# Transform Video Compression

Prof. Ioannis Pitas
Aristotle University of Thessaloniki
pitas@csd.auth.gr
www.aiia.csd.auth.gr

Version 3.1

Artificial Intelligence &
Information Analysis Lab

# Transform Video Compression

- **Video compression**
- Intraframe video coding
- Interframe video coding
- Transform Video Coding
- Predictive coding
- MPEG2
- MPEG4

# Video compression

Video compression facilitates:

- Handling and storage of high resolution video
- Video transmission over computer networks
- TV broadcasting

Application areas:

- Digital television
- Video conferencing
- Video streaming
- Digital Cinema
- Distance learning

# Video compression

Use of inherent ***spatiotemporal video redundancy.***

- If we compress each frame **seperately** (as an image), we only employ spatial redundancy within the frame
- Prediction of current blocks of frames $f(\mathbf{n}, t) = f(n_1, n_2, t)$ from previous (or future) video frame blocks $f(\mathbf{n}, t - l)$ employs temporal redundancy.
- Compression of displaced frame difference (assumed to be small).

Two operation modes:

1. Intraframe coding.
2. Interframe coding.

# Transform Video Compression

- Video compression
- **Intraframe video coding**
- Interframe video coding
- Transform Video Coding
- Predictive coding
- MPEG2
- MPEG4

# Intraframe video coding

In ***intraframe video coding***, Video frame $f(\mathbf{n}, t)$ coding does not take input from video other frames.

- $f(\mathbf{n}, t)$ is transformed using ***Discrete Cosine Transform*** **(*DCT*).**

DCT coefficients are:

- Quantized and
- VLC encoded.

The video frame is:

- compressed and transmitted and received by decoder

The decoder decodes compressed frame and produces $\hat{f}(\mathbf{n}, t)$.

# Intraframe video coding

An ***I-frame*** is a fully intra-encoded video frame.

- They are used periodically to stop decompression error propagation.
- Very useful for quick video browsing.
- First video frame always encoded as an I-frame.

# Transform Video Compression

- Video compression
- Intraframe video coding
- **Interframe video coding**
- Transform Video Coding
- Predictive coding
- MPEG2
- MPEG4

# Interframe video coding

Predicted video frames:

- ***P-frames*** (forward prediction).
  - Prediction from previous video frame.
  - Reference frame encoded using intraframe transformation or prediction from another frame.
- ***B-frames*** (bidirectional prediction).
  - Bidirectional prediction from previous and subsequent video frames.
- Both encoding methods employ block matching.

# Interframe video coding

***Motion estimation***:

- It estimates motion vectors between two video frames.

***Motion compensation***:

- It uses motion vectors $\mathbf{d}_t$ and previous reconstructed frame $\hat{f}(\mathbf{n}, t-1)$
- Produces prediction $p(\mathbf{n}, t)$ for current frame.

# Interframe video coding

***Interframe video coding*** produces a ***prediction error*** between current frame $f(\mathbf{n},t)$ and ***predicted frame*** $p(\mathbf{n},t)$.

***Video encoder***:

- ***Motion vectors*** $\mathbf{d}_t$ and reconstructed video frames $\hat{f}(\mathbf{n},t-l)$ produce a predicted image $p(\mathbf{n},t)$.
- ***Prediction error***:

$$e(\mathbf{n},t)=f(\mathbf{n},t)-p(\mathbf{n},t).$$

is trasformed using DCT.

- DCT coefficients are quantized and VLC encoded.
- Motion vectors $\mathbf{d}_t$ are VLC encoded.

# Interframe video coding

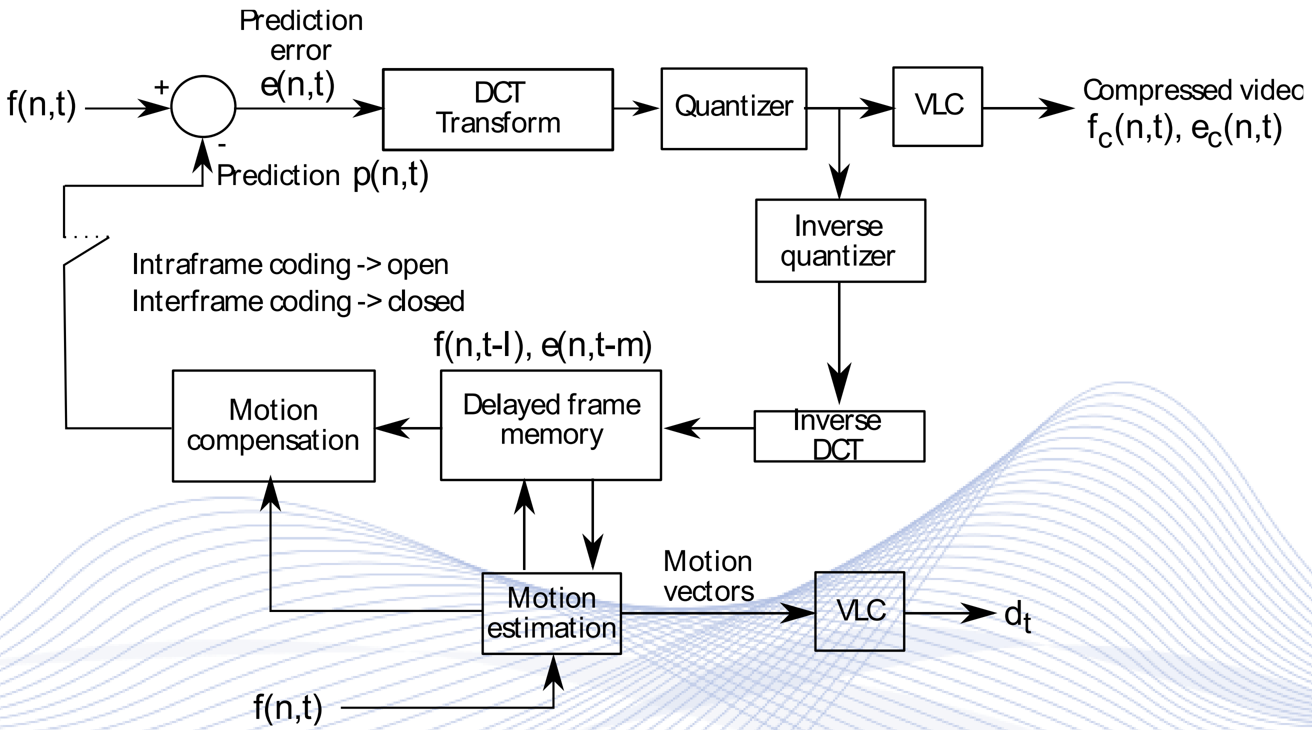
- Encoded DCT coefficients and motion vectors are sent to the decoder.

***Video decoder***:

- Receives DCT coefficients and motion vectors.
- Decodes $e(\mathbf{n}, t)$ producing $\hat{e}(\mathbf{n}, t)$.
- Produces recostructed prediction $\hat{p}(\mathbf{n}, t)$ using reconstructed motion vectors.
- Produces reconstructed video frame:

$$\hat{f}(\mathbf{n}, t) = \hat{p}(\mathbf{n}, t) + \hat{e}(\mathbf{n}, t).$$

# Interframe video coding



General system for Transform Video Compression.

# Interframe video coding

***Forward prediction*** predicts video ***P-frame*** pixel values based on corresponding pixel in a previous frame.

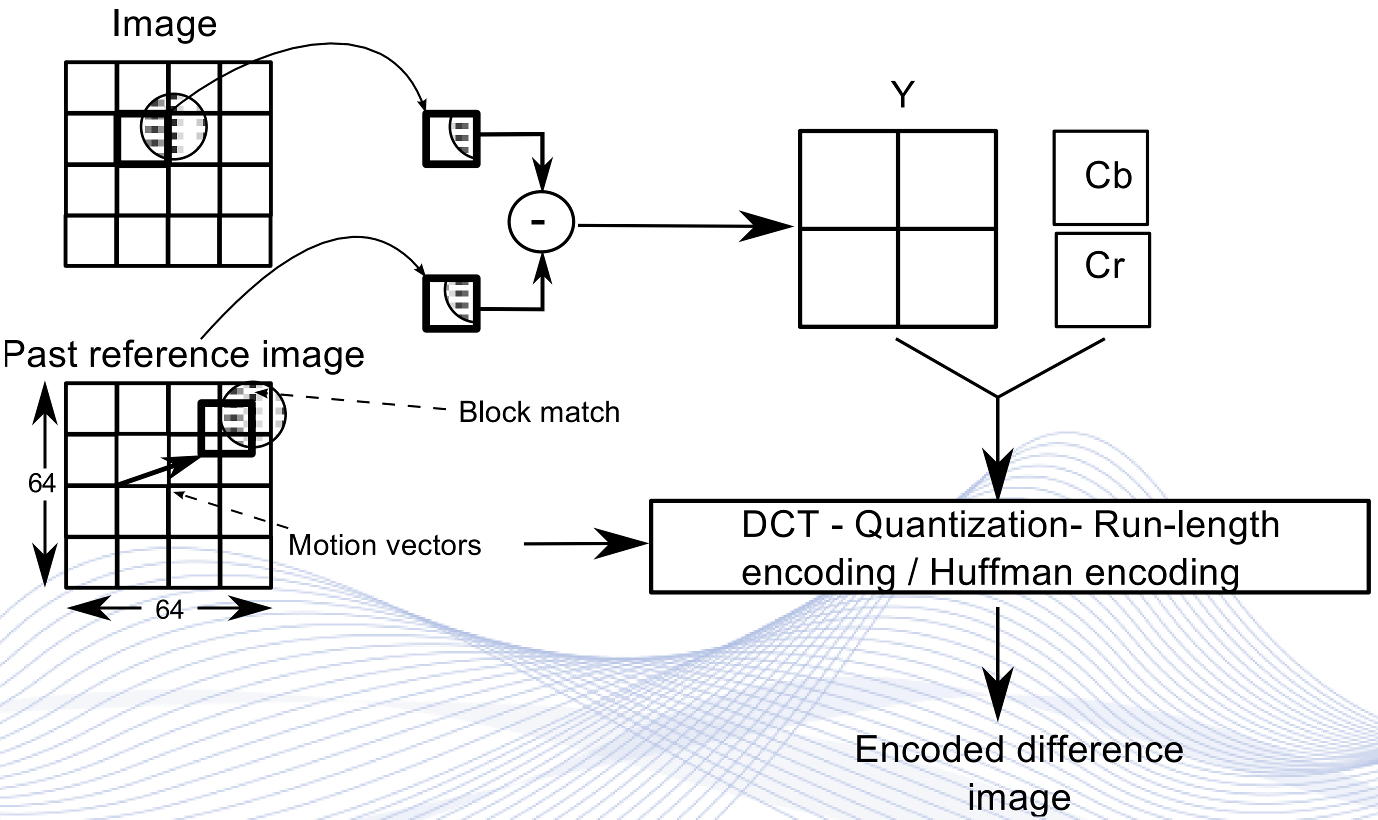
- Object from a video moves between frames - Pixel values change

***Motion compensated prediction***:

$$p(\mathbf{n},t) = f(\mathbf{n}+\mathbf{d}_t, t-1).$$

- $f(\mathbf{n},t)$ : pixel luminance in location $\mathbf{n}=(n_1,n_2)$
- $\mathbf{d}_t=[dx,dy]^T$ : motion vector of a pixel from frame $t-1$ (***reference frame***) to frame $t$ (***predicted frame***).

# Interframe video coding



P-frame prediction.

# Interframe video coding

***Bidirectional Temporal Prediction***:

- Current frame is predicted both from previous and subsequent frames.
- The delayed frame memory (video frame buffer) is employed.
- Predicted video frame $t$:

$$p(\mathbf{n}, t) = a_1 f(\mathbf{n} + \mathbf{d}_t^-, t - 1) + a_2 f(\mathbf{n} + \mathbf{d}_t^+, t + 1).$$

- $\mathbf{d}_t^- = [dx^-, dy^-]^T$: motion vector from frame $t - l$ to frame $t$.
- $\mathbf{d}_t^+ = [dx^+, dy^+]^T$: motion vector from frame $t - m$ to frame $t$.

# Interframe video coding

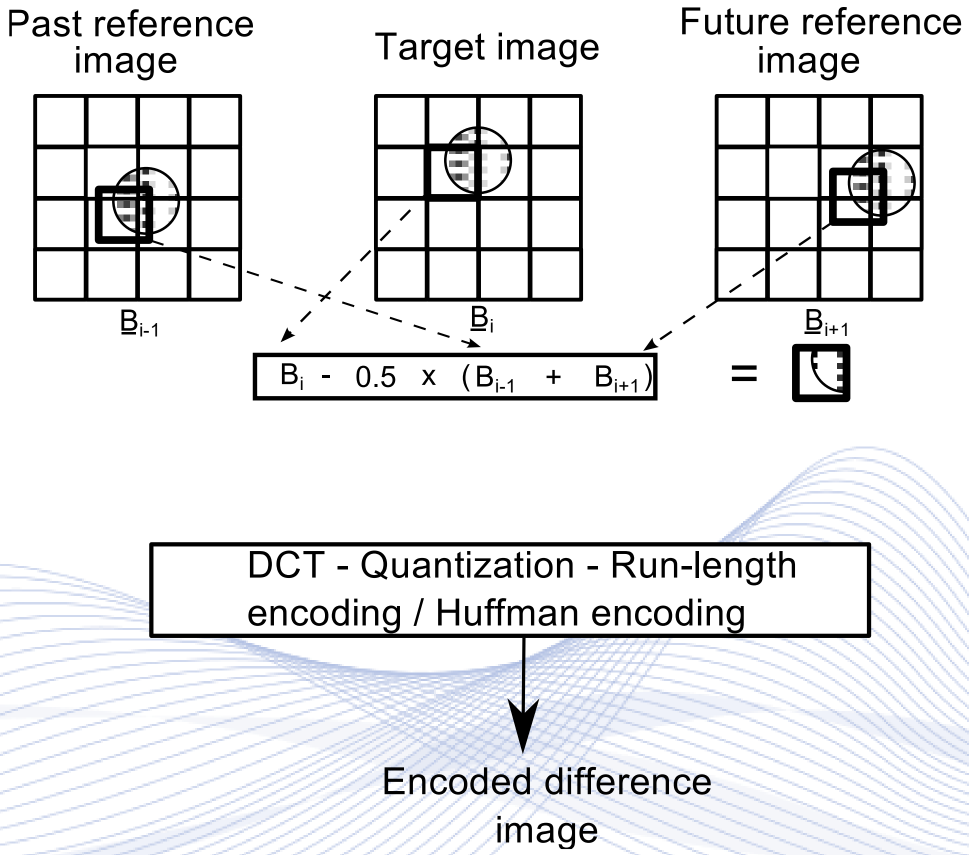
Bidirectional temporal prediction coefficients:

- $a_1 + a_2 = 1$, if mean pixel luminance does not change a lot.
- $a_1 = a_2 = 0.5$ is a good option.

Bidirectional temporal prediction is beneficial if:

- some regions in current frame do not appear in previous frames.
- For example: new objects enter the camera view field.

# Interframe video coding



Past reference image

Target image

Future reference image

$B_{i-1}$

$B_i$

$B_{i+1}$

$B_i - 0.5 \times (B_{i-1} + B_{i+1}) =$

DCT - Quantization - Run-length encoding / Huffman encoding

Encoded difference image

B-frame prediction.

# Interframe video coding

Video encoding:

- DCT, quantization and VLC encoding of error:

$$e(\mathbf{n},t) = f(\mathbf{n},t) - p(\mathbf{n},t).$$

Video decoding:

- Inverse quantization and inverse DCT on the encoded frames $f_c(\mathbf{n},t)$ or encoded errors $e_c(\mathbf{n},t)$.
  - Reconstruction of coded video frames.
  - Storage of decoded video frames to create prediction of subsequent frames.

Artificial Intelligence &
Information Analysis Lab

# Interframe video coding

- ***Delayed video frame memory***:

  - It retains current and previous frames or prediction errors.
  - Necessary for the reconstruction of the prediction error $p(\mathbf{n}, t)$.
  - Number of frames stored depends on the coding algorithm.

# Interframe video coding

- ***Problem***: During interframe transmission, possible coding errors propagate from one video frame to the next one.

- ***Solution***: ***Periodic intraframe video coding.***
  - Transmission errors will spread only till the next intraframe coded video frame.

- Intraframe coding can also used when interframe coding compression does not produce good results:
  - E.g., in case of poor video frame prediction.

# Transform Video Compression

- Video compression
- Intraframe video coding
- Interframe video coding
- **Transform Video Coding**
- Predictive coding
- MPEG2
- MPEG4

# Transform video coding

***Block-based video coding***:

- Video frame content changes in various image regions.
  - Therefore, entire video frame coding is suboptimal.
- In block-based video coding, each video frame is divided into image blocks, e.g., of $8 \times 8$ or $16 \times 16$ pixels.
- Each block is:
  - processed independently;
  - encoded using temporal prediction and transform coding.
- Block matching can be used for motion estimation.

# Transform video coding

There are many 2D linear image transformations:

- DFT, ***DCT***, DST, Haar transform, Hadamard transform, Slant transform.
- They utilize the high special correlation of neighboring pixels.
- They carry image energy to few transform coefficients.
- As image content is not spatially stationary, they are applied to small frame blocks (e.g., of $8 \times 8$ or $16 \times 16$ pixels).

Most common transform for image coding: ***Discrete Cosine Transform*** (***DCT***).

# Transform video coding

***Quantization*** is applied to transform coefficients:

- Lossy compression.
- Significant reduction of bit number.
- Allocated bit number depends on ***Human Visual System*** (**HVS**) characteristics:
    - HVS is more sensitive to low and middle frequencies.
    - Low frequency coefficients: more allocated bits.
    - High frequency coefficients: less allocated bits.
- Use of ***Variable Length Coder*** (***VLC***) on quantizer output.
- Minimization of ***source entropy***.

# Discrete Cosine Transform

***Two-dimensional DCT***:

- DCT expresses a digital signal as of a sum of cosine functions at different frequencies.

- 2D DCT is a separable transformation:

$$C(k_1,k_2)=\sum_{n_1=0}^{N_1-1}\sum_{n_2=0}^{N_2-1}4x(n_1,n_2)\cos\frac{(2n_1+1)k_1\pi}{2N_1}\cos\frac{(2n_2+1)k_2\pi}{2N_2},$$

$0 \leq k_1 \leq N_1-1, 0 \leq k_2 \leq N_2-1.$

Artificial Intelligence & Information Analysis Lab

# Discrete Cosine Transform

- Weight functions $w_1(k_1), w_2(k_2)$:

$$w_1(k_1) = \begin{cases} 1/2, & k_1 = 0 \\ 1, & 1 \leq k_1 \leq N_1 - 1 \end{cases}, \qquad w_2(k_2) = \begin{cases} 1/2, & k_2 = 0 \\ 1, & 1 \leq k_2 \leq N_2 - 1 \end{cases}.$$

- Inverse 2D DCT:

$$x(n_1, n_2) = \frac{1}{N_1 N_2} \sum_{k_1=0}^{N_1-1} \sum_{k_2=0}^{N_2-1} w_1(k_1) w_2(k_2) C(k_1, k_2) \cos \frac{(2n_1+1)k_1\pi}{2N_1} \cos \frac{(2n_2+1)k_2\pi}{2N_2}.$$

# Discrete Cosine Transform

- Both forward and inverse DCT are separable and can be easily calculated along rows and columns:

$$C'(n_1, k_2) = 2\sum_{n_2=0}^{N_2-1} x(n_1, n_2)\cos\frac{(2n_2+1)k_2\pi}{2N_2},$$

$$C(n_1, k_2) = 2\sum_{n_2=0}^{N_2-1} C'(n_1, n_2)\cos\frac{(2n_1+1)k_1\pi}{2N_1}.$$

# Discrete Cosine Transform

- 2D $N_1 \times N_2$ DCT $C(k_1, k_2)$ is related to 2D DFT $F(k_1, k_2)$ of a signal $f(n_1, n_2)$ of size $2N_1 \times 2N_2$:

$$f(n_1,n_2) = \begin{cases} x(n_1,n_2) & 0 \le n_1 \le N_1-1,\ 0 \le n_2 \le N_2-1 \\ x(2N_1-n_1-1,n_2) & N_1 \le n_1 \le 2N_1-1,\ 0 \le n_2 \le N_2-1 \\ x(n_1,2N_1-n_2-1) & 0 \le n_1 \le N_1-1,\ N_2 \le n_2 \le 2N_2-1 \\ x(2N_1-n_1-1,2N_2-n_2-1) & N_1 \le n_1 \le 2N_1-1,\ N_2 \le n_2 \le 2N_2-1 \end{cases},$$

$$F(k_1,k_2) = \sum_{n_1=0}^{2N_1-1} \sum_{n_2=0}^{2N_2-1} f(n_1,n_2) W_{2N_1}^{n_1k_1} W_{2N_2}^{n_2k_2},$$

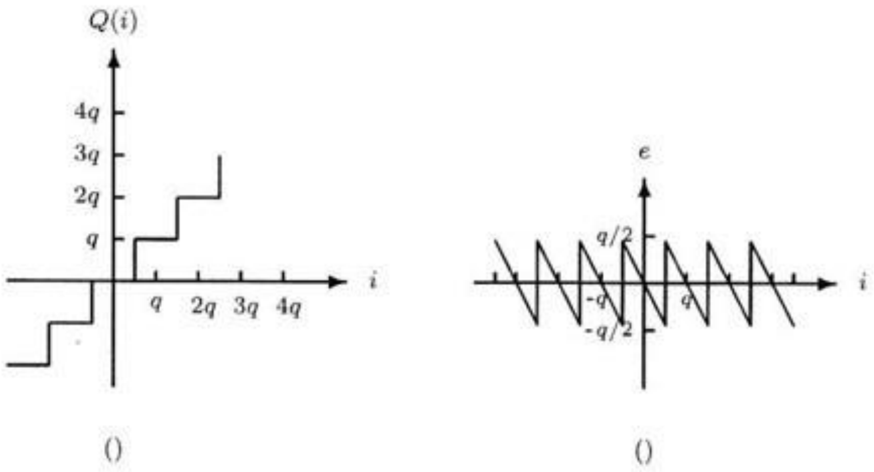
$$C(k_1,k_2) = W_{2N_2}^{k_1/2} W_{2N_2}^{k_2/2} F(k_1,k_2).$$

- A $2N_1 \times 2N_2$ 2D FFT can be used for the fast calculation of an $N_1 \times N_2$ DCT.

Artificial Intelligence & Information Analysis Lab

# DCT Coefficient Quantization

- DCT coefficient quantization results in lossy image/video compression.
- Minimization of mean square error (MSE) between original and quantized coefficients.
- Two ***types*** of quantization:
  - Uniform quantization.
  - Non-uniform quantization.



a) Input-output relation of a uniform quantizer;
b) b) Quantization error.

# DCT Coefficient Quantization

***Uniform Quantizer***:

- Equal quantization intervals for input signal.
- Output values should be much lower than allowable number of input values.
- Smaller mean square error when input follows uniform distribution.

***Non-uniform Quantizer***:

- Used for non-uniform DCT coefficient distribution (e.g., natural images).
- Non-uniform quantization steps.
- Smaller Mean Square Error for non-uniform input distributions.

# DCT Coefficient Quantization

- $N_1 \times N_2$ DCT coefficients form a vector:

$$\mathbf{c} = [C_1, C_2, \ldots, C_N], N = N_1 N_2.$$

- $\mathbf{c}$ is quantized by selecting best matched code-words.

- Mean Square Error (MSE) between original and quantized coefficients:

$$E = \frac{1}{N} E\{\|\mathbf{c} - \hat{\mathbf{c}}\|^2\} = \frac{1}{N} \sum_{k=1}^{N} E\left\{\left(C_k - \hat{C}_k\right)^2\right\}.$$

# DCT coefficient Quantization

- When the number of allocated bits $B_k$ is quite large, it can be shown that:

$$E_k(B_k) = \epsilon_k^2 \sigma_k^2 2^{-2B_k}, \qquad k = 1, \dots, N.$$

- $\sigma_k^2$ : variance of $C_k$ coefficient.
- $\epsilon_k^2$ : parameter that depends on the probability distribution of $C_k$.

Assume we have a mean number of $B$ bits per DCT coefficient:

- How do we allocate $B_N$ bits to $N$ coefficients, so that MSE is minimized?

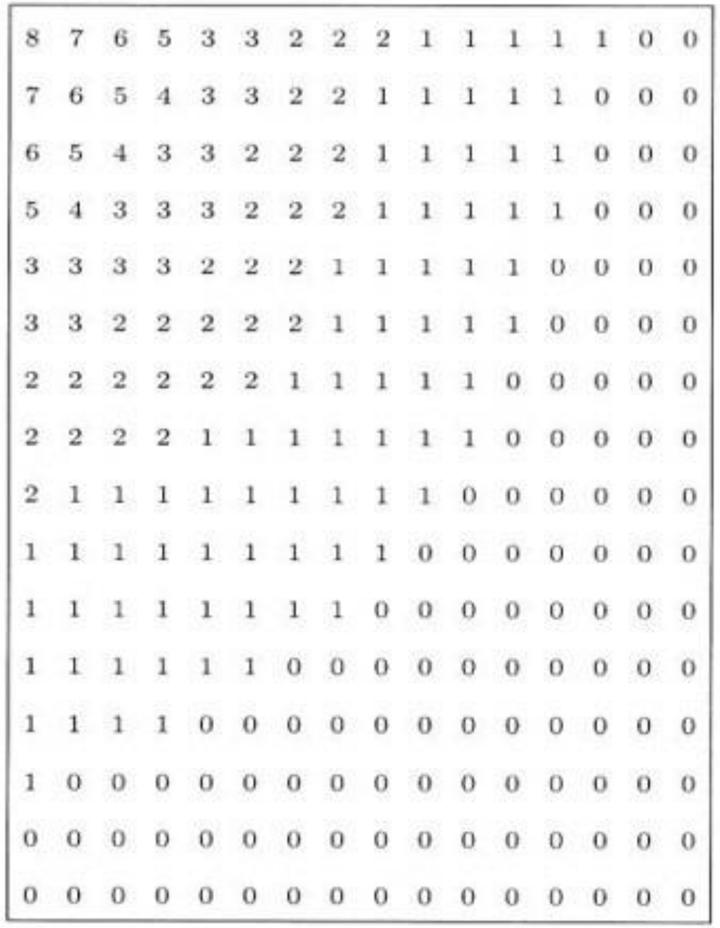
# DCT Coefficient Quantization

- Resulting bit number:

$$B_k = B + \frac{1}{2}\left[\log_2 \epsilon_k^2\sigma_k^2 - \frac{1}{N}\log_2\left(\prod_{k=1}^{N} \epsilon_k^2\sigma_k^2\right)\right]$$

- Corresponding Mean Square Error:

$$E = \left(\prod_{k=1}^{N} \epsilon_k^2\sigma_k^2\right)^{1/N} 2^{-2B_k}.$$



| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 8 | 7 | 6 | 5 | 3 | 3 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 0 | 0 |
| 7 | 6 | 5 | 4 | 3 | 3 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 6 | 5 | 4 | 3 | 3 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 5 | 4 | 3 | 3 | 3 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 |
| 3 | 3 | 3 | 3 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 3 | 3 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 |
| 2 | 2 | 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 2 | 2 | 2 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 |
| 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 1 | 1 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 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 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

Bit allocation to $16 \times 16$ DCT coefficients.

# DCT coefficient Quantization

- Large DCT coefficient variance: – More bits are required.

- Small DCT coefficient variance: – Less bits are required.

- Optimal bit allocation :
- More bits to low frequency DCT coefficients close to DC term. They contain most of the low-frequency content.

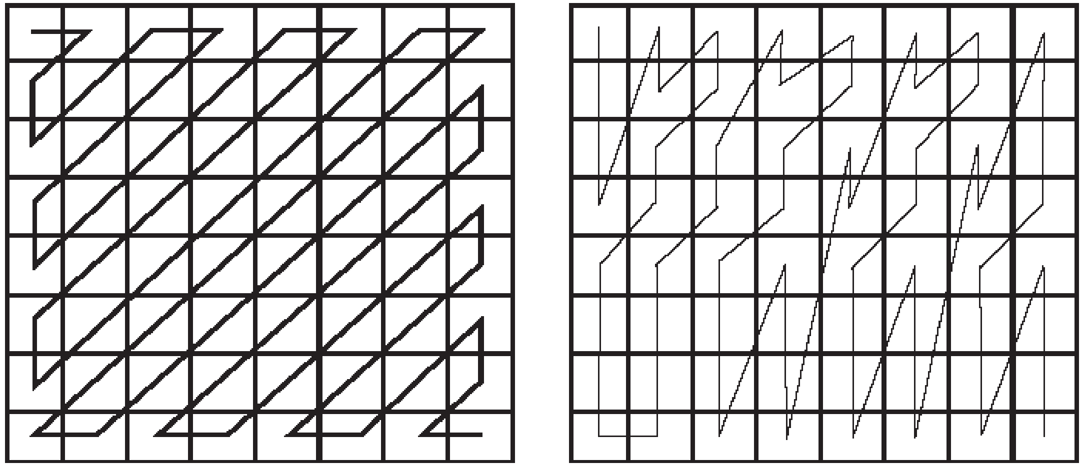
- High-frequency coefficients are zeroed.

# Quantized DCT Coefficient Compression

Quantized DCT coefficients are scanned to form an one-dimensional vector.

- ***Zig-zag scanning***:
  - It is used for video frames where low-frequency DCT coefficients have  equal importance along horizontal and vertical direction.
- ***Alternate scanning***:
  - For interlaced video.
  - Higher frequency content in the vertical direction.

# Quantized DCT Coefficient Compression



DCT coefficient scanning: a) Zig-zag scanning; b) Alternate scanning.

# Quantized DCT Coefficient Compression

- Appropriate scanning method must be chosen.
- Low frequency coefficients are scanned before high frequency coefficients.
- As many DCT coefficients are zeroed, it is inefficient to code them separately:
  - Use ***Run-Length Encoding*** (***RLE***) instead.

# Quantized DCT Coefficient Compression

- RLE starts from the DC coefficient.
- It produces a coding-symbol pair (two numbers):
- a) number of zeros, until next non-zero coefficient and
- b) new non-zero coefficient value.

- ***End-Of-Block*** (***EOB***) symbol is placed after last non-zero coefficient.
- DC and other RLE symbols are encoded using Huffman or arithmetic coding.

# Quantized DCT Coefficient Compression

Example: if the DCT coefficients of a $4 \times 4$ block are:

$$[9300002000010000]$$

the run-length code is:

$$9, 3, (4,0), 2, (3,0), 1, (4,0).$$

Artificial Intelligence &
Information Analysis Lab

# Quantized DCT Coefficient Compression

***Huffman encoding***:

It is an ***entropy coding*** method. Main characteristics:

- It produces variable length codewords.
- It allocates ewer bits to frequent symbols  and more bits to rare symbols.
- For images/video frames, pixel luminance/chrominance values are encoded, based on their probability distribution $p(i)\,, 0 \leq i \leq 2^B - 1$ ($B$: number of bits).
- Different codeword lengths to each luminance level.
- Reduced average codeword length.

# Quantized DCT Coefficient Compression

***Average code-word length***:

$$\bar{L} = \sum_{i=0}^{l-1} L(i) p(i).$$

- $L(i)$ should be chosen so that $\bar{L}$ is minimized.
- Lower limit of $\bar{L}$:

$$\bar{L} \geq H(B).$$

- $H(B)$: symbol entropy:

$$H(B) = -\sum_{i=0}^{l-1} p(i) \log_2 p(i).$$

- When $p(i)$ is a uniform distribution, entropy is maximized.

# Quantized DCT Coefficient Compression

Huffman encoding employs a tree structure.

- Number of Huffman tree leaves is equal to the number luminance values.
- Huffman tree is created in $B$ steps.

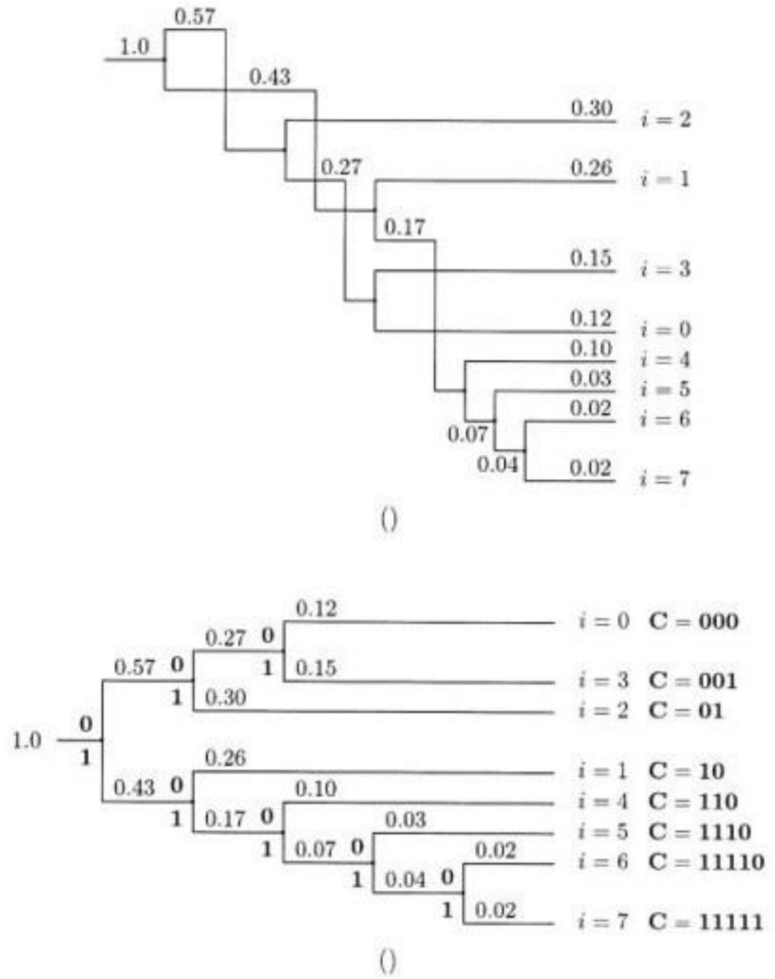
# Quantized DCT Coefficient Compression

***Huffman encoding*** process:

1. Pick the two child nodes with the smallest probabilities.
2. Form a parent node, whose probability is the sum of its children node probabilities.
3. Repeat process until all luminance levels (symbols) are used.
4. Root node probability  should be 1.
5. Rearrange tree branches to disentangle them.
6. Assign 0/1, when traversing tree from root to leaves (upwards/downwords).
7. The codeword of  each luminance value consists of ones and zeros in the path from the tree root to the corresponding leaf.

# Quantized DCT Coefficient Compression

- Image with 8 luminance levels.
- 3 bits/pixel required for PCM encoding ($B = 3$).
- $p(i), \ i = 0, \dots, 2^{B-1}$ known probabilities.



a) Huffman tree; b) Tree re-arrangement.

# Quantized DCT Coefficient Compression

- Only the most frequent symbols are Huffman encoded.
- Large codetables are avoided.
- Rarely applied to raw images – Low compression levels.
- Usually combined with transform coding:
  - Quantized DCT coefficients and run lengths are Huffman encoded.

# Quantized DCT Coefficient Compression

***Arithmetic encoding***:

- It addresses inherent weaknesses in Huffman encoding as they fail, when symbol occurrence is above a certain level, e.g., $p(i) > 0.5$.

- Closest to optimal encoding performance.

- Encodes entire input sequence (not just each symbol) into a number (fraction) $f$, $0 \leq f \leq 1$.

# Quantized DCT Coefficient Compression

Arithmetic encoding algorithm:

1. Each symbol assigned to an interval starting from interval $[0, \ldots, 1]$.
2. Each interval is divided to subintervals, whose span is proportional to current symbol probability.
3. Subinterval of a coded symbol is chosen as interval of next symbol.
4. Arithmetic encoder output is the interval of the last symbol.

# Quantized DCT Coefficient Compression

Arithmetic coding properties:

- Better compression than Huffman,  but slower computation.
- The entire message must be available for decoding.
- One encoded bit error may fail entire message decoding.

# Transform Video Compression

- Video compression
- Intraframe video coding
- Interframe video coding
- Transform Video Coding
- **Predictive coding**
- MPEG2
- MPEG4

# Predictive Coding

Predictive coding can be used to remove information redundancy from:

- DCT coefficients.
- Motion vector of a video frame block.

Value $f(n)$ prediction from its prediction window $\mathcal{A}$:

$$\hat{f}(n) = L[f(n-i), i \in \mathcal{A}, i \neq 0].$$

- $f(n)$, $n = 1,2,\ldots$: DCT coefficient value or one component of motion vector $\mathbf{d}_t = [dx, dy]^T$.
- $f(n-i)$, $i \in \mathcal{A}$: values in prediction window $\mathcal{A}$.
- operator $L$ is usually a linear function.

# Predictive Coding

- If prediction window $\mathcal{A}$ scans the frame blocks row-wise, the prediction $\hat{f}(n)$ is causal and can be based on already reconstructed past values $f_r(n-i)$:

$$\hat{f}(n) = L[f_r(n-i), i \in \mathcal{A}].$$

- Prediction error:

$$e(n) = f(n) - \hat{f}(n).$$

# Predictive Coding

- Signal $f(n)$ is reconstructed as follows:

$$f_r(n) = L[f_r(n-i), i \in A] + e_q(n).$$

- $e_q(n)$: quantized prediction error.

- Good prediction produces small error and results in better compression.

# Predictive Coding

- Its performance depends on prediction operator $L$ window and on optimal prediction coefficients.
- 1D signal $f(n)$ is modeled as stationary AR process of order $p$:

$$f(n) = \sum_{k=1}^{p} a(k) f(n-k) + \epsilon(n), \qquad E[\epsilon^2(n)] = \sigma^2.$$

$\epsilon(n)$: white Gaussian noise that is uncorrelated to $f(n)$.

# Predictive Coding

Linear prediction $L$:

$$\hat{f}(n) = \sum_{k=1}^{p} a(k) f_r(n-k)\,.$$

- $\boldsymbol{a} = [a(1), \dots, a(p)]^T$: parameter vector.
- ***Differential Pulse Code Modulation*** (***DPCM***) for $p = 1$:

$$f(n) = af(n-1) + \epsilon(n).$$

- Typically, $a = 1$.

Artificial Intelligence & Information Analysis Lab

# Predictive Coding

- Prediction window $\mathcal{A}$ contains reconstructed values: $\{f_r(n-1), \dots, f_r(n-p)\}$.

- Quantized error that is transmitted to the receiver:

$$e_q(n) = Q[e(n)] = Q\left[f(n) - \hat{f}(n)\right].$$

- Reconstructed values $f_r(n)$ at the receiver:

$$f_r(n) = \sum_{k=1}^{p} a(k) f_r(n-k) + e_q(n).$$

# Predictive Coding

- Optimal prediction coefficient estimation:

$$\begin{bmatrix} R(0) & R(1) & \cdots & R(p-1) \\ R(1) & R(0) & \cdots & R(p-2) \\ \vdots & \vdots & \vdots & \vdots \\ R(p-1) & R(p-2) & \cdots & R(0) \end{bmatrix} \begin{bmatrix} a(1) \\ a(2) \\ \vdots \\ a(p) \end{bmatrix} = \begin{bmatrix} R(1) \\ R(2) \\ \vdots \\ R(p) \end{bmatrix}.$$

- Circulant $p \times p$ matrix.

# Predictive Coding

- For ***block DC term*** or ***motion vector***, first order prediction is used ($p = 1$):

$$f(n) = af(n-1) + e(n).$$

- The difference of current block DC term or motion vector components $(dx, dy)$ from those of previous block within a group of blocks (GOP) are transmitted to the receiver.

# Transform Video Compression

- Video compression
- Intraframe video coding
- Interframe video coding
- Transform Video Coding
- Predictive coding
- **MPEG2**
- MPEG4

# MPEG-2 Standard

The MPEG-2 standard describes a combination of lossy video compression and lossy audio data compression methods:

- It is used as a digital television signal format for terrestrial (over-the-air), cable or satellite TV broadcasting.
- It is also used as a digital cinema and video streaming/storage format.
- It is compatible with the MPEG-1 standard.

- It serves many applications of various video rates (2-20 Mbps) and resolutions.

# MPEG-2 Standard

Main MPEG-2 features:

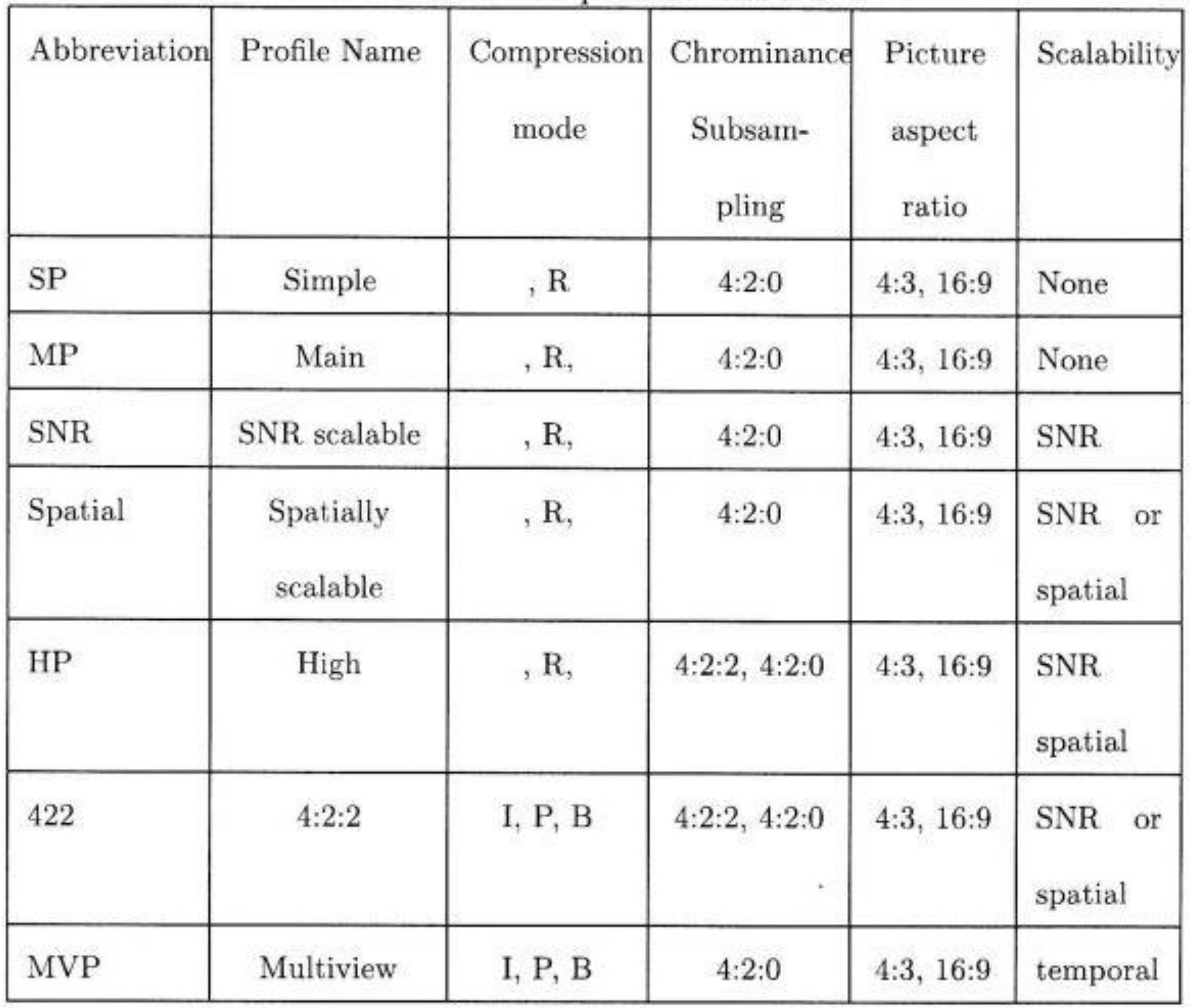
- It supports progressive, interlaced video.
- Scalable bit flow enables MPEG-2 to encode video at different resolutions and/or quality, including high-definition video.
- It can encode ***Standard Definition Television*** (***SDTV***) video at bit rates from about 3-15 Mbps and ***High Definition Television*** (***HDTV***) video at 15-30 Mbps.

# MPEG-2 Standard

- It provides improved quantization and coding options.
- It improves subjective video quality by obtaining higher peak signal to noise ratio.
- It supports chromaticity subsampling (4:4:4,4:2:0, 4:2:2).
- It has several ***MPEG-2 profiles***.

# MPEG-2 Standard

Table 13.9.1: The profiles MPEG-2

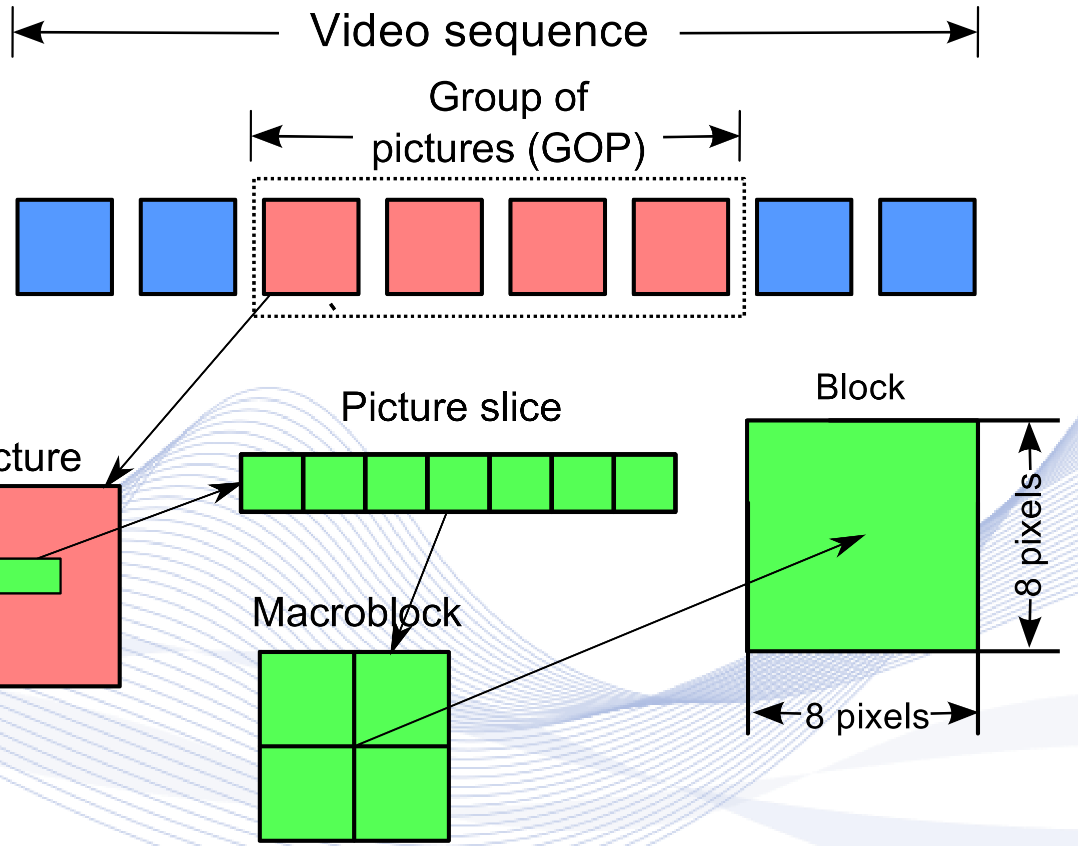


| Abbreviation | Profile Name | Compression mode | Chrominance Subsampling | Picture aspect ratio | Scalability |
|---|---|---|---|---|---|
| SP | Simple | , R | 4:2:0 | 4:3, 16:9 | None |
| MP | Main | , R, | 4:2:0 | 4:3, 16:9 | None |
| SNR | SNR scalable | , R, | 4:2:0 | 4:3, 16:9 | SNR |
| Spatial | Spatially scalable | , R, | 4:2:0 | 4:3, 16:9 | SNR or spatial |
| HP | High | , R, | 4:2:2, 4:2:0 | 4:3, 16:9 | SNR spatial |
| 422 | 4:2:2 | I, P, B | 4:2:2, 4:2:0 | 4:3, 16:9 | SNR or spatial |
| MVP | Multiview | I, P, B | 4:2:0 | 4:3, 16:9 | temporal |

# MPEG-2 Standard

***Macroblock***:

- Four 8 × 8 luminance blocks.
- Motion estimation performed at Macroblock level.
- Resulting motion vector used in its 4 constituent image blocks.



MPEG2 stream structure.

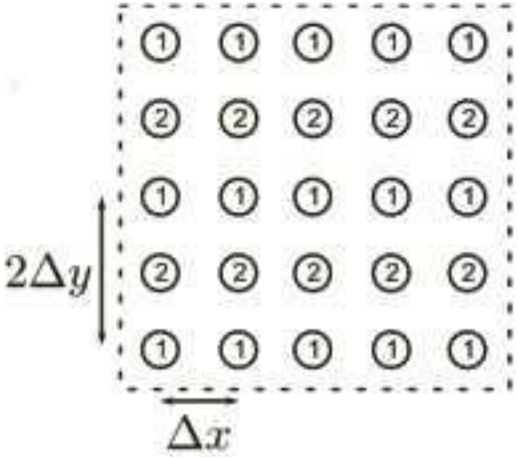
# MPEG-2 Standard

- MPEG-2 supports $YC_bC_r$ color system (for EU).
- ***Chromaticity subsampling*** is the reduction of $C_b, C_r$ chromaticity resolution, by color component subsampling.
- There are 3 types of chromatinance subsampling:
  - 4:2:0 (as in MPEG-1, 1/4 the Luma Samples);
  - 4:2:2 (subsmapling in horizontal direction, 1/2 the Luma Samples);
  - 4:4:4 (no subsampling, No Color Reduction).
- Total number of $8 \times 8$ blocks per MB:

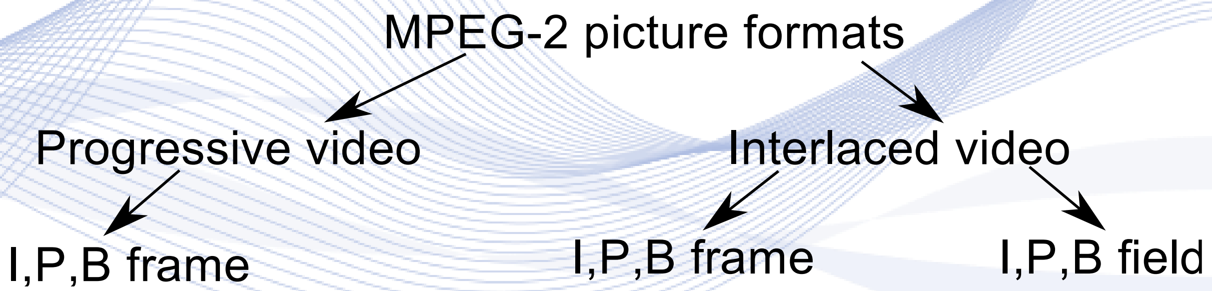
  6 $(4\,Y, 1\,C_b, 1\,C_r)$, 8 $(4\,Y, 2\,C_b, 2\,C_r)$, 12 $(4\,Y, 4\,C_b, 4\,C_r)$.

# MPEG-2 Standard

- Two interlaced video picture types:
- ***Frames pictures***: obtained by deinterlacing even and odd-numbered fields (I-, P- or B-type).
- ***Field pictures***: even and odd-numbered fields as seperate images (I-, P- or B- type).
- Support of both interlaced and progressive video.



Interlaced luminance video.


MPEG-2 picture formats

Progressive video

I,P,B frame

Interlaced video

I,P,B frame

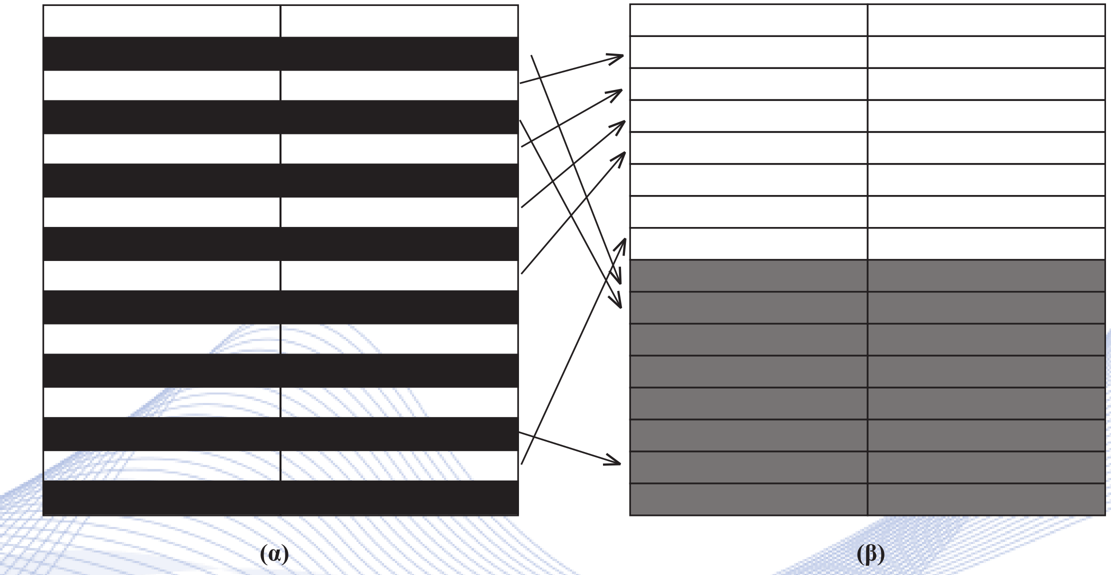
I,P,B field

MPEG-2 picture formats.

# MPEG-2 Standard

Two encoding options:

- ***Field encoding***: Each field block is encoded independently, if significant motion is present.
- ***Frame encoding***: Two fields encoded together as frame picture (better for static video content).



a) Frame DCT; b) Field DCT.

# MPEG-2 Standard

- Option to choose between field DCT or frame DCT for each MB.

- Field or frame motion compensated prediction for interlaced video.
  - Choice dependending on motion presence/absence in MB.

# MPEG-2 Standard

- ***Single field prediction***:
  - Each video field is predicted independently from data of previously decoded fields.
  - Field pictures use only field prediction.

- **Single frame prediction**:
  - Each video frame is predicted from previously decoded frames.
  - Frame pictures use frame or field prediction (depending on MB content).

# MPEG-2 Standard

- Support of two different DCT coefficient scanning methods:
  - Zigzag scan
  - Alternative scan.

Compression types:

- Intraframe coding
  - 11 bits for DC coefficient;
  - AC coefficients are quantized in  range [-2048, 2047].

- Interframe coding
  - All coefficients are quantized in range [-2048, 2047].

# MPEG-2 Standard

***Scalability*** is the possibility of decoding certain part of a video bit stream for obtaining desired video resolution:

Support of different decoders to display video at different spatio-temporal analysis of same bit flow:

- ***Base level***: decoding of minimal bit stream subset.
- ***Augmented decoding levels***: improve video quality vs base level.

- Scalability advantage: robustness to transmission errors.

# Transform Video Compression

- Video compression
- Intraframe video coding
- Interframe video coding
- Transform Video Coding
- Predictive coding
- MPEG2
- **MPEG4**

# MPEG-4 AVC

***MPEG-4 Advanced Video Coding*** (***AVC***):

also called H.264 or MPEG-4 Part 10, it is a video compression standard based on block-oriented, motion-compensated and DCT coding.

- It was developed by ITU-T Video Coding Experts Group (VCEG) with the ISO/IEC Moving Picture Experts Group (MPEG).
- It has different philosophy from MPEG-2 motion compensation sections.
- Good quality at lower transmission rates.
- Design to avoid increased implementation complexity/cost.

# MPEG-4 AVC

Support of a wide variety of networks and systems:

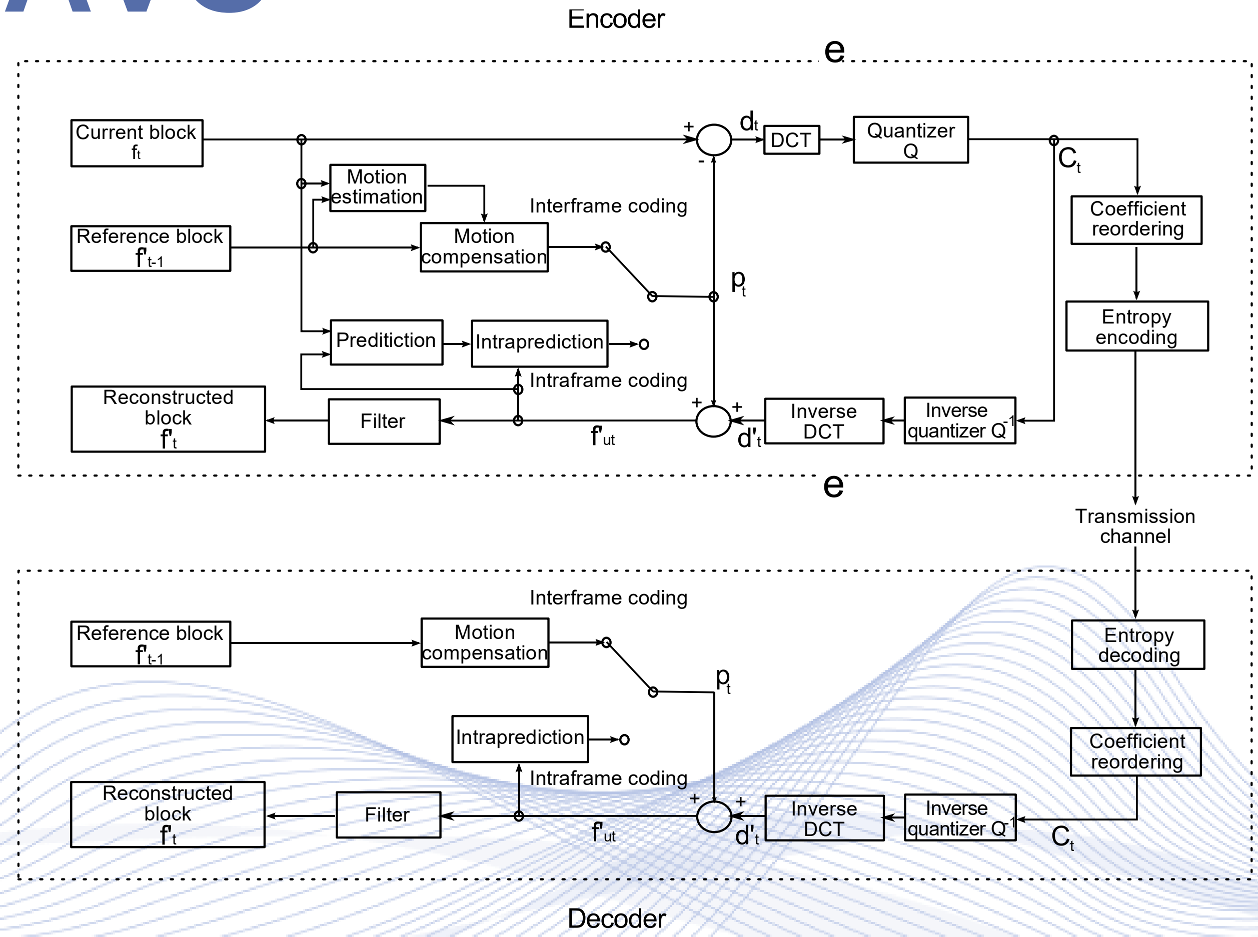
- Low and high bit rates;
- Low and high resolution video;
- Broadcasting;
- DVD storage;
- Video streaming (RTP/IP packet networks);
- ITU-T multimedia telephony systems.

# MPEG-4 AVC

MPEG-4 AVC does not explicitly define a unique video ***Codec*** (***Encoder/Decoder***).

- A video frame split into Macroblocks.
- Macroblocks are encoded in inter or intra mode.
- For each image block $f_t$ in a macroblock, a prediction $p_t$ is created, based on reconstructed image samples.

# MPEG-4 AVC



e

MPEG-4 AVC structure.

# MPEG-4 AVC

***Intra coding mode***: Predictions are created from previously encoded and decoded image samples $f'_{ut}$ the same Slice.

***Intra coding mode*** (***Transcoding***): Predictions generated through motion estimation and compensation (motion-compensated prediction) from one or more selected reference blocks $f'_{t-1}$:

- Prediction $p_t$ is subtracted from current block to produce one difference block $e_t$ to be encoded (DCT transform, quantization, VLC encoding).
- If prediction $p_t$ is added to decoded difference $e'_t$, it generates a decoded version $f'_{ut}$, of the initial section $f_t$.

# MPEG-4 AVC

- MPEG-4 AVC decoder:
  - Receives a compressed stream bit.
  - Performs VLC decoding, to create a set of quantized coefficients $C_t$.
- After DCT coefficient scaling and inverse DCT, $e_t'$ is produced.

- Using stream header data, the decoder creates a prediction $p_t$ identical with that of the encoder.
- The prediction $p_t$ is added to $e_t'$, to create $f_{ut}'$, which is filtered to give the decoded block $f_t'$.

# MPEG-4 AVC

MPEG-4 AVC contains 17 profiles.

- Main profile supports:
  - Intraframe and interframe using I-, P- and B- (I-slices, P-slices, B-slices)
  - Weighted prediction results in increased flexibility in prediction sections with motion compensation.
  - Entropy coding using ***Context-Adaptive Binary Arithmetic Coding*** (**CABAC**).

# MPEG-4 AVC

- Progressive chrominance video sampling is 4:2:0.
- Other profiles use: 4:1:1 (1/4 of the Luma Samples), 4:2:2 (1/2 the Luma Samples), 4:4:4 and 4:4:4:4 (No Color Reduction).
- Each B slice in each inter-encoded macroblock can be predicted from one or two reference images, before or after current image.
- 3 B-prediction options:
  - a) from a past and a future image; 2
  - b) from 2 past images;
  - c) from 2 future images.

# MPEG-4 AVC

***Weighted Prediction Method*** weights the contribution of the prediction data obtained from motion compensation.

There are three WPM prediction types:

- Explicit weighted prediction in P-macroblock.
- Explicit weighted prediction in B-macroblock.
- Implicit weighted prediction in B-macroblock.

***Explicit weighted prediction*** weights are determined by the encoder and are transmitted in the Slice header.

***Implicit weighted prediction*** weights are calculated from relative temporal position of reference image.

# Bibliography

[PIT2017] I. Pitas, “Digital video processing and analysis”, China Machine Press, 2017 (in Chinese).

[PIT2013] I. Pitas, “Digital Video and Television”, Createspace/Amazon, 2013.

[WAN2002] Wang Y, Ostermann J, Zhang YQ. Video processing and communications, Prentice Hall, 2002.

[PIT2000] I. Pitas, Digital Image Processing Algorithms and Applications, J. Wiley, 2000.

# Q & A

**Thank you very much for your attention!**

**More material in**
**http://icarus.csd.auth.gr/cvml-web-lecture-series/**

**Contact: Prof. I. Pitas**
**pitas@csd.auth.gr**