$$Y = K_r R + (1 - K_r - K_b)G + K_b B$$
$$P_b = 0.5(B - Y)/(1 - K_b)$$
$$P_r = 0.5(R - Y)/(1 - K_r)$$

- Constants $K_b$ and $K_r$ depend on the $RGB$ color space that is used.
- With $R, G, B \in [0, 1]$, we have $Y \in [0, 1]$, $P_b \in [-0.5, 0.5]$ and $P_r \in [-0.5, 0.5]$.

# The $YC_bC_r$ color space

- In the case of standard definition television, these constants are $K_b = 0.114$ and $K_r = 0.299$, which result in the conversion equations (ITU-R BT.601):

$$
\begin{aligned}
Y &= \phantom{-}0.299R + 0.587G + 0.114B \\
P_b &= -0.169R - 0.331G + 0.500B \\
P_r &= \phantom{-}0.500R - 0.419G - 0.081B
\end{aligned}
$$

- The conversion to the digital format is given by:

$$
\begin{aligned}
Y &= 16 + 65.481R + 128.553G + 24.966B \\
C_b &= 128 - 37.797R - 74.203G + 112.0B \\
C_r &= 128 + 112.0R - 93.786G - 18.214B
\end{aligned}
$$

- In this case, with $R, G, B \in [0, 1]$, we have $Y \in \{16, \ldots, 235\}$, $C_b, C_r \in \{16, \ldots, 240\}$.

# The $YC_bC_r$ color space

- The JPEG standard allows all 256 values in a 8 bits per component representation.

- In this case, considering $R, G, B \in \{0, \ldots, 255\}$, we have:

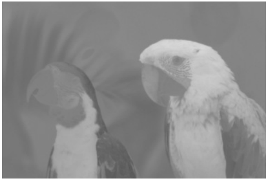
$$
\begin{aligned}
Y &= 0.299R + 0.587G + 0.114B \\
C_b &= 128 - 0.168736R - 0.331264G + 0.5B \\
C_r &= 128 + 0.5R - 0.418688G - 0.081312B
\end{aligned}
$$

- After the conversion, $Y, C_b, C_r \in \{0, \ldots, 255\}$.

# The $YC_bC_r$ color space

$Y$ component

$C_b$ component



$C_r$ component

# The $YC_oC_g$ color space

- This is an easy to compute, losslessly reversible color space (also known as $YC_gC_o$), supported in recent image and video codecs.
- The transformation from $RGB$ to $YC_oC_g$ is given by

$$\begin{bmatrix} Y \\ C_o \\ C_g \end{bmatrix} = \begin{bmatrix} 1/4 & 1/2 & 1/4 \\ 1/2 & 0 & -1/2 \\ -1/4 & 1/2 & -1/4 \end{bmatrix}$$

- The transformation from $YC_oC_g$ to $RGB$ is given by

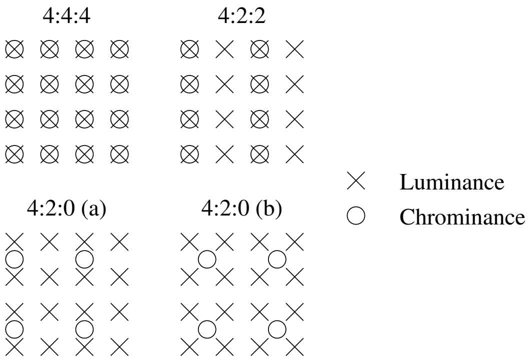
$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 1 & -1 \\ 1 & 0 & 1 \\ 1 & -1 & -1 \end{bmatrix} \begin{bmatrix} Y \\ C_o \\ C_g \end{bmatrix}$$

# Chrominance sub-sampling

- The *YUV* or $YC_bC_r$ color spaces separate the chrominance component (*UV* / $C_bC_r$) from the luminance component ($Y$).
- The human eye is more sensitive to the greens, which are represented mainly by the $Y$ component.
- For this reason, it is common to sub-sample the chrominance components *UV* / $C_bC_r$, producing a reduction in the data rate.
- This reduction is used by both the video coding standards (H.261, MPEG-1, MPEG-2, ...) and the image coding standards (JPEG).

# Chrominance sub-sampling

- The most common types of chrominance sub-sampling:




- The 4:2:0 mode has two variants: (a) used by MPEG-2; (b) used by JPEG, MPEG-1, H.261,. . .

# Example YUV 4:2:0

RGB

$YC_bC_r$ 4:2:0

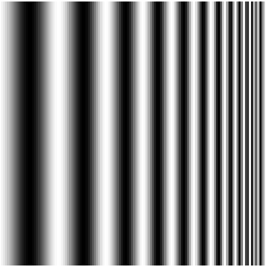
$Y$ component

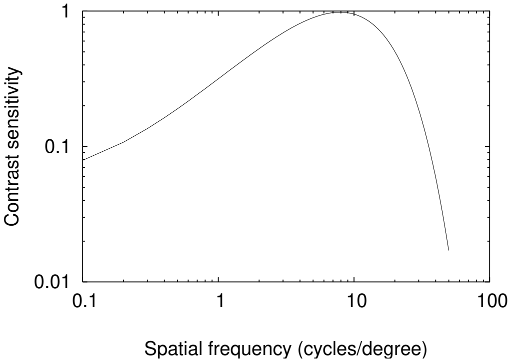
$C_b$ component

$C_r$ component

# Spatial frequency

- The human visual system is characterized by a **bandpass** behavior in the spatial frequency domain:



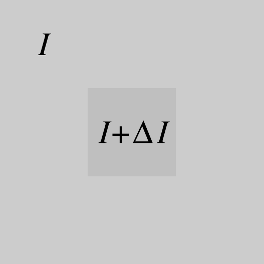


# Weber's law

- Non-linear response to the light intensity (Weber's law):

$$\frac{\Delta I}{I} \approx d(\log I) \approx \text{const.}$$

where $\Delta I$ represents the minimum intensity variation that can be perceived on a background of intensity $I$.



# Visual masking

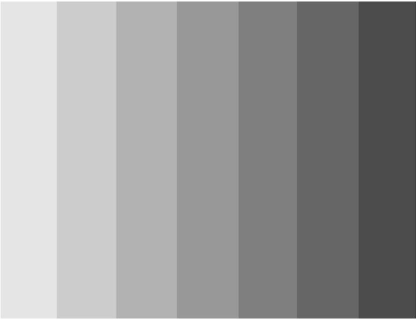
- In zones where large intensity variations occur, small imperfections are masked (i.e., cannot be seen):

Original

Uniform noise: $[-20, 20]$

# Intensity vs. perception

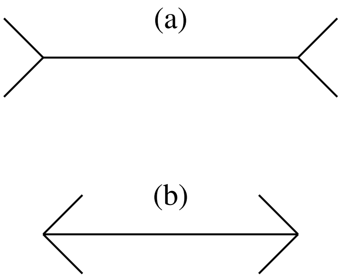
- Mach bands:



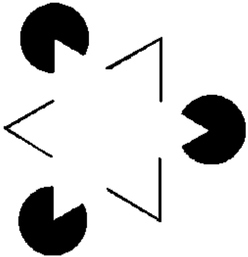
# Intensity vs. perception

- Simultaneous contrast:

# Other illusions. . .

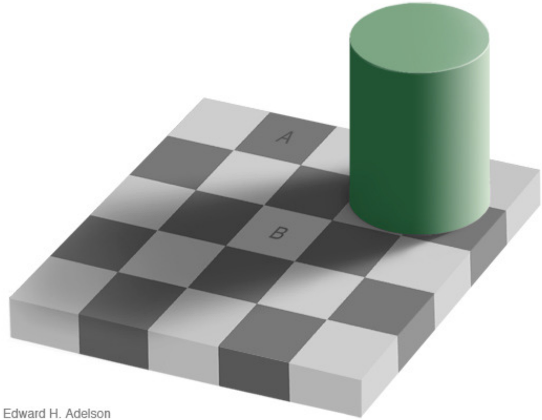



Which is the longer one?



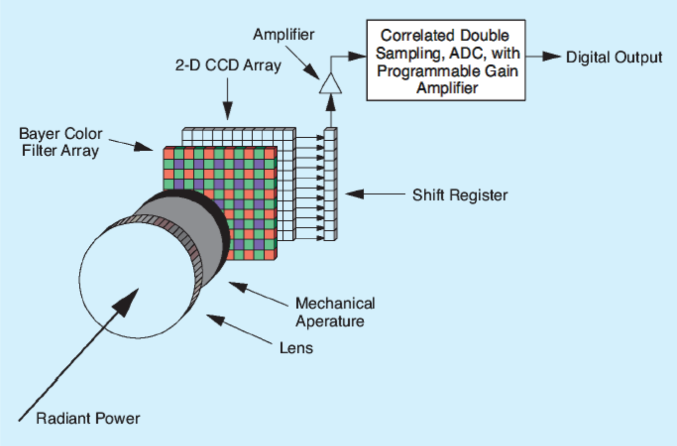
A triangle?

# Other illusions. . .

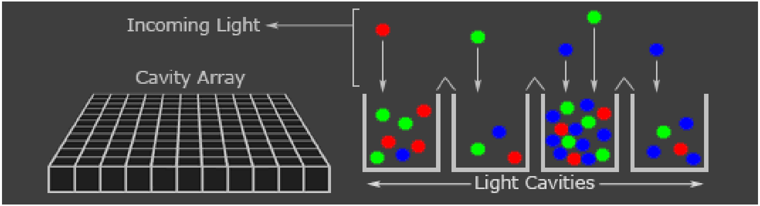


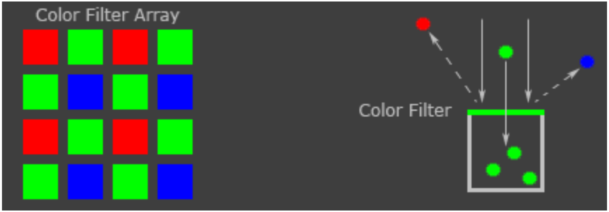
# Digital camera

- Image acquisition using a digital camera:
  (IEEE SP Magazine, Jan 2005)

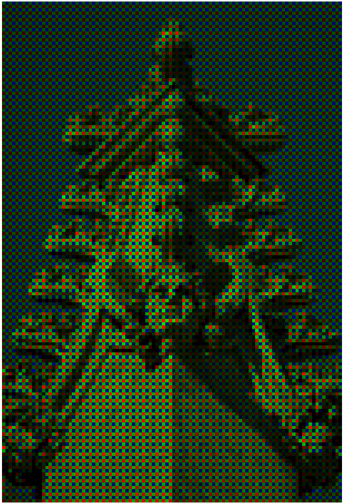


# The Bayer matrix



# The Bayer matrix



# Quality assessment of images

- How to measure the **quality** of images? This is an important and still open problem...
- The techniques for assessing the quality of the images can be classified as **subjective** or **objective**.
- A **subjective evaluation** involves a number of human observers, which can perform **absolute** or **relative** assessments.
- In **relative** assessments, the images are ranked according to the perceived quality.

# Quality assessment of images

- In **absolute** assessments, the observers have to assign a classification, according to some predefined scale, such as,

| | |
|---|---|
| 5. | Excellent |
| 4. | Good |
| 3. | Fair |
| 2. | Poor |
| 1. | Very poor |

- **Subjective evaluation** is the most reliable criterion when the images are intended to be seen by persons. However, these methods are not very practical. . .

# Quality assessment of images

- Typically, the objective criteria are based on the mean squared error or on some other similar measures.
- One of the most popular is the **peak signal to noise ratio**,

$$\mathrm{PSNR} = 10 \log \frac{A^2}{e^2},$$

where $A$ is the maximum value of the signal, $e^2$ is the mean squared error between the reconstructed image, $\tilde{f}$, and the original image, $f$,

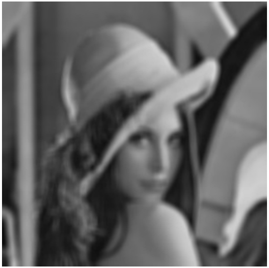
$$e^2 = \frac{1}{N_R N_C} \sum_{i=1}^{N_R} \sum_{j=1}^{N_C} [f(i,j) - \tilde{f}(i,j)]^2.$$

# Quality assessment of images

- This type of similarity measures is used often, due to its simplicity.
- However, it is known that, in some cases, they may fail to provide a good indication of the distortion that really affects the image.

Emax: 32; PSNR: 18.5 dB

Original



Emax: 123; PSNR: 23.9 dB

# Contents

# Motivation

- The main objective of using transforms in the context of data compression is to convert the original data into a new data set **more simple to quantize and encode**.
- Transforms are used **to reduce the statistical dependencies** among the original data (Ideally, the resulting coefficients should be statistically independent).
- Transforms are also used **to separate the relevant information from the irrelevant**, in order to permit course quantization or even removal of the irrelevant information.

# The DCT

- The DCT (Discrete Cosine Transform) is a real and orthonormal transform.
- The analysis/synthesis vectors, $\mathbf{s}_q$, are formed by equally spaced samples of a cosine function with frequencies $f_q = q/(2m)$:

$$s_{q,p} = c_q \cos\left(2\pi f_q (p + 0.5)\right)$$

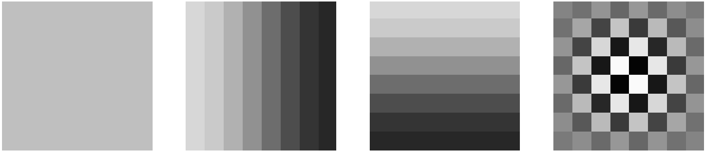
- The normalization factor, $c_q$, is such that $\|\mathbf{s}_q\| = 1$:

$$c_q = \begin{cases} \sqrt{\frac{1}{m}} & \text{if} \quad q = 0 \\ \sqrt{\frac{2}{m}} & \text{if} \quad q \neq 0 \end{cases}$$

# The DCT 2D

- The DCT 2D is obtained through a separable extension of the 1D version:

$$S_{q_1,q_2} = (\mathbf{s}_{q_1,q_2})_{p_1,p_2} = c_{q_1} c_{q_2} \cos\left(2\pi f_{q_1}(p_1 + 0.5)\right) \cos\left(2\pi f_{q_2}(p_2 + 0.5)\right)$$



$S_{0,0}$ $S_{0,1}$ $S_{1,0}$ $S_{7,7}$

- For calculating a single coefficient of a non-separable transform we need $m^2$ operations.
- In the separable case, only $2m$ operations are required.

# The DCT 2D

- The DCT has been one of the most used transforms in the context of image and video coding.
- There are several reasons for this choice:
  - It provides a good energy compaction and reduction of the correlation among the coefficients.
  - It uses only real numbers.
  - There are fast algorithms, based on the FFT (Fast Fourier Transform), that can be used for its calculation.

# The DCT 2D

- Let us see how the DCT attains energy compaction, considering, for example, the following $8 \times 8$ block of pixels:

$$\mathbf{X} = \begin{bmatrix} 183 & 160 & 94 & 153 & 194 & 163 & 132 & 165 \\ 183 & 153 & 116 & 176 & 187 & 166 & 130 & 169 \\ 179 & 168 & 171 & 182 & 179 & 170 & 131 & 167 \\ 177 & 177 & 179 & 177 & 179 & 165 & 131 & 167 \\ 178 & 178 & 179 & 176 & 182 & 164 & 130 & 171 \\ 179 & 180 & 180 & 179 & 183 & 169 & 132 & 169 \\ 179 & 179 & 180 & 182 & 183 & 170 & 129 & 173 \\ 180 & 179 & 181 & 179 & 181 & 170 & 130 & 169 \end{bmatrix}$$

# The DCT 2D

- The coefficients (rounded to the integers), resulting from applying the DCT to the block (after subtracting $2^7 = 128$ to each pixel), are:

$$\mathbf{Y} = \begin{bmatrix} 313 & 56 & -27 & 18 & 78 & -60 & 27 & -27 \\ -38 & -27 & 13 & 44 & 32 & -1 & -24 & -10 \\ -20 & -17 & 10 & 33 & 21 & -6 & -16 & -9 \\ -10 & -8 & 9 & 17 & 9 & -10 & -13 & 1 \\ -6 & 1 & 6 & 4 & -3 & -7 & -5 & 5 \\ 2 & 3 & 0 & -3 & -7 & -4 & 0 & 3 \\ 4 & 4 & -1 & -2 & -9 & 0 & 2 & 4 \\ 3 & 1 & 0 & -4 & -2 & -1 & 3 & 1 \end{bmatrix}$$

# The JPEG standard

- The JPEG (Joint Photographic Experts Group) standard is a family of coding methods for images of continuous tones of grays or colors.
- The group was established in 1986, the standard was proposed in 1992 and approved in 1994 (ISO 10918-1).
- The JPEG standard comprises **four coding methods**: sequential, progressive, hierarchical and lossless.
- The JPEG standard is based on a number of compression techniques, such as the DCT, statistical coding and predictive coding.

# The sequential mode of JPEG

- Every codec should include this mode in order to be considered JPEG-compatible (it is also known as the “baseline” mode).
- The sequential mode of JPEG comprises the following steps:
  - Calculation of the DCT.
  - Quantization of the DCT coefficients, in order to eliminate less relevant information, according to the characteristics of the human visual system.
  - Statistical coding (Huffman or arithmetic) of the quantized DCT coefficients.

# The sequential mode of JPEG

- Calculation of the DCT:
  - The image is partitioned into $8 \times 8$ blocks of pixels. If the number of rows or columns is not multiple of 8, then they are internally adjusted (using padding).
  - Subtract $2^{b-1}$ to each pixel value, where $b$ is the number of bits used to represent the pixels.
  - Calculate the DCT 2D of each block.

# The sequential mode of JPEG

- Quantization of the DCT coefficients:
  - The DCT coefficients are quantized using a quantization matrix, previously scaled by a compression quality factor.
  - Next, the coefficients are organized in a one-dimensional vector according to a zig-zag scan.
- Statistical coding:
  - The non-zero AC coefficients are encoded using Huffman or arithmetic coding, representing the value of the coefficient, as well as the number of zeros preceding it.
  - The DC coefficient of each block is predictively encoded in relation to the DC coefficient of the previous block.

# Quantization of the coefficients

- The DCT, alone, does not provide data compression.
- In fact, each $m \times m$ block of pixels is transformed into another $m \times m$ block, usually requiring higher precision for representing its elements.
- Typically, compression is obtained through the concatenation of two distinct processes:
  - Quantization of the coefficients resulting from the transformation.
  - Use of statistical coding.

# Quantization of the coefficients

- Compression is obtained due to the **low-pass characteristic of the human visual system**.
- Because of this characteristic, generally more bits are assigned to the low frequencies (those appearing in the upper left corner of the transformed block).
- This is done using **threshold coding** (non-linear approximation).

# Quantization of the coefficients

- **Threshold coding** is based on the use of one or more decision levels, such that coefficients below the thresholds are eliminated.
- This way, block to block variations can be accommodated.
- Generally, the thresholding and quantization operations are done together, through a **quantization matrix**

$$\tilde{y}(r,c) = \mathrm{ROUND}\left(\frac{y(r,c)}{q(r,c)}\right),$$

where $\tilde{y}(r,c)$ is the quantized version of $y(r,c)$, and $q(r,c)$ is the corresponding element of the quantization matrix, $Q$.

# Quantization of the coefficients

- Generally, the elements of $Q$ are 8 bit integers that determine the quantization step according to the position of each coefficient.
- **Example:** quantization matrix of JPEG (luminance):

$$Q = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 129 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

# Quantization of the coefficients

- By applying this quantization matrix to the block that we have previously used as example, we obtain the matrix $\widetilde{Y}$:

$$\widetilde{Y} = \begin{bmatrix} 20 & 5 & -3 & 1 & 3 & -2 & 1 & 0 \\ -3 & -2 & 1 & 2 & 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

- In this example, 45 of the 64 coefficients are eliminated.

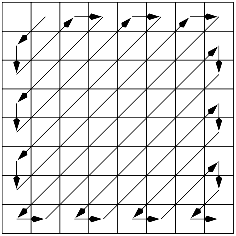
# Quantization of the coefficients

- Because the sensitivity of the human eye to the colors is different from that of the luminance, JPEG provides a different quantization matrix for the chrominance components:

$$Q = \begin{bmatrix} 17 & 18 & 24 & 47 & 99 & 99 & 99 & 99 \\ 18 & 21 & 26 & 66 & 99 & 99 & 99 & 99 \\ 24 & 26 & 56 & 99 & 99 & 99 & 99 & 99 \\ 47 & 66 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \\ 99 & 99 & 99 & 99 & 99 & 99 & 99 & 99 \end{bmatrix}$$

# Coefficient coding

- JPEG uses a zig-zag scanning of $\widetilde{Y}$ in order to encode the quantized coefficients, except for the $(0, 0)$ position, i.e., the DC coefficient.



- The objective of this scanning is to group together the zero coefficients, allowing a more efficient representation.
- This efficiency is obtained using a variant of run-length coding.

# Coefficient coding

- Using again the same example, a JPEG encoder would generate the following codewords:

$$(0,5),(0,-3),(0,-1),(0,-2),(0,-3),(0,1),(0,1),(0,-1),(0,-1),$$
$$(2,1),(0,2),(0,3),(0,-2),(0,1),(0,1),(6,1),(0,1),(1,1),\mathrm{EOB}$$

$$\begin{bmatrix} 20 & 5 & -3 & 1 & 3 & -2 & 1 & 0 \\ -3 & -2 & 1 & 2 & 1 & 0 & 0 & 0 \\ -1 & -1 & 1 & 1 & 1 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

# The visual effect of the coding blocks

- The coding techniques that are based on a partition of the image into blocks are generally affected by a visual phenomenon known as the **blocking artifact**.

This artifact is more visible when the compression ratio is high and happens because the blocks are encoded independently (except for the DC coefficients).

**Example:** $8 \times 8$ DCT, 0.31 bpp.

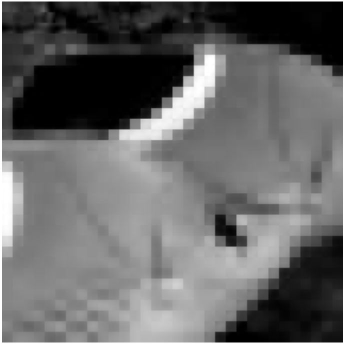
# The progressive mode of JPEG

- This mode relies on encoding the DCT coefficients using several passes, such that in each pass only part of the information associated to those coefficients is transmitted.
- JPEG provides two methods for doing this:
  - **Spectral selection**: the coefficients are organized in spectral bands, and those corresponding to the lower frequencies are transmitted first.
  - **Successive approximation**: all coefficients are first transmitted using a limited precision. Afterward, additional detail is sent using more passes through the coefficients.

# The progressive mode of JPEG

- Sequential *vs.* progressive:

Sequential, 1000 bytes



Progressive, 1000 bytes

# The progressive mode of JPEG

- Sequential *vs.* progressive:

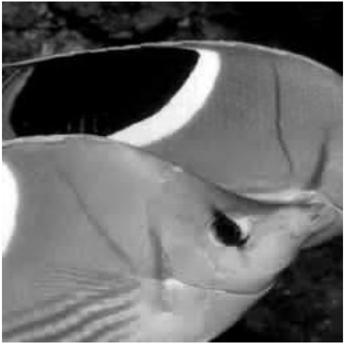
Sequential, 2000 bytes

Progressive, 2000 bytes

# The progressive mode of JPEG

- Sequential *vs.* progressive:

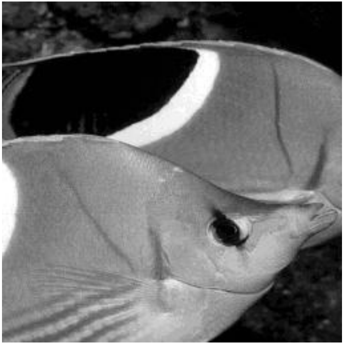
Sequential, 4000 bytes



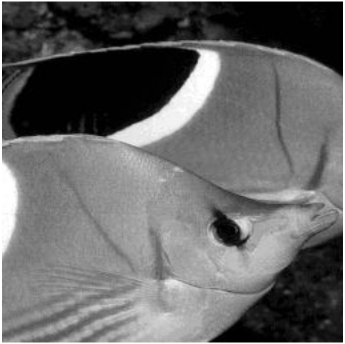
Progressive, 4000 bytes

# The progressive mode of JPEG

- Sequential *vs.* progressive:



Sequential, 10023 bytes



Progressive, 10198 bytes

# Contents

# H.261

- H.261 (1990) is a ITU-T video coding standard (Video Codec for Audiovisual Services at p$\times$64 kbit/s) that was developed with the aim of being used
  - In video-phone applications.
  - In video-conference applications.
  - Over ISDN links at $p \times 64$ kbps, $p = 1, \ldots, 30$.
- For example, $p = 1$ (64 kbps) would be appropriated for video-phone, where the video signal was transmitted at 48 kbps and the audio signal at 16 kbps.
- Generally, video-conference required better image quality, implying typically $p \geq 6$ (384 kbps).
- For $p = 30$ we have 1.92 Mbps, which was sufficient for a video quality similar to the old VHS tapes.

# H.261

- Because this standard was intended for bi-directional real-time communication, the maximum delay allowed in the coding process is 150 milliseconds.
- It allows only two frame formats: CIF (Common Intermediate Format) $352 \times 288$, and QCIF (Quarter Common Intermediate Format) $176 \times 144$...
- ...and frame-rates of 30 Hz, 15 Hz, 10 Hz and 7.5 Hz.
- Notice that, even for QCIF at 10 Hz, it is required a compression of at least 1:48 for transmission in a 64 kbps channel.

# H.261

- The encoded stream has the following structure:
  - At the top, the **frame**.
  - Each frame is partitioned into several **groups of blocks**.
  - Each group of blocks is formed of several **macroblocks**.
  - The macroblock is the smallest region that can have a particular coding mode assigned to.
  - The macroblock is composed of four basic **blocks** (a basic block is $8 \times 8$) of luminance ($Y$) and by the corresponding $8 \times 8$ chrominance blocks ($C_r$ and $C_b$).

# H.261

- The H.261 uses two compression modes:
  - **Intraframe**: similar to the JPEG compression, i.e., relies on DCT applied to $8 \times 8$ blocks of pixels.
  - **Interframe**: temporal prediction (motion compensation), followed by DCT of the prediction residuals.
- Motion compensation (MC) is performed in macroblocks, within a search area of 15 pixels around the macroblock.
- It has 32 quantizers, one of them dedicated to the DC coefficient in intraframe mode (quantization step of 8). The others have quantization steps from 2 to 62.
- Statistical coding is performed with Huffman codes.

# MPEG–1

- MPEG–1 (1992) is a ISO/IEC (11172) coding standard that has been developed with the aim of storing video and audio in CD-ROMs.
- Target bitrates were around 1.5 Mbps, which was the bitrate associated to the early CD-ROM readers.
- The main objective of MPEG–1 was to provide means for encoding audio and video for interactive multimedia applications.
- For video segments having a moderate motion content, quality similar to VHS could be attained for MPEG–1 video at 1.2 Mbps.

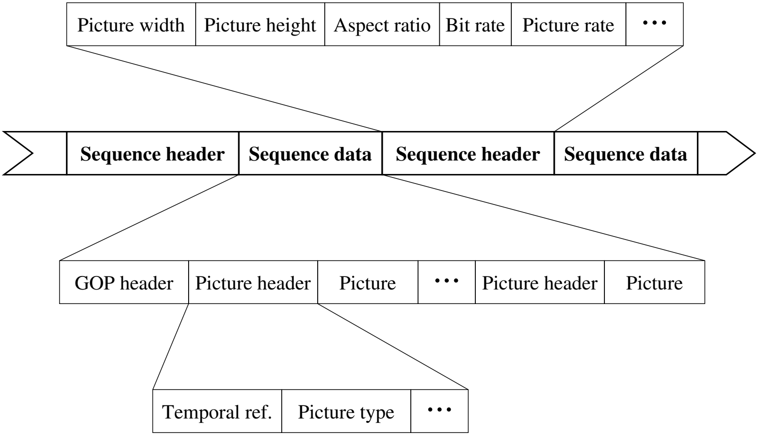
# MPEG–1

- The algorithms used in MPEG–1 are similar to those of H.261, although having some additional characteristics, such as
  - Random access (using type I frames)
  - Fast forward and reverse.
  - Backwards playing.

- Generally, the input is in the CCIR 601 format ($576 \times 720$, for a 50 fps or $480 \times 720$ for 60 fps), and is converted to SIF (Source Input Format) before encoding (luma with $288(240) \times 352$ pixels and chroma with $144(120) \times 176$ pixels).

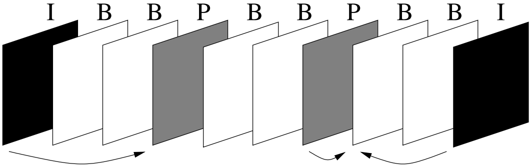
# MPEG–1

## Organization of the bitstream



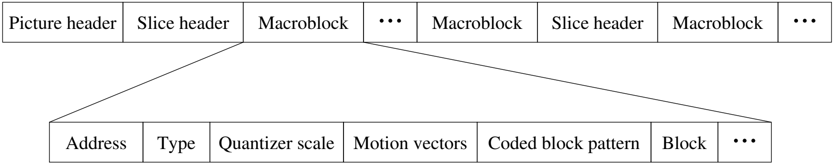
# MPEG–1

- MPEG–1 allows three types of **frames**:
    - **Type I**: encoding is similar to that of JPEG. These frames serve as entry points for random access.
    - **Type P**: frames encoded in predictive mode, using as reference previous frames of type I or P.
    - **Type B**: frames encoded in predictive mode, using both reference frames from the past and from the future (of type I or P).
- The number of I, P and B frames composing a **group of frames** depends on the application.



# MPEG–1

- The **slices** provide resynchronization capabilities, in case of errors.




- In summary, the main operations performed by a MPEG–1 video encoder are
  - Choose the type of frame (I, P or B).
  - Estimate the motion vector for each macroblock (only for type P and B frames).
  - Find the coding mode for each macroblock.
  - Find the appropriate quantization step for rate control.

# MPEG–2

- MPEG–2 (1994) has been developed aiming applications such as
  - Transmission of television signals in standard definition formats (PAL, SECAM, NTSC).
  - High definition television (HDTV).
  - Electronic cinema.
  - Games and high quality multimedia applications.
  - ...
- Some characteristics of MPEG–2 video:
  - Bitrates up to 100 Mbps.
  - More choices in terms of spatial and temporal resolution.
  - Support for interlaced video (notion of even and odd field).
  - More possibilities for the chrominance sub-sampling.
  - More coding and quantization options.
  - Support for bitstream **scalability**.

# H.263

- Initially (1993), the MPEG–4 group started developing a video coding standard for bitrates $< 64$ kbps, i.e., for **very low bitrates**.
- However, some time after, this line was reformulated into a much more ambitious objective: that of creating a standard for coding audiovisual objects.
- Due to the urgent need for a low bitrate standard (for example, for enabling video over the analog public telephone network or over wireless channels), the work was divided in two phases:
  - One, for the immediate development of a video coding standard for very low bitrates: recommendation H.263 (1995).
  - The other, directed to a more vast set of tools, originated the MPEG–4 standard.

# H.263

- Recommendation H.263 specifies an algorithm for video coding, similar to that of H.261, for bitrates of about 22 kbps of a total of 28.8 kbps.
- The main differences between H.261 and H.263 are:
  - New formats available: sub-QCIF, 4CIF and 16CIF, in addition to those already supported by H.261, CIF and QCIF.
  - Possibility of using a motion vector per block as well as one motion vector per macroblock.
  - Half-pixel precision motion estimation and prediction of motion vectors.
  - Arithmetic coding.
  - PB-frames (bi-directional prediction, similar to that used in MPEG).

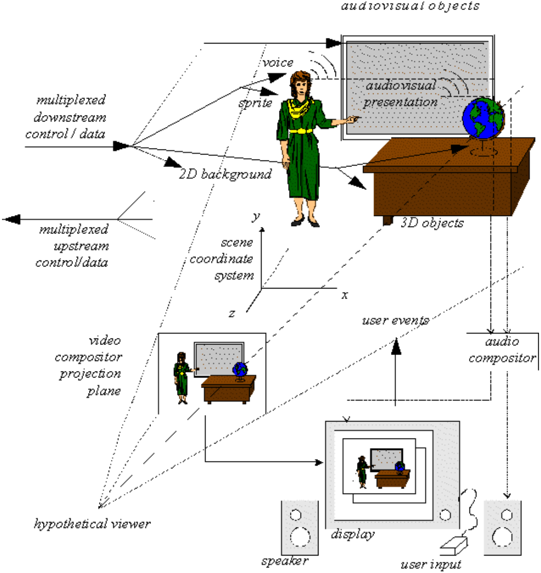
# MPEG–4

- MPEG–4 (initial version in 1998) is a ISO/IEC standard providing tools for:
  - **Representing** audio, video or audiovisual data through **media objects** that can be natural (i.e., captured by a microphone or video camera) or synthetic (i.e., computer generated).
  - Describing the **composition** of these objects for creating composed objects and audiovisual scenes.
  - **Multiplexing and synchronizing** the data associated to the media objects, for transmission through the communication channels, providing an appropriate quality of service (QoS) to each object.
  - Enabling the **interaction** of the clients (receptor) with the audiovisual scene.

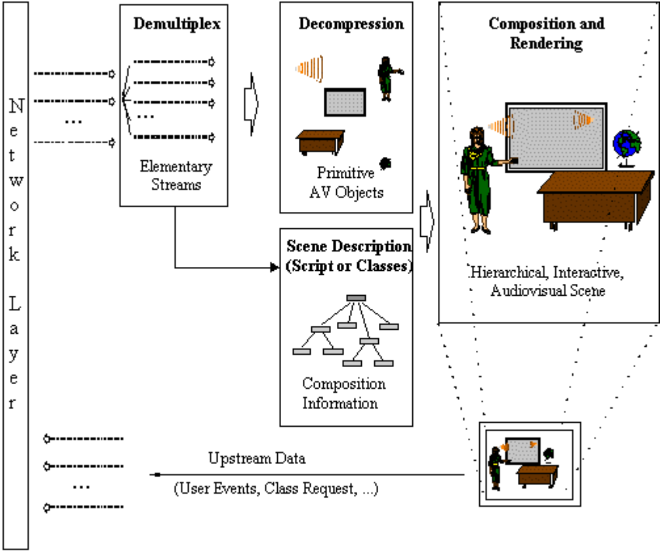
# MPEG–4

- MPEG–4 defines several primitive objects for representing **natural** and **synthetic** information, as well as **2D** and **3D** data.
- The **audiovisual scenes** are composed of these media objects, hierarchically organized:
  - Images (for example, a fixed background).
  - Video objects (for example, a person talking).
  - Audio objects (for example, the voice of the person, background music, . . . ).
  - Text and graphics.
  - Synthetic talking heads and the corresponding text used by the speech synthesizer; animated synthetic bodies.
  - Synthetic sound.
  - . . .

# MPEG–4

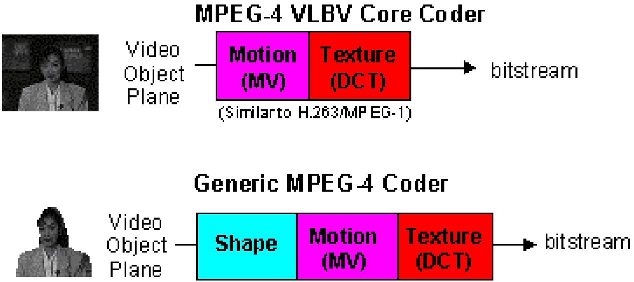

audiovisual objects
voice
audiovisual
presentation
multiplexed
downstream
control / data
sprite
2D background
3D objects
multiplexed
upstream
control/data
y
scene
coordinate
system
x
z
user events
video
compositor
projection
plane
audio
compositor
hypothetical viewer
display
speaker
user input

# MPEG–4


Network Layer
Demultiplex
...
Elementary Streams
Decompression
Primitive AV Objects
Scene Description (Script or Classes)
Composition Information
Composition and Rendering
Hierarchical, Interactive, Audiovisual Scene
Upstream Data
(User Events, Class Request, ...)

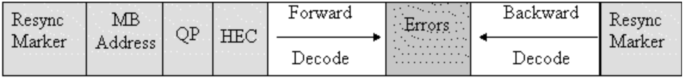
# MPEG–4

- Conventional video coding is performed as in MPEG–1/2.
- In **content-based** coding, it is possible to encode regions with arbitrary shape, but, it this case, the shape of the object also needs to be efficiently represented.
- **Shape** is represented using a 8 bit transparency component or a binary mask.



# MPEG–4

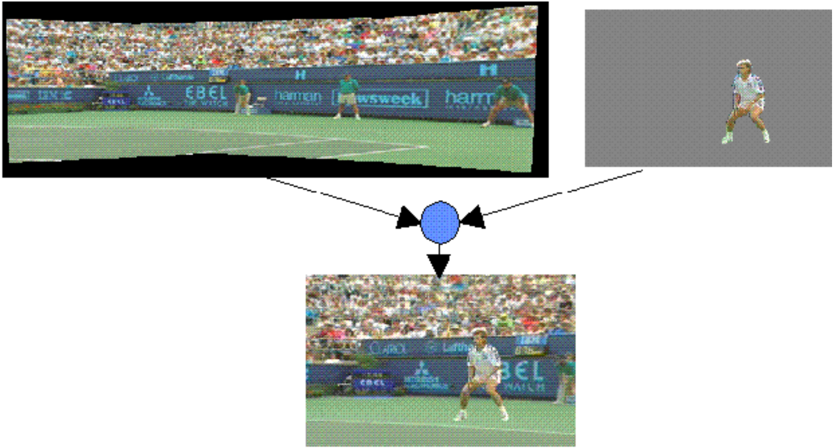
- MPEG–4 provides **error protection** (essential, for example, in wireless transmission), through:
  - Re-synchronization:
    - At the beginning of each GOB.
    - Periodically in the bitstream.
  - Data recovery:
    - Reversible variable length codes (RVLC).




  - Error concealment.
    - Reproduction of the block from the previous frame.

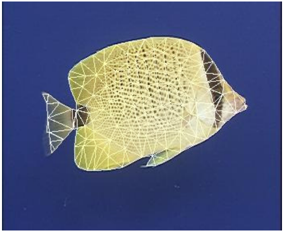
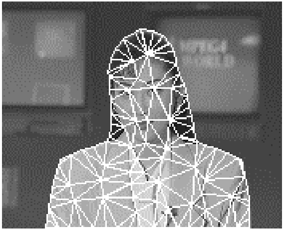
# MPEG–4

- Sprites:



# MPEG–4

- MPEG–4 supports **synthetic visual objects**:
  - Parametric description of **human heads and bodies** (also body animation in Version 2).
  - Parametric description of **static or dynamical meshes** with texture mapping.

  - **Scalable texture coding**.

# H.264/AVC

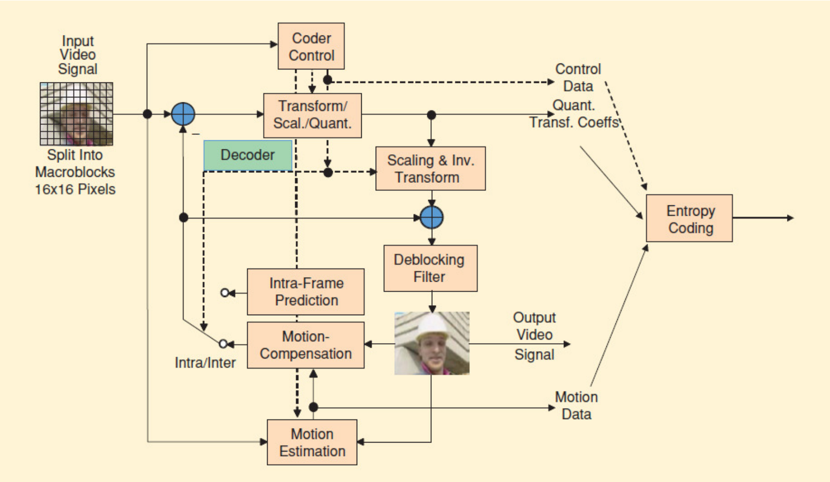
## Overview

- H.264/AVC (Advanced Video Coding) was jointly developed by the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC MPEG.
- It was finalized in March 2003 and approved by the ITU-T in May 2003.
- H.264/AVC provides gains in compression efficiency of up to 50% over a wide range of bit rates and video resolutions compared to previous standards.
- The decoder complexity is about four times that of MPEG-2 and two times that of MPEG-4 Visual Simple Profile.

# H.264/AVC

## Overview

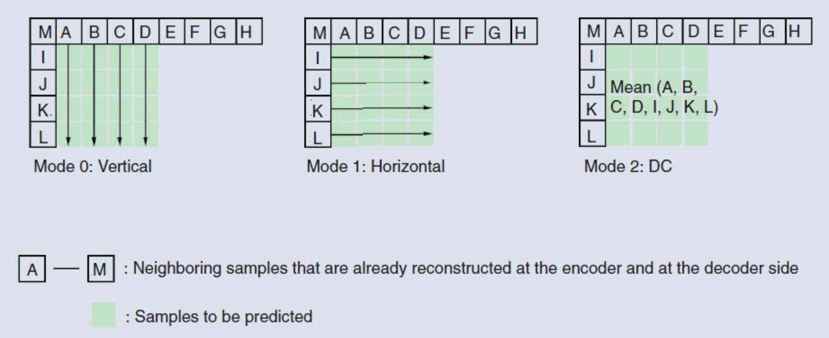
Block diagram of a typical encoding process of H.264/AVC



# H.264/AVC

## Intra prediction

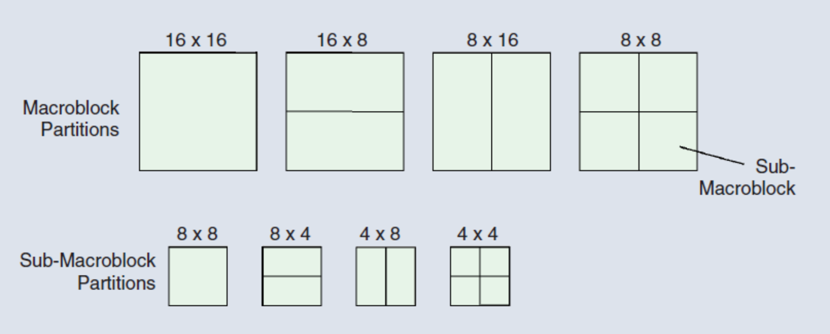
Three out of nine possible intra prediction modes for INTRA_4$\times$4



# H.264/AVC

## Motion-compensated prediction

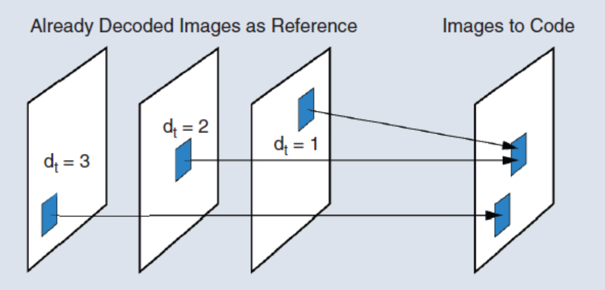
Partition of macroblock/sub-macroblock for motion-compensation



# H.264/AVC

## Motion-compensated prediction

Motion-compensated prediction with multiple reference images



# H.264/AVC

## Transform coding

- Instead of the DCT, three different integer transforms are used:

$$H_1 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \quad H_2 = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 \\ 1 & -1 & 1 & -1 \end{bmatrix} \quad H_3 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

- They are mostly applied to $4\times4$ blocks, but can also be applied to $2\times2$ blocks.
- $H_1$ is applied to all prediction error blocks of $Y$, $C_b$ and $C_r$. If the macroblock is predicted using type INTRA_16$\times$16, then $H_2$ is applied in addition to $H_1$.
- $H_3$ is used for transforming the 4 DC coefficients of each chrominance component.

# H.264/AVC

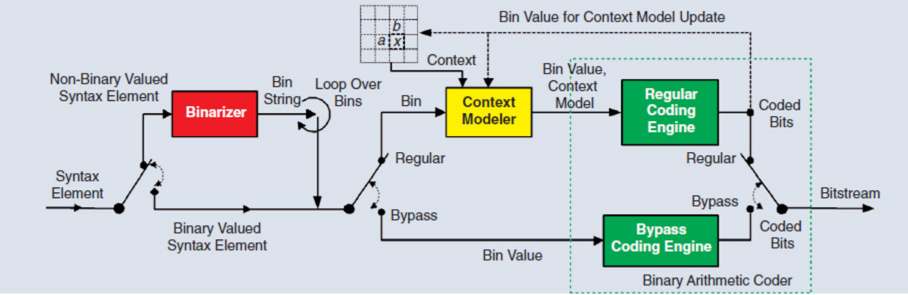
## Entropy coding

- H.264/AVC provides two methods for entropy coding:
    - CAVLC, a low-complexity technique based on context-adaptive sets of variable length codes.
    - CABAC, a context-based adaptive binary arithmetic encoder.
- By incorporating context modeling, both methods offer a high degree of adaptation to the underlying source.
- CAVLC relies on 32 different VLCs. For typical coding conditions, it is 2–7% better than conventional codes.
- Typically, CABAC provides bit rate reductions of 5–15% compared to CAVLC.

# H.264/AVC

## CABAC

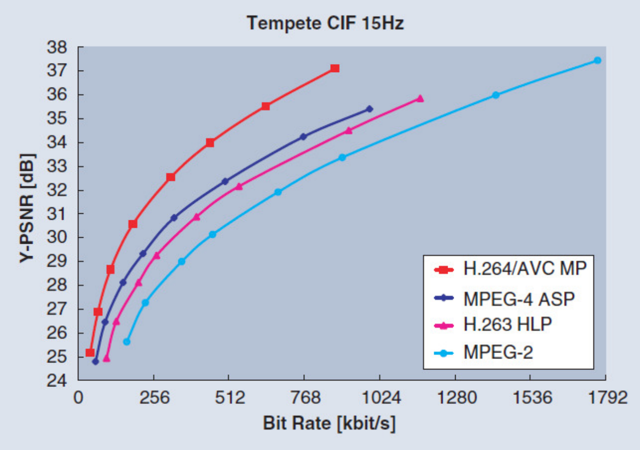
Context-based adaptive binary arithmetic coding



# H.264/AVC

## Performance

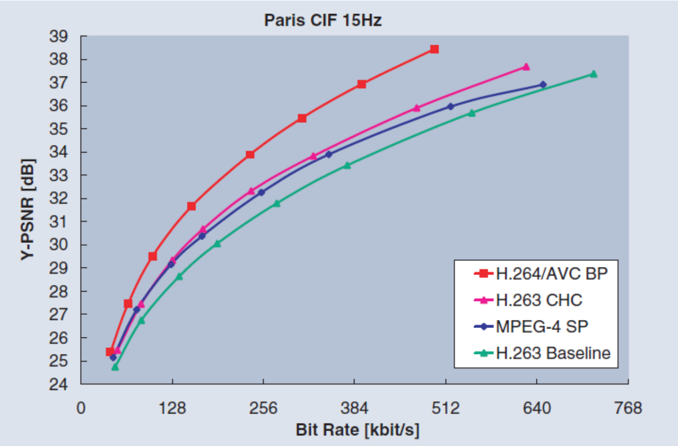
Video streaming application



# H.264/AVC

## Performance

Video conferencing application



# H.265/HEVC

## Overview

- H.265/HEVC (High Efficiency Video Coding) was (again) the result of a collaboration between the ITU-T Video Coding Experts Group (VCEG) and the ISO/IEC MPEG.
- It is also known as the MPEG-H Part 2 and the first version was finalized in 2013.
- H.265/HEVC can provide gains in compression efficiency of about 50%, when compared to H.264/AVC.
- This is mostly attained by further exploring existing techniques, but at a cost of increasing the complexity of the encoder.
- As with H.264/AVC, H.265/HEVC is dependent of a considerable number of patents, which is preventing its wide use. . .

# H.265/HEVC

## Block diagram