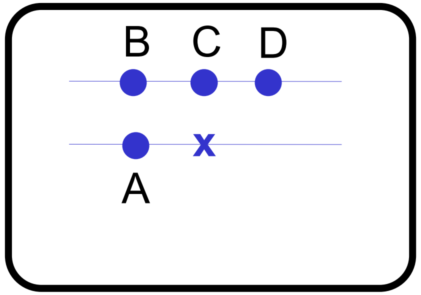
Methods and standards for image and video signal compression

Compression methods

- compression
  - analog or digital signal processing to reduce or eliminate redundancy
  - subjective redundancy
    - determined by the properties of the human visual system
    - the human eye is more sensitive to low spatial frequencies than to high ones (recall the shape of the contrast sensitivity function)
  - spatial redundancy
    - result of correlation (similarity) between the elements of an image
  - tempotal redundancy
    - result of correlation between successive frames in video sequence
  - statistical redundancy
    - result of correlation in a sequence of code characters
- the consequence of removing redundancy is bandwidth reduction for signal transmission, bit-rate reduction and a reduction in the space required for signal storage

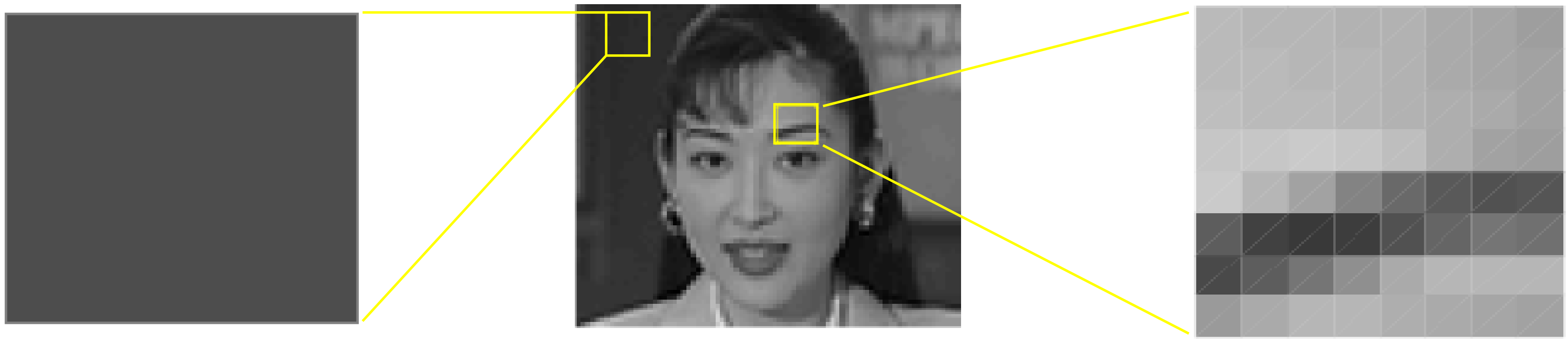
Compression methods

- spatial redundancy
  - exists due to the correlation of nearby picture elements within a single image (x is correlated with A, B, C and D)



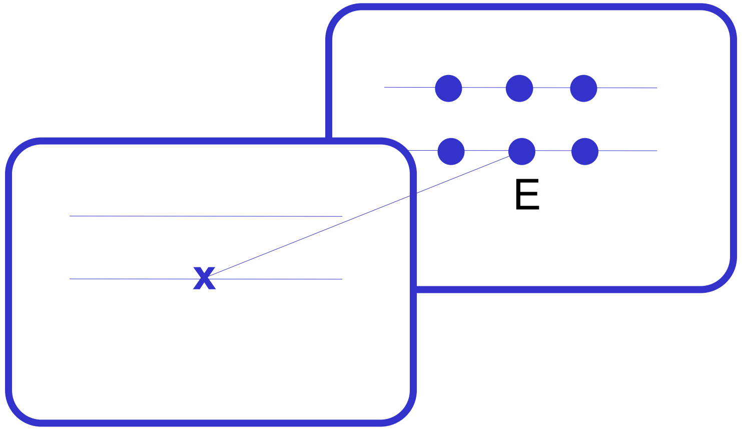
Spatial correlation

  - neighboring picture elements in the image have similar values



Compression methods

- temporal redundancy
  - exists due to the temporal correlation between the picture elements of consecutive frames (sample x from the current frame is similar to sample E at the corresponding position in the previous frame)
  - neighboring frames in video sequnce are similar

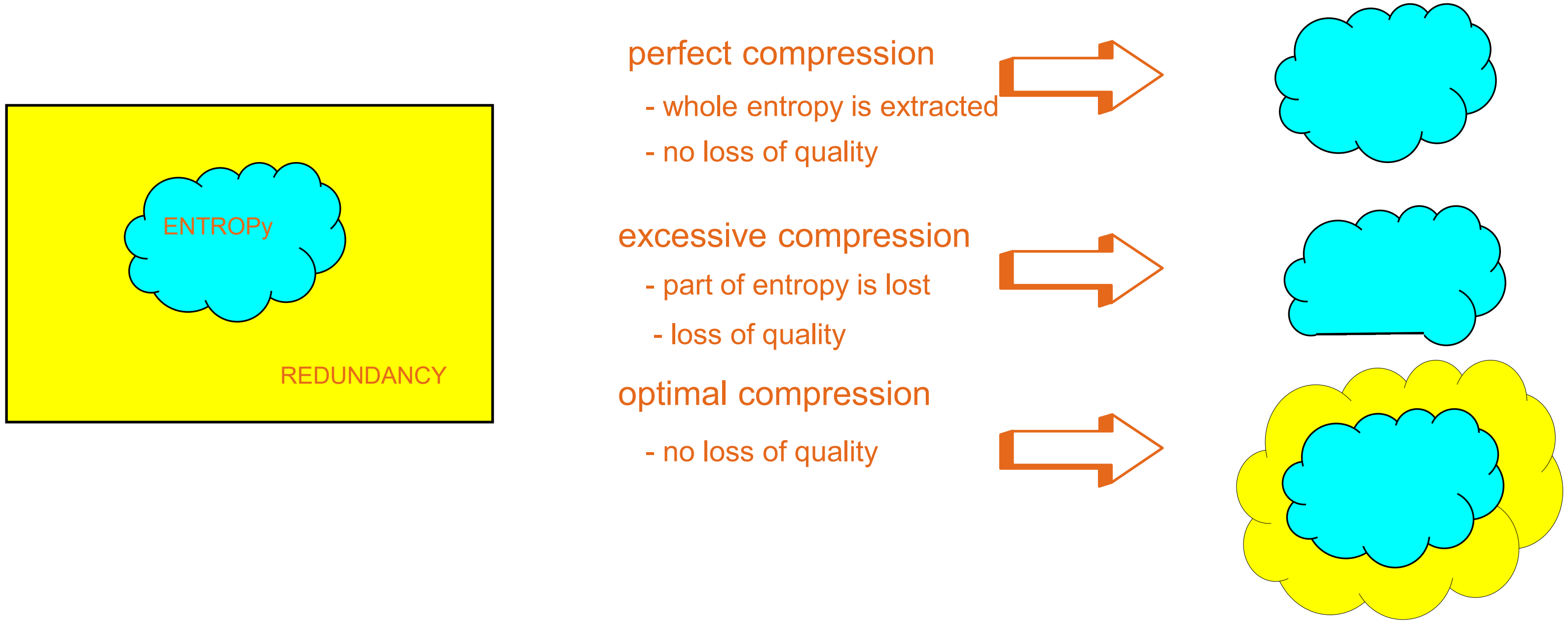


Temporal correlation


Frame 1 Frame 2 Frame 3

Compression methods

- optimal compression or lossless compression
  - the compression device extracts the entire entropy in addition to part of the redundancy
- excessive compression or lossy compression
  - the compression device removes part of the entropy
- perfect compression
  - the compression device extracts only entropy
  - in practice, such a device would be very complex and require a long processing time



Compression methods

- when compressing an image and a video signal, it is necessary to achieve an optimal relationship between the compression ratio, image quality and processing time
  - compression ratio (CR)
    - the ratio of the total number of bits for image coding before and after the compression (eg 40: 1)
    - the average number of bits for coding the image sample after compression (eg 0.2 bpp) obtained as the ratio of the number of bits for coding the image sample before compression and compression ratio (8 bpp / 40 = 0.2 bpp)
  - image quality
    - Mean Square Error (MSE)

$$MSE = \frac{1}{N} \cdot \sum_{i=0}^{N-1} \left(x_i - x_i' \right)^2$$

    - Peak Signal-to-Noise Ratio (PSNR)

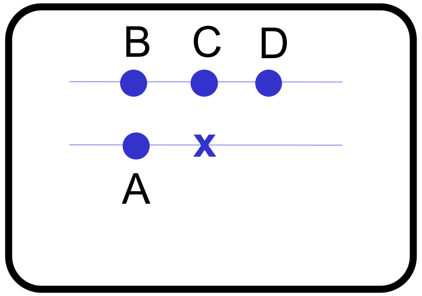
$$PSNR = 10 \log \frac{\left(2^n - 1\right)^2}{MSE}$$

N – number of pixels in image
x_i – pixel value in original image
x_i' – pixel value in reconstructed image
n – number of bits per sample

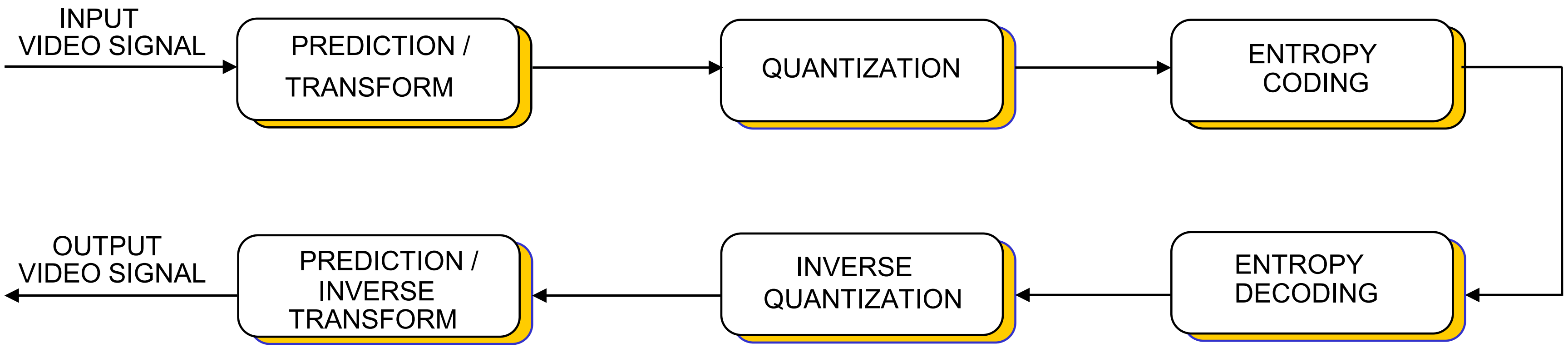
    - subjective quality
      - Mean Opinion Score (MOS) obtained by experiments in which observers evaluate image quality under controlled conditions

Compression methods

- intraframe coding
  - frame is processed independently of the other frames in the video sequence, and the spatial and statistical redundancy is removed
  - transform coding is most commonly used
  - it is possible to apply predictive coding in which the value of each pixel to be coded (X) is predicted from the previously coded pixel (A) in the same line or from the nearest pixels (B, C, D) of the previous line



Spatial correlation

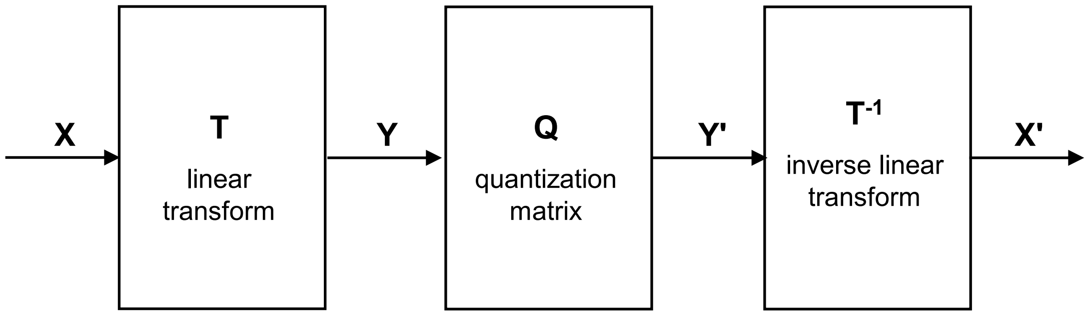


Transform coding

- $\mathbf{X} = \{x_1, x_2, \dots, x_N\}^T$ is the vector of the input image samples
  - there is a high degree of correlation between adjacent samples in natural images
  - most of the power of natural images is contained in the low frequency range
- by applying the linear transformation **T** to the the vector of the samples in the input image **X**, a vector of transformed samples $\mathbf{Y} = \mathbf{TX}$ is created
  - components of the vector **Y** are called transform coefficients
- if the linear transform **T** results in components of the vector **Y** that are less correlated than the components of the vector **X**, the vector **Y** may be more efficiently coded than **X**
- invere transform $\mathbf{T}^{-1}$ restores the input data sequence $\mathbf{X}=\mathbf{T}^{-1}\mathbf{Y}$
  - If the matrix **T** is ortogonal then inverse tranform can be obtained by its transposition: $\mathbf{T}^{-1}=\mathbf{T}^{T}$

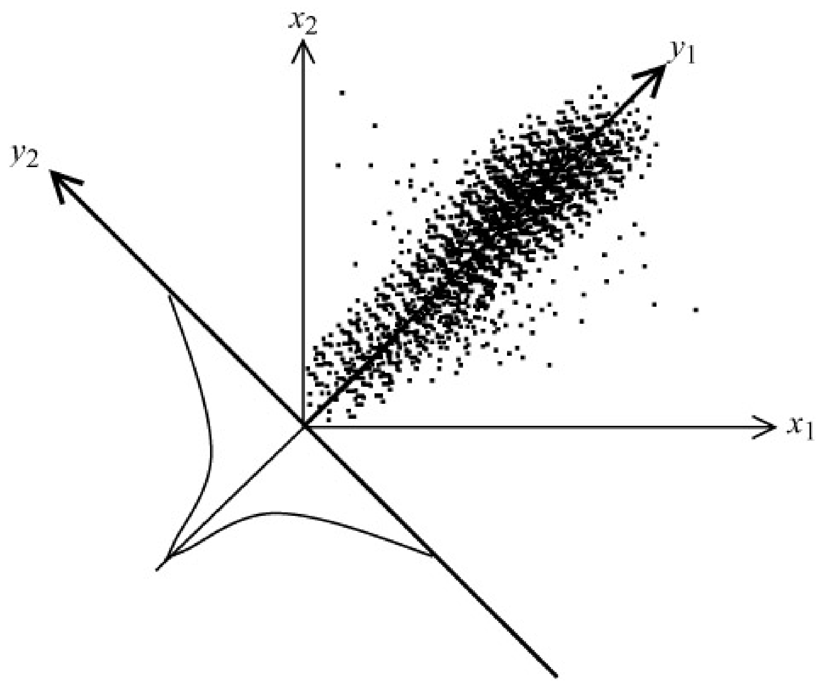
Transform coding

- desirable properties of the transformation
  - the components of the vector **Y** are statistically independent
  - after transformation most of the power is compressed into a small number of components
  - the components of the **Y** vector with the largest values are located in the low frequency range
- if **T** meets the above properties, the signal information is contained in the first few components of the vector **Y**
  - less significant components can be roughly quantized or equaled to 0 without noticeable effect on signal quality
  - **Y’** is the vector of quantized transform coefficients, and **X'** is the vector of reconstructed image samples
  - quantization is the step by which compression is achieved and due to which **X≠X'**



Transform coding

- example of transfrom coding
  - x_1 i x_2 are adjacent pixels that can occupy any value in between 0 i 255
  - due to the high degree of correlation, the points x_1 i x_2 are grouped around the line that closes an angle of 45^o with x_1 axis
  - if we rotate the coordinates x_1x_2 by 45^o, a new coordinate system y_1y_2 will be obtained
  - along the y_1 axis the points denoting the appearance of pairs of values y_1 and y_2 have a uniform distribution
  - along the y_2 axis the points denoting the appearance of pairs of values y_1 and y_2 are condensed around the values y_2=0
  - the number of bits required to code y_1 is similar to the number of bits required to code x_1
  - y_2 requires a smaller number of bits
  - in general, the samples shown in the y_1y_2 system can be coded with fewer bits than the samples in the x_1x_2 system



Transform coding

- rotation of x_1x_2 by 45° represents transformation of vector **X** = $\{x_1, x_2\}^T$ by transform matrix **T**

$$T = \begin{bmatrix} \cos 45 & \sin 45 \\ \sin 45 & -\cos 45 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

- elements of transform matrix are basis vectors
- transform coefficients y_1 and y_2 are

$$[y_1, y_2] = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

$$y_1 = \frac{1}{\sqrt{2}}(x_1 + x_2) \qquad y_2 = \frac{1}{\sqrt{2}}(x_1 - x_2)$$

- y_1 is Direct Current (DC) coefficient proportional to the mean of x_1 and x_2, and y_2 represents the remaining difference
- the normalization factor $\frac{1}{\sqrt{2}}$ ensures that the signal strength will not be changed
  - the power of the components in the pixel domain ($x_1^2+x_2^2$) is equal to the power of the components in the transform domain ($y_1^2+y_2^2$)

Discrete cosine transform

- Discrete Cosine Transform (DCT)
  - its properties approach the optimal transform (optimal transform completely decorrelates the transform coefficients)
  - there are fast algorithms for its calculation
  - it is used in almost all standards for image and video compression

- 1D DCT

$$F(u) = \sqrt{\frac{2}{M}} \cdot C(u) \cdot \sum_{i=0}^{M-1} f(i) \cdot \cos\frac{(2 \cdot i + 1) \cdot u \cdot \pi}{2 \cdot M}$$

$$C(u) = \begin{cases} \frac{1}{\sqrt{2}} & for\ u = 0 \\ 1 & for\ \ u = 1,2,\ldots,M-1 \end{cases}$$

F(u) - transform coefficient

f(i) - pixel value

u - coordinate in the transform domain

i - coordinate in the pixel domain

- DCT can be understood as the expansion of the function *f*(*i*) by orthogonal functions

$$W(i,u) = \sqrt{\frac{2}{M}} \cdot C(u) \cdot \cos\frac{(2 \cdot i + 1) \cdot u \cdot \pi}{2 \cdot M}, \quad i,u = 0,1,\ldots,M-1$$

- orthogonality condition

$$\sum_{i=0}^{M-1} W(i,p) \cdot W(i,q) = \begin{cases} = 0 & za\ p \neq q \\ \neq 0 & za\ p = q \end{cases}$$

Discrete cosine transform

- 2D DCT

$$F(u,v) = \frac{2}{\sqrt{M \cdot N}} \cdot C(u) \cdot C(v) \cdot \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} f(i,j) \cdot \cos\left[\frac{(2 \cdot i + 1) \cdot u \cdot \pi}{2 \cdot M}\right] \cdot \cos\left[\frac{(2 \cdot j + 1) \cdot v \cdot \pi}{2 \cdot N}\right]$$

$F(u,v)$ - transform coefficient

$f(i,j)$ - pixel value

u, v - coordinates in the transform domain (spatial frequencies)

i , j - coordinates in the pixel domain

$C(u)=C(v) = (1/2)^{1/2}$, for $u,v = 0$

$C(u)=C(v) = 1$, for $u = 1,2,...M-1$, $v = 1,2,...N-1$

$u=v=0$ - $F(0,0)$- DC $u,v\neq 0$ - $F(u,v)$ - Alternating Current (AC) coefficients

- 2D inverse DCT (2D IDCT)

$$f(i,j) = \frac{2}{\sqrt{M \cdot N}} \cdot \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} C(u) \cdot C(v) \cdot F(u,v) \cdot \cos\left[\frac{(2 \cdot i + 1) \cdot u \cdot \pi}{2 \cdot M}\right] \cdot \cos\left[\frac{(2 \cdot j + 1) \cdot v \cdot \pi}{2 \cdot N}\right]$$

Discrete cosine transform

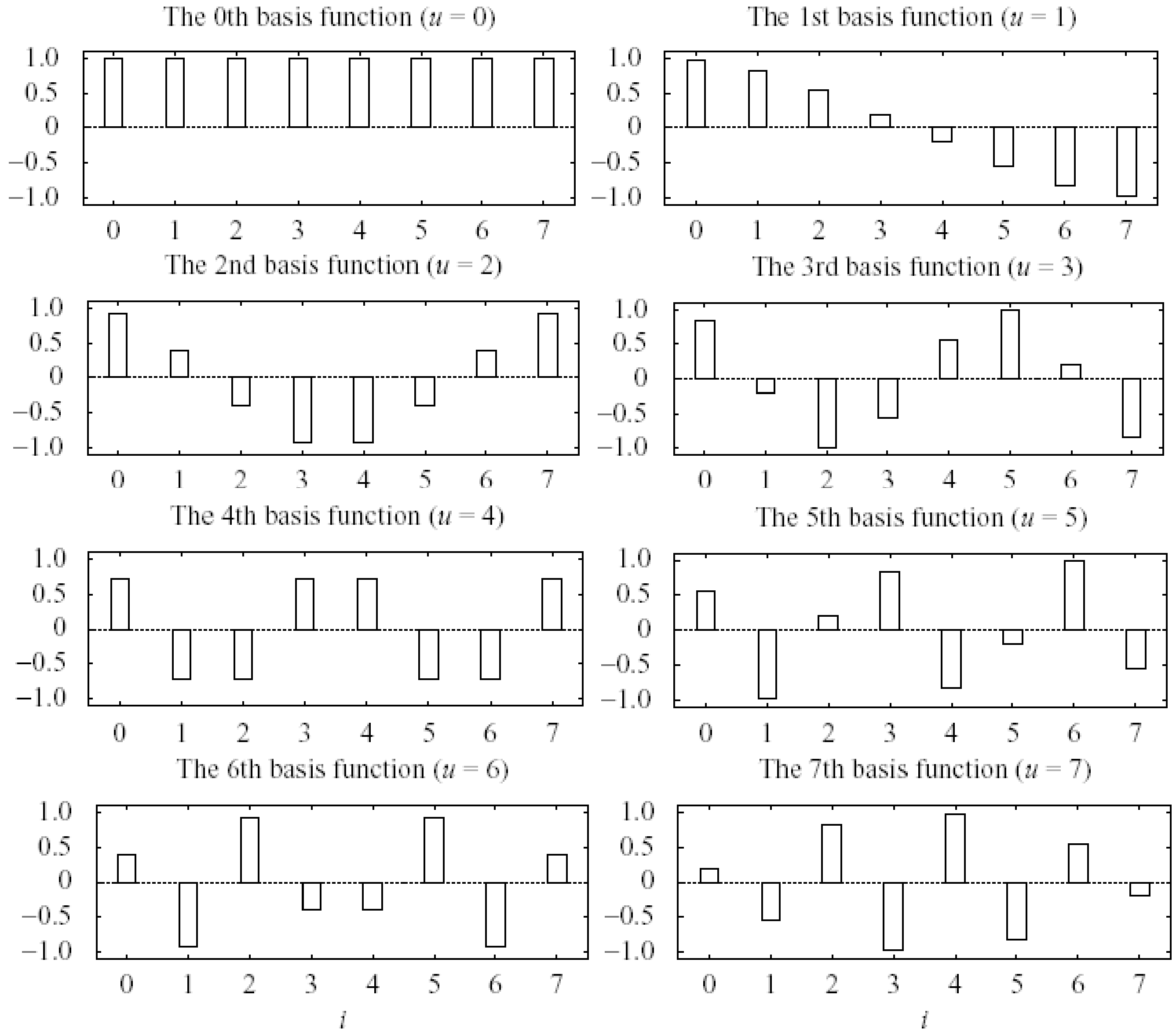
- DCT separates the input signal into a DC component and AC components
  - DCT transforms input signal samples into a frequency domain where compression implementation is simpler and more efficient
  - DCT coefficients have a larger dynamic range than the dynamic range of the input signal (DCT does not perform image compression)
- IDCT reconstructs the input signal
- in the transform process and input signal reconstruction DCT and IDCT use the same set of cosine functions called basis functions
- the basis functions are orthogonal and orthonormal
- basis functions are orthonormal if they are orthogonal and if they meet the following condition

$$\sum_{i=0}^{M-1} W(i,p) \cdot W(i,q) = 1 \quad za\, p = q$$

  - orthonormality is a desirable condition because it ensures that the signal will not be amplified during transformation
  - orthonormality is achieved by selecting *C(u)* and *C(v)*

Discrete cosine transform

- 1D DCT basis functions (i, u =0,1,..., 7)



Discrete cosine transform

- Example: The input signal is a DC signal with amplitude 100, $f(i)$=100, i=0, 1, ...,7. Determine $F(u)$.

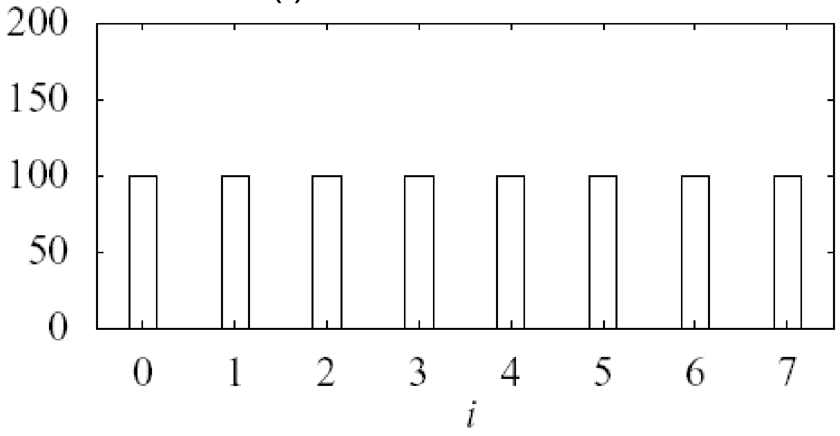
$$F(0)=\sqrt{\frac{2}{8}}\cdot\frac{1}{\sqrt{2}}\sum_{i=0}^{7}100\cdot\cos\frac{(2\cdot i+1)\cdot 0\cdot\pi}{2\cdot 8}=\frac{1}{2\sqrt{2}}800=282{,}84$$

$$F(1)=\sqrt{\frac{2}{8}}\cdot 1\cdot\sum_{i=0}^{7}100\cdot\cos\frac{(2\cdot i+1)\cdot 1\cdot\pi}{2\cdot 8}=\frac{1}{2}(100\cos\frac{\pi}{16}+100\cos\frac{3\pi}{16}+100\cos\frac{5\pi}{16}$$

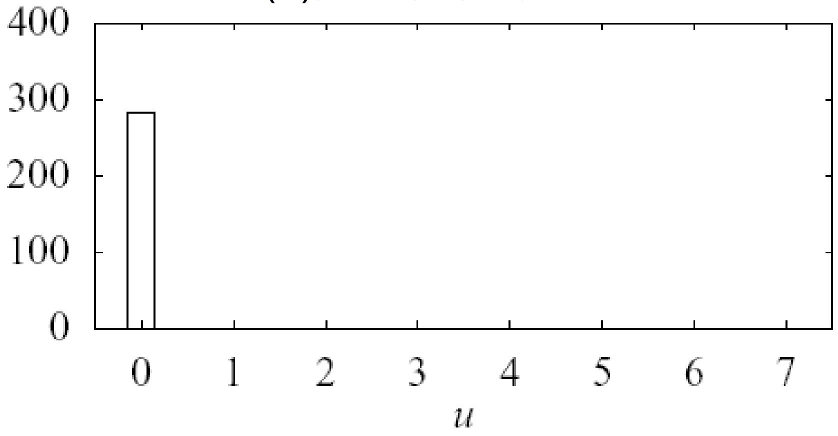
$$+100\cos\frac{7\pi}{16}+100\cos\frac{9\pi}{16}+100\cos\frac{11\pi}{16}+100\cos\frac{13\pi}{16}+100\cos\frac{15\pi}{16})=0$$

$$F(2)=F(3)=F(4)=F(5)=F(6)=F(7)=0$$

$f(i)$=100, i=0, 1, ...,7



$F(u)$, u=0, 1, ...,7



Discrete cosine transform

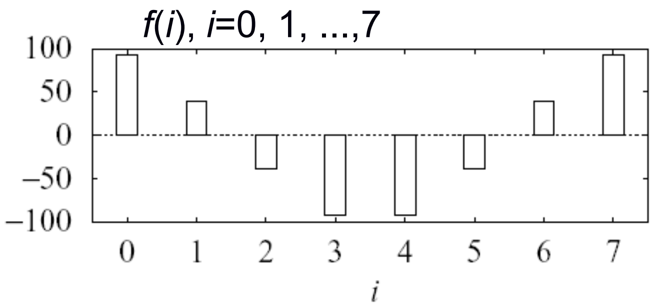
- Example: The input signal corresponds to the second cosine basis function with an amplitude of 100. Determine $F(u)$.

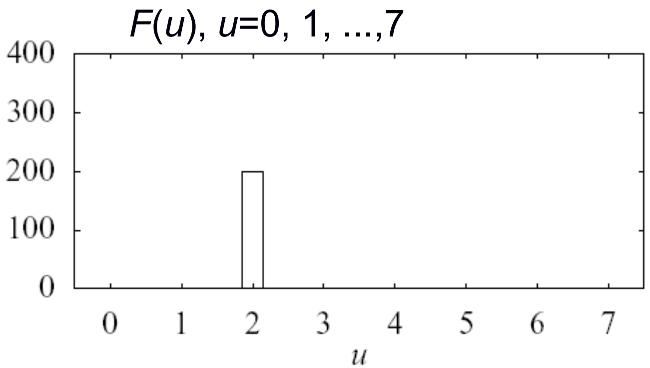
$$f(i)=\sum_{i=0}^{7}100\cdot\cos\frac{(2\cdot i+1)\cdot 2\cdot\pi}{16}$$

$$F(0)=\sqrt{\frac{2}{8}}\cdot\frac{1}{\sqrt{2}}\sum_{i=0}^{7}100\cdot\cos\frac{(2\cdot i+1)\cdot 2\cdot\pi}{16}\cdot\cos\frac{(2\cdot i+1)\cdot 0\cdot\pi}{2\cdot 8}=\frac{1}{2\sqrt{2}}100(\cos\frac{\pi}{8}+\cos\frac{3\pi}{8}+\cos\frac{5\pi}{8}$$
$$+\cos\frac{7\pi}{8}+100\cos\frac{9\pi}{8}+\cos\frac{11\pi}{8}+\cos\frac{13\pi}{8}+\cos\frac{15\pi}{8})=0$$

$$F(2)=\sqrt{\frac{2}{8}}\cdot 1\cdot\sum_{i=0}^{7}100\cdot\cos\frac{(2\cdot i+1)\cdot 2\cdot\pi}{16}\cdot\cos\frac{(2\cdot i+1)\cdot 2\cdot\pi}{16}=\frac{1}{2}100(\cos\frac{\pi}{8}\cos\frac{\pi}{8}+\cos\frac{3\pi}{8}\cos\frac{3\pi}{8}$$
$$+\cos\frac{5\pi}{8}\cos\frac{5\pi}{8}+\cos\frac{7\pi}{8}\cos\frac{7\pi}{8}+\cos\frac{9\pi}{8}\cos\frac{9\pi}{8}+\cos\frac{11\pi}{8}\cos\frac{11\pi}{8}+\cos\frac{13\pi}{8}\cos\frac{13\pi}{8}$$
$$+\cos\frac{15\pi}{8}\cos\frac{15\pi}{8})=\frac{1}{2}(1+1+1+1)=200$$

$$F(1)=F(3)=F(4)=F(5)=F(6)=F(7)=0$$



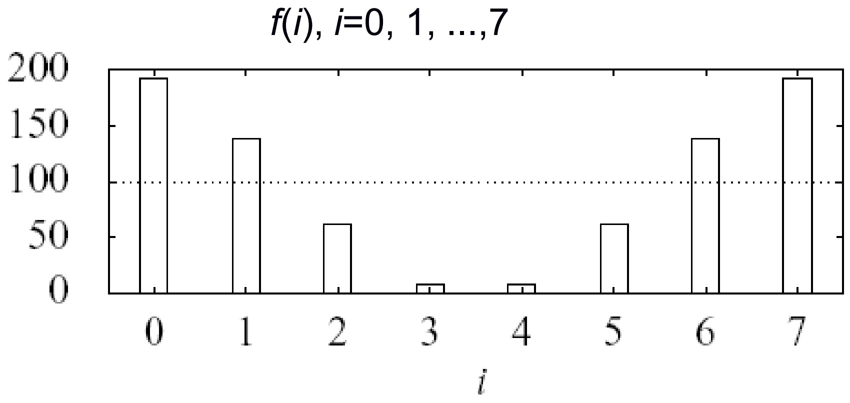


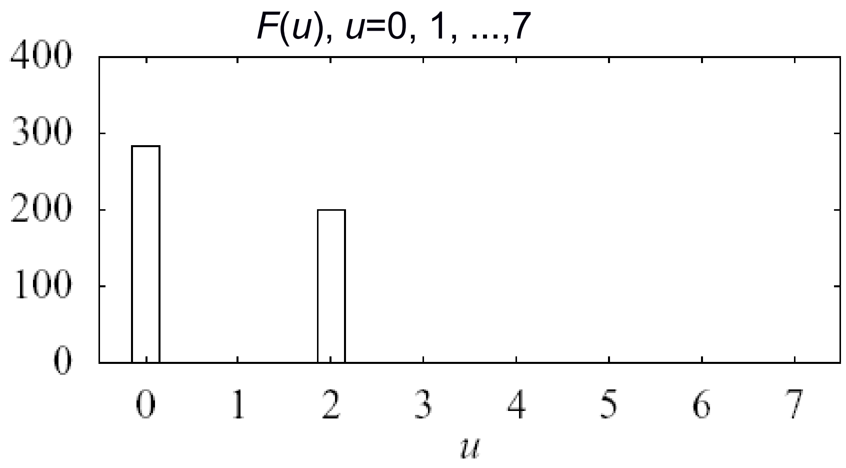
Discrete cosine transform

- Example: The input signal is the sum of the signals from the previous two examples. Determine *F*(*u*).

$$f(i)=\sum_{i=0}^{7}100+100\cdot\cos\frac{(2\cdot i+1)\cdot 2\cdot\pi}{16}$$

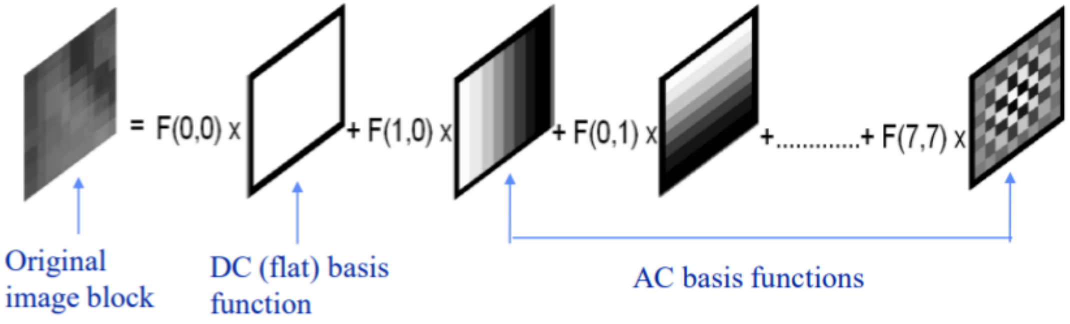
$$F(0)=282{,}84$$
$$F(2)=200$$
$$F(1)=F(3)=F(4)=F(5)=F(6)=F(7)=0$$

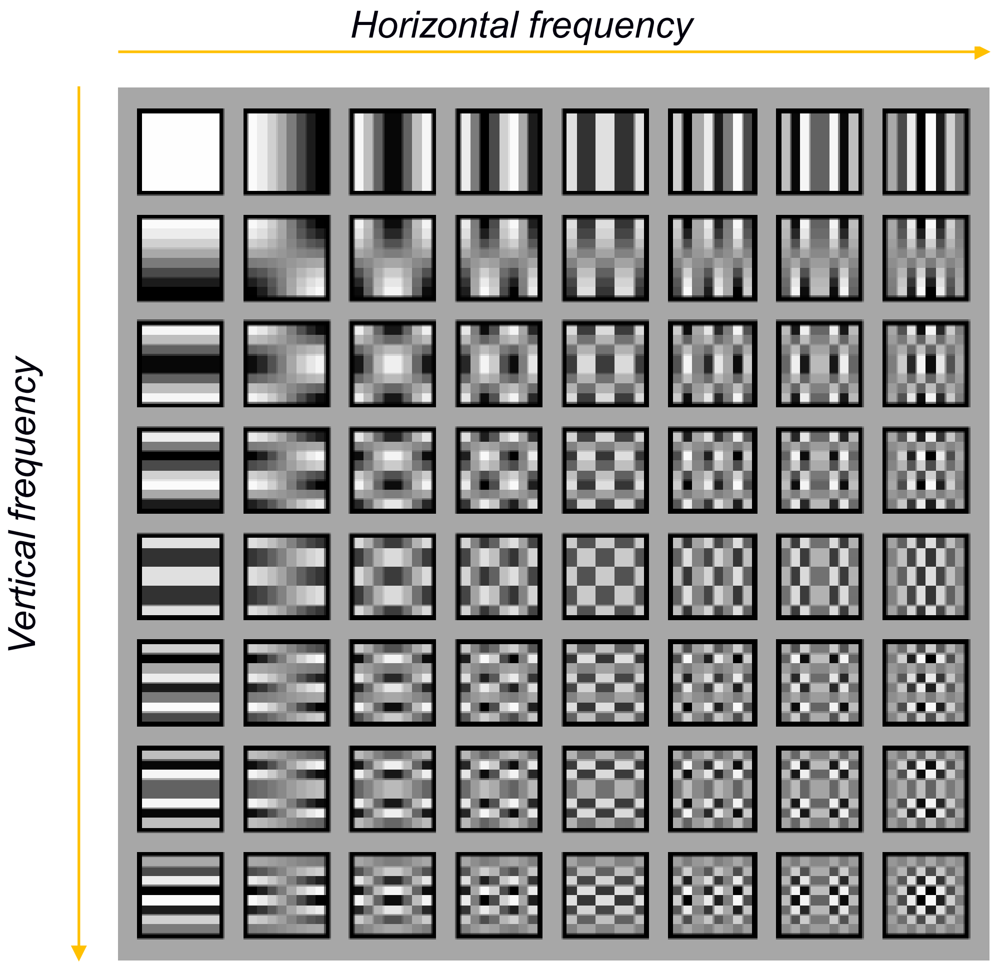




Discrete cosine transform

- image and video compression standards use 2D DCT with a block size of 8x8 (M = N = 8) pixels or 4x4 pixels
- 2D DCT basis functions for M=N=8
  - white represents positive values and black negative values of the basis functions
  - each DCT coefficient is associated with one basis function
  - basis functions determine the meaning of individual DCT coefficients $F(u,v)$ in relation to the horizontal (u) and vertical (v) frequency





Discrete cosine transform

- 8x8 2D DCT

$$F(u,v)=\frac{1}{4}\cdot C(u)C(v)\cdot\sum_{i=0}^{7}\sum_{j=0}^{7}f(i,j)\cdot\cos\left[\frac{(2\cdot i+1)\cdot u\cdot\pi}{16}\right]\cdot\cos\left[\frac{(2\cdot j+1)\cdot v\cdot\pi}{16}\right]$$

$$F(0,0)=\frac{1}{8}\cdot\sum_{i=0}^{7}\sum_{j=0}^{7}f(i,j)$$

eight times the mean of the pixels in the block

- to calculate each *F(u,v)* it is necessary to go through all the values of the pixels *f(i,j)*
- for example, to calculate the third coefficient in the second row (u=2, v=1, $C(u)=C(v)$=1) 64 additions are required

$$F(2,1)=\frac{1}{4}\sum_{i=0}^{7}\sum_{j=0}^{7}f(i,j)\cdot\cos\left[\frac{(2\cdot i+1)\cdot 2\cdot\pi}{16}\right]\cdot\cos\left[\frac{(2\cdot j+1)\cdot\pi}{16}\right]$$

Discrete cosine transform

- 8x8 2D DCT can be considered as product of two 1D DCTs which reduces the number of operations required to calculate DCT coefficients

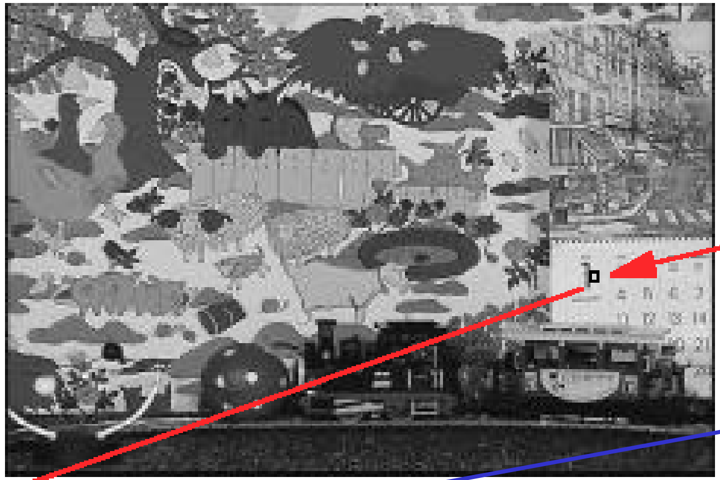
$$G(i,v) = \frac{1}{2} \cdot C(v) \cdot \sum_{j=0}^{7} f(i,j) \cdot \cos\left[\frac{(2 \cdot j + 1) \cdot v \cdot \pi}{16}\right]$$

$$F(u,v) = \frac{1}{2} \cdot C(u) \cdot \sum_{i=0}^{7} G(i,v) \cdot \cos\left[\frac{(2 \cdot i + 1) \cdot u \cdot \pi}{16}\right]$$

  - in that case to calculate each *F(u,v)* 16 additions are required

Discrete cosine transform

- 8x8 2D DCT



i →, j ↓

181	207	230	154	194	230	200	200
164	181	194	194	181	181	177	230
194	181	181	194	194	188	194	217
194	171	194	194	194	217	211	194
194	181	181	164	194	217	230	200
194	171	181	194	177	217	194	230
171	217	194	211	200	194	164	154
181	181	194	181	154	160	154	154

Pixel values: $f(i,j)$

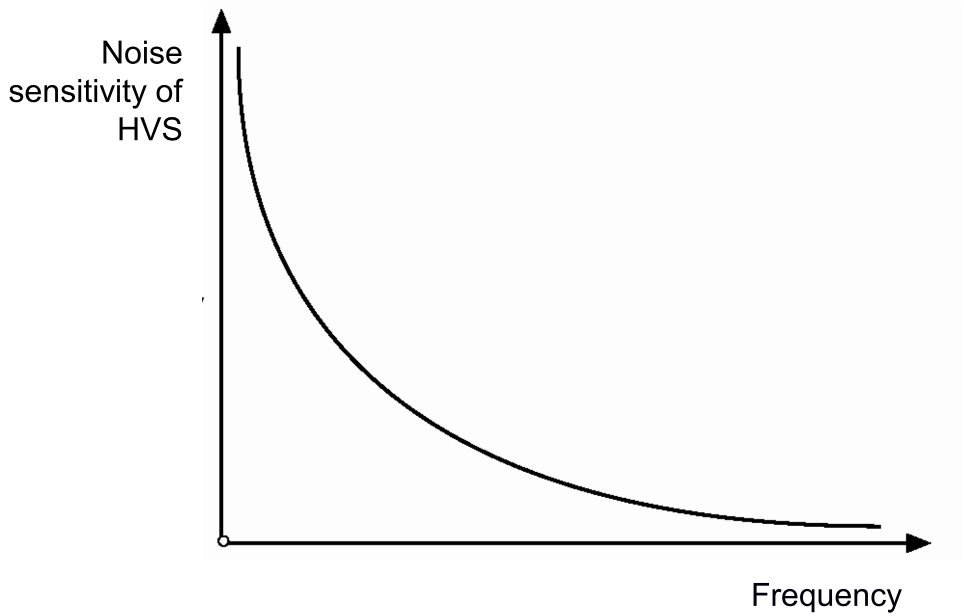
u →, v ↓

1524.1	-28.8	3.3	-4.7	-17.6	-6.7	28.0	-8.1
39.5	-40.8	19.5	-9.0	-6.9	-15.9	15.7	18.0
-34.9	50.9	-18.1	-32.7	-19.2	-25.2	6.2	-2.1
36.8	-25.9	9.1	17.2	-27.8	-14.8	14.9	1.1
-2.9	5.5	7.3	26.2	-40.1	11.8	10.1	26.6
24.5	22.2	-31.3	17.1	-17.0	-7.0	-3.1	-2.2
3.5	-11.3	27.9	5.0	8.1	8.2	18.1	1.8
4.9	11.9	-2.7	16.3	-14.4	-11.9	-20.7	12.5

Transform coefficients: $F(u,v)$

Quantization of DCT coefficients

- the goal of quantization is to achieve compression
- quantization discards coefficients that are not important to HVS
- HVS is more sensitive to noise in the low frequency area than in the high frequency area
    - HVS is very sensitive to errors of DC coefficient and low frequency (LF) AC coefficients
    - HVS is not sensitive to errors of high frequency (HF) AC coefficients



Quantization of DCT coefficients

- quantization is performed using quantization table with 64 values of $q(u,v)$
- DCT coefficients $F(u,v)$ are divided by $q(u,v)$ and the result is converted to integer
  - the values in the quantization table $q(u,v)$ determine the quantization step for the corresponding DCT coefficient $F(u,v)$
    - the values of $q(u,v)$ are determined in accordance with the characteristics of HVS (they are smaller for the DC coefficient and LF AC coefficients and increase by increasing spatial frequency)
  - $S(u,v)$ – quantized coefficients

$$S(u,v) = round\left(\frac{F(u,v)}{Q(u,v)}\right) = round\left(\frac{F(u,v)}{q(u,v)S}\right)$$

$S(u,v)$ - quantized value of DCT coefficient

$Q(u,v)$ - quantization coefficient (product of values from quantization table $q(u,v)$ and scaling factor S)

  - scaling factor S changes the compression ratio and image quality
- in decoder quantized value of DCT coefficient is multiplied by the corresponding quantization coefficient

$$F'(u,v) = S(u,v)Q(u,v)$$

Quantization of DCT coefficients

- examples of quantization tables *q(u,v)* for luminance and chrominance blocks (JPEG)
  - the quantization table for the chrominance components has larger values in general implying that the quantization of the chrominance planes is coarser when compared with the luminance plane
  - this is done to exploit the human visual system's (HVS) relative insensitivity to chrominance components as compared with luminance components

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	120	101
72	92	95	98	112	100	103	99

Quantization table for luminance component

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

Quantization table for chrominance components

Quantization of DCT coefficients

7842	199	448	362	342	112	31	22
198	151	181	264	59	37	14	3
142	291	218	87	27	88	27	12
111	133	159	119	58	65	36	2
49	85	217	50	8	3	14	12
58	120	60	40	41	11	2	1
30	121	61	22	30	1	0	1
22	28	2	33	24	51	44	81

$F(u,v)$

Division by $q(u,v)$

Division by S

980	12	23	16	13	4	1	0
12	9	8	11	2	1	0	0
7	13	8	3	0	2	0	1
5	6	6	4	2	1	0	0
2	3	8	1	0	0	0	0
2	4	2	1	1	0	0	0
1	4	2	1	0	0	0	0
0	0	1	0	0	0	0	0

$S(u,v)$

8	16	19	22	26	27	29	34
16	16	22	24	27	29	34	37
19	22	26	27	29	34	34	38
22	22	26	27	29	34	37	40
22	26	27	29	32	35	40	48
26	27	29	32	35	40	48	58
26	27	29	34	38	48	56	69
27	29	35	38	46	56	69	83

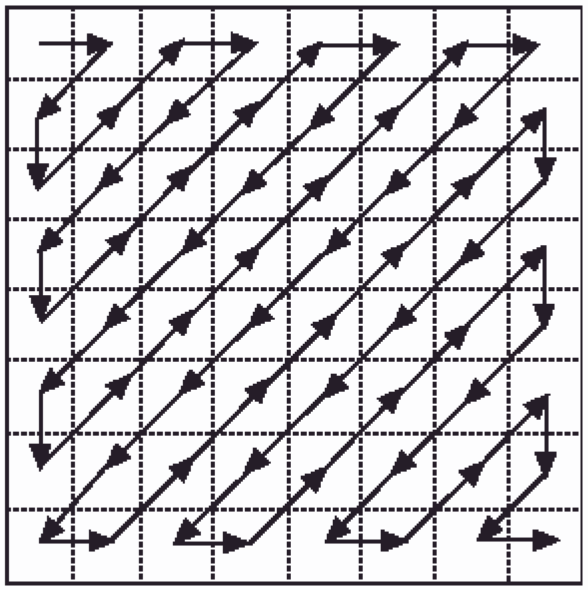
$q(u,v)$

Code	Linear Quant Scale	Non-Linear Quant Scale
1	2	1
8	16	8
16	32	24
20	40	40
24	48	56
28	56	88
31	62	112

S

Zig-zag scanning

- for natural images after quantization many high frequency DCT coefficients will be equal to 0
- by zig-zag scanning, the two-dimensional block of DCT coefficients is transformed into a one-dimensional data set in such a way that the low-frequency coefficients are read firstly and after that the high-frequency coefficients
- after zig-zag scanning, long series of coefficients equal to 0 are created, which can be efficiently entropy coded



Non-Zero values are at the top left corner of the block

−13	−3	2	0	0	0	1	0
6	0	0	0	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

Zig-Zag scan concentrates the non-zero coefficients at the start of the stream

-13, -3, 6, 0, 0, 2, 0, 0, 0, -1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 36 more zeros, the end

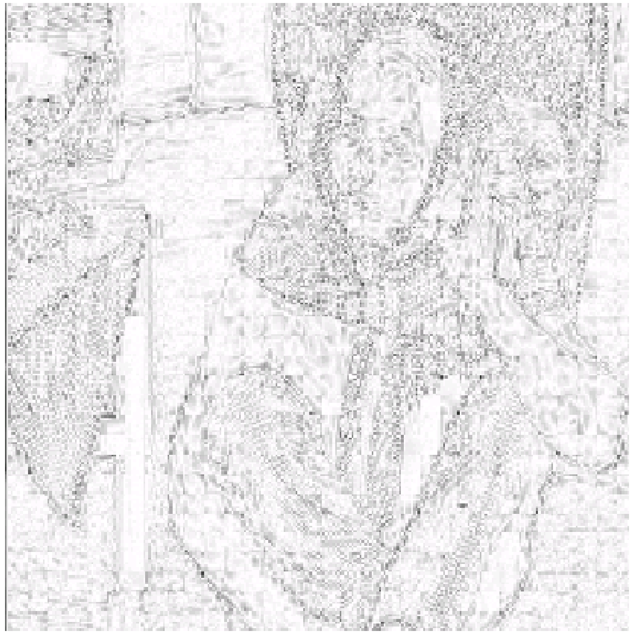
Picture quality

- high compression in systems with DCT leads to distortion which can be seen as the visibility of the block edges in the image (blocking artifacts)

Original image Barbara

Decoded image,
compression ratio 20:1

Difference between original
and decoded image



Interframe coding

- interframe coding
  - difference frame is coded and time redundancy is removed
  - frame A and frame C = (A - B) are transmitted to the decoder
  - frame C can be coded more efficiently by applying intraframe coding in comparison to frame B
  - frame B is reconstructed in the decoder as (A + C)
- in the case of a sequence with fast movements, large compression cannot be achieved by coding the difference of consecutive frames
  - the difference between consecutive images is reduced by the motion prediction and motion compensation process

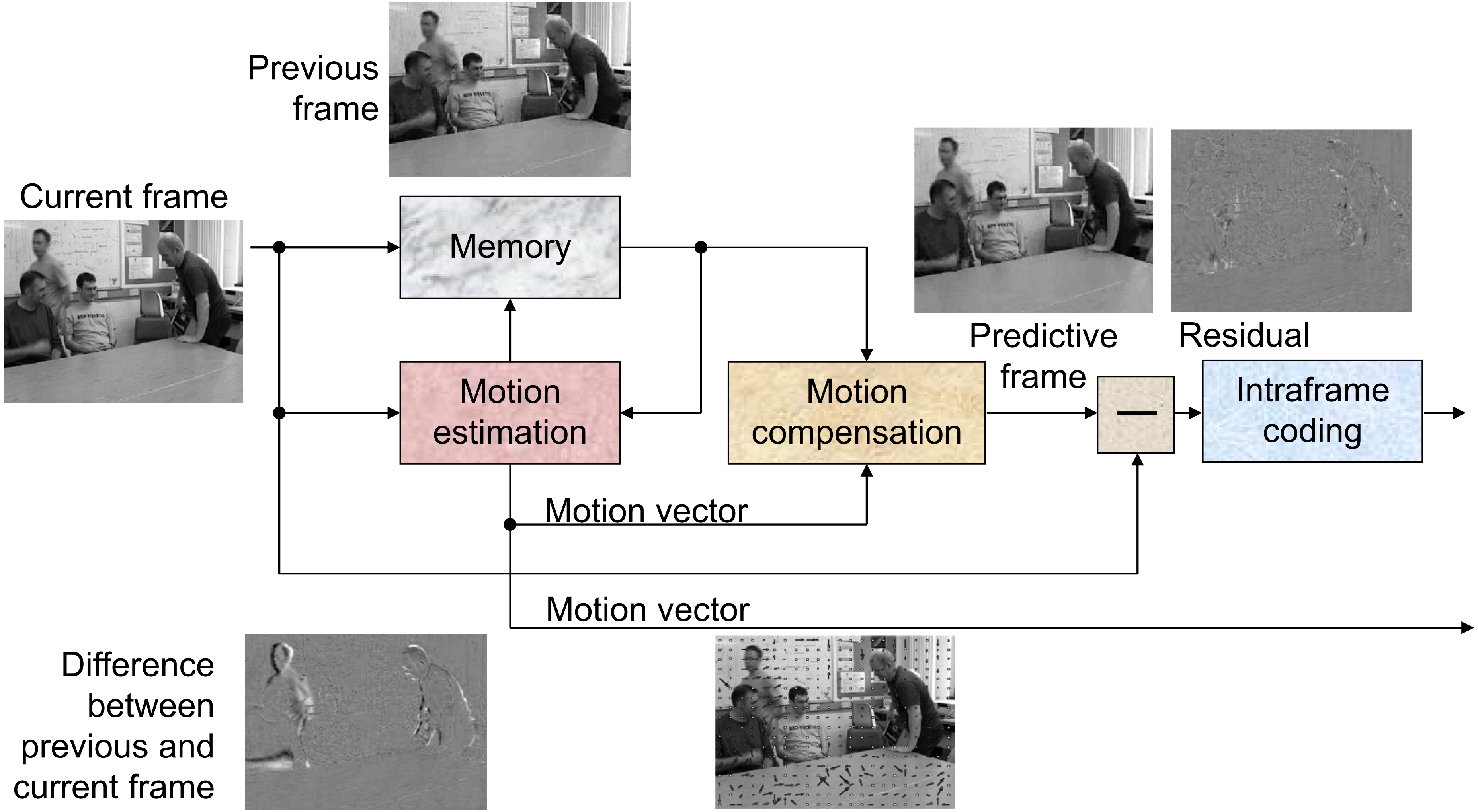
A

B

C

Interframe coding

- motion compensated predictive coding



Interframe coding

- motion compensated predictive coding
  - is used to encode the frame differnce in video signals with a high level of movement that causes a large difference between consecutive frames
  - a comparison of the current and previous image is performed, in order to determine the direction and distance of moving objects between frames
  - the direction and distance of the object's displacement is expressed by a two-dimensional motion vector
  - the coder uses motion vectors to obtain the predictive frame, which is created by moving the picture elements of the previous frame by the motion vectors (thus reducing the difference between the current and previous frame)
  - the predictive frame is subtracted from the current frame
  - the difference between the current and the predictive frame and the motion vectors are transmitted to the decoder

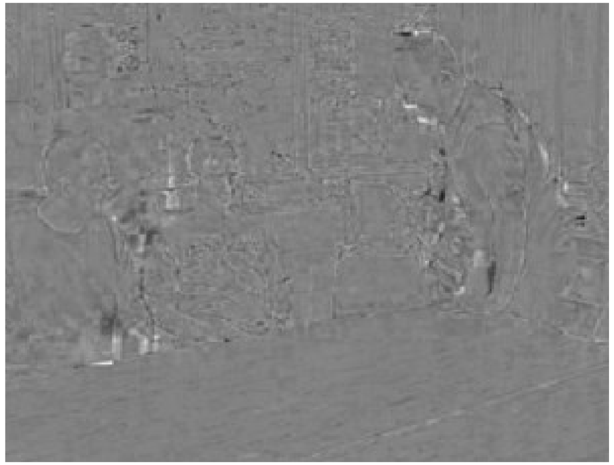
Interframe coding

- motion compensated predictive coding

Previous frame

Current frame

Difference between previous and current frame



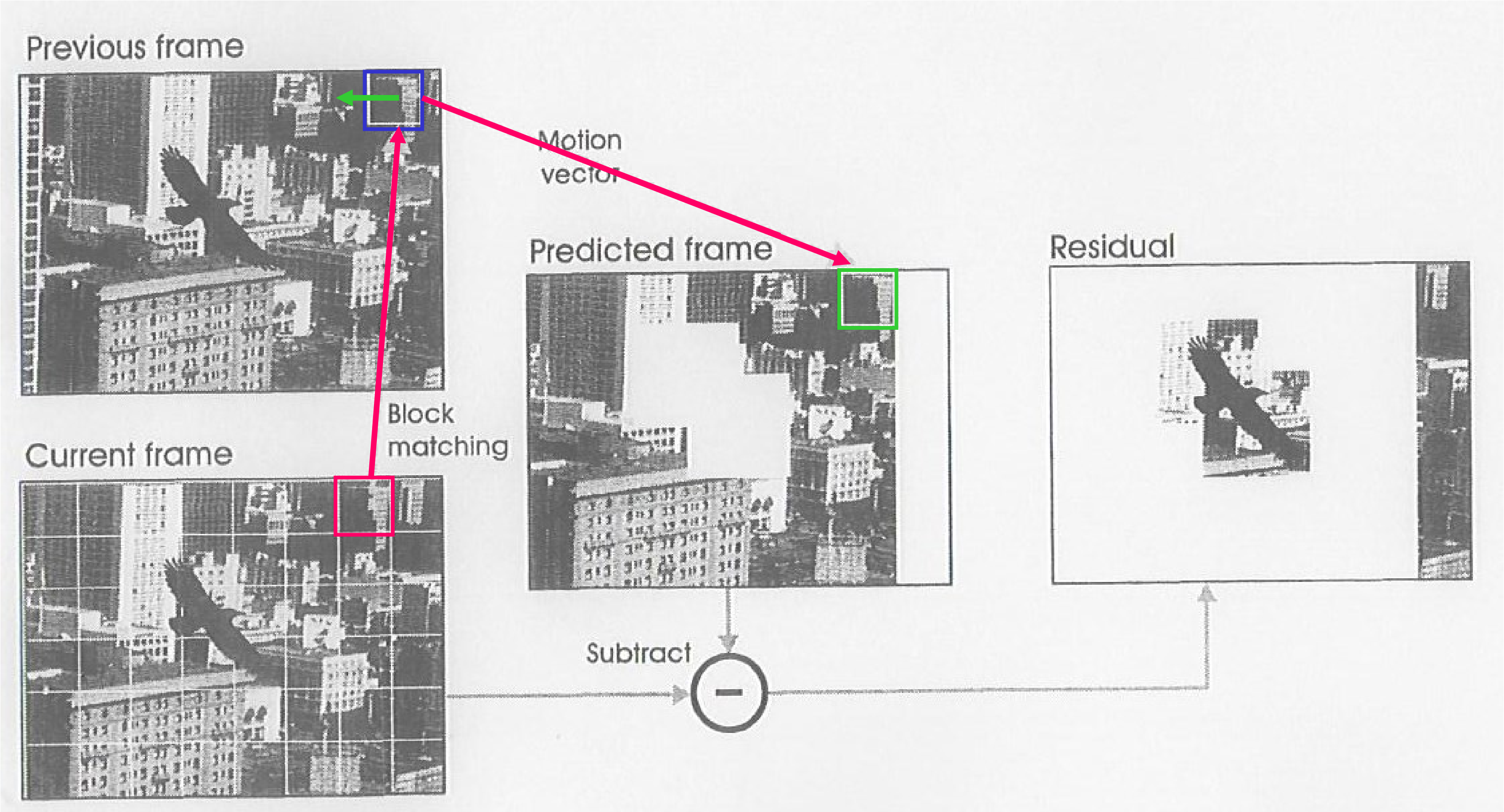
Difference between current and predictive frame (residual)

Interframe coding

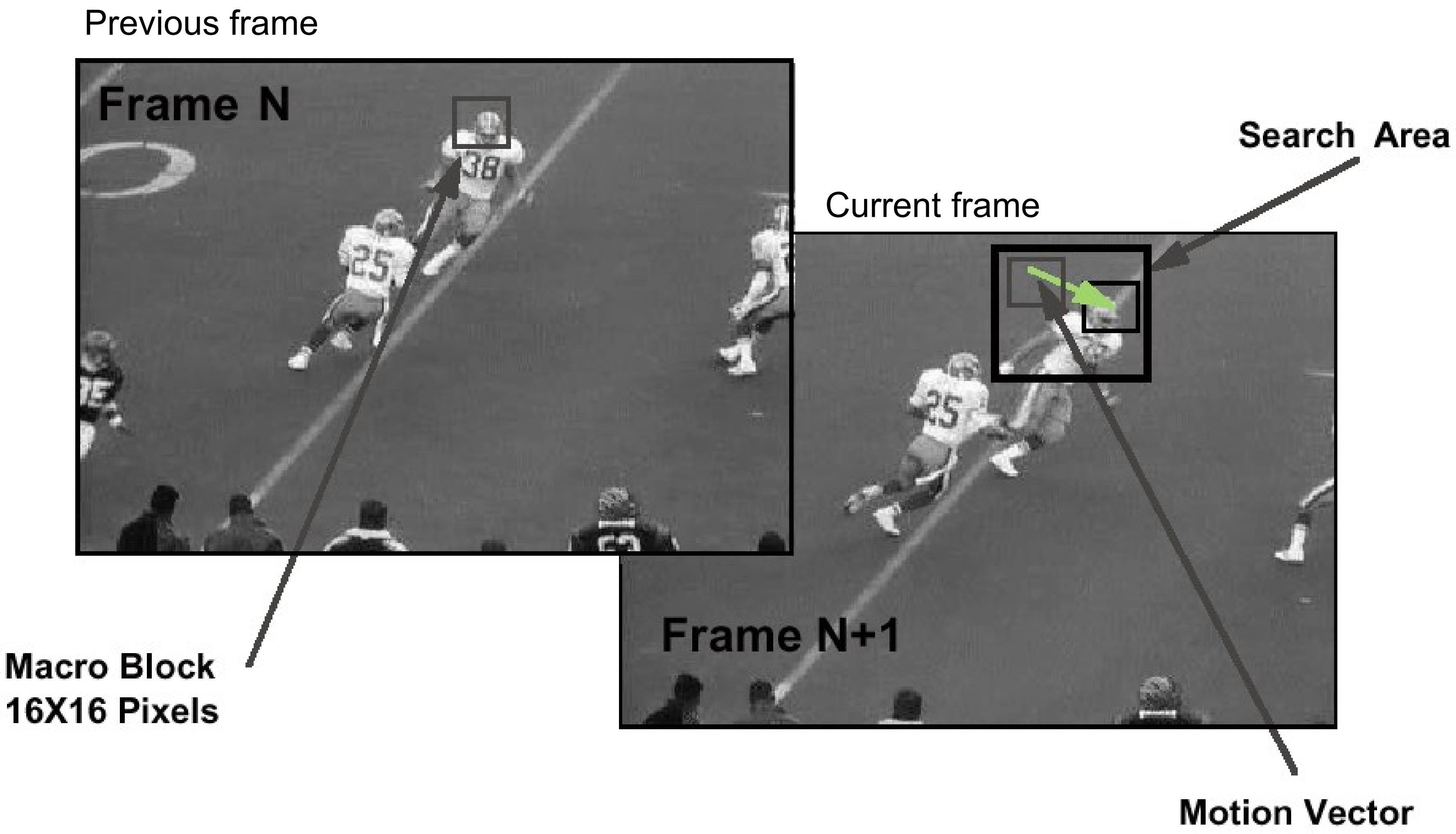
- motion compensation is performed for square or rectangular search areas (macroblock)
  - the macroblock from the current frame is searched in the previous frame
  - the search is not usually performed on the whole image, but on the limited search area (in order to perform the search faster)
  - if a particular macroblock exists in the previous frame, a motion vector is derived that shows the difference in position of the particular macroblock between the previous and current frame
  - if there is no specific macroblock in the previous frame, motion prediction is not performed
- the predictive frame is created by applying a motion vector to the macroblocks in the previous frame
- the predictive frame is subtracted from the current frame and the difference frame (residual) is transmitted to the decoder

Interframe coding

- motion compensated predictive coding

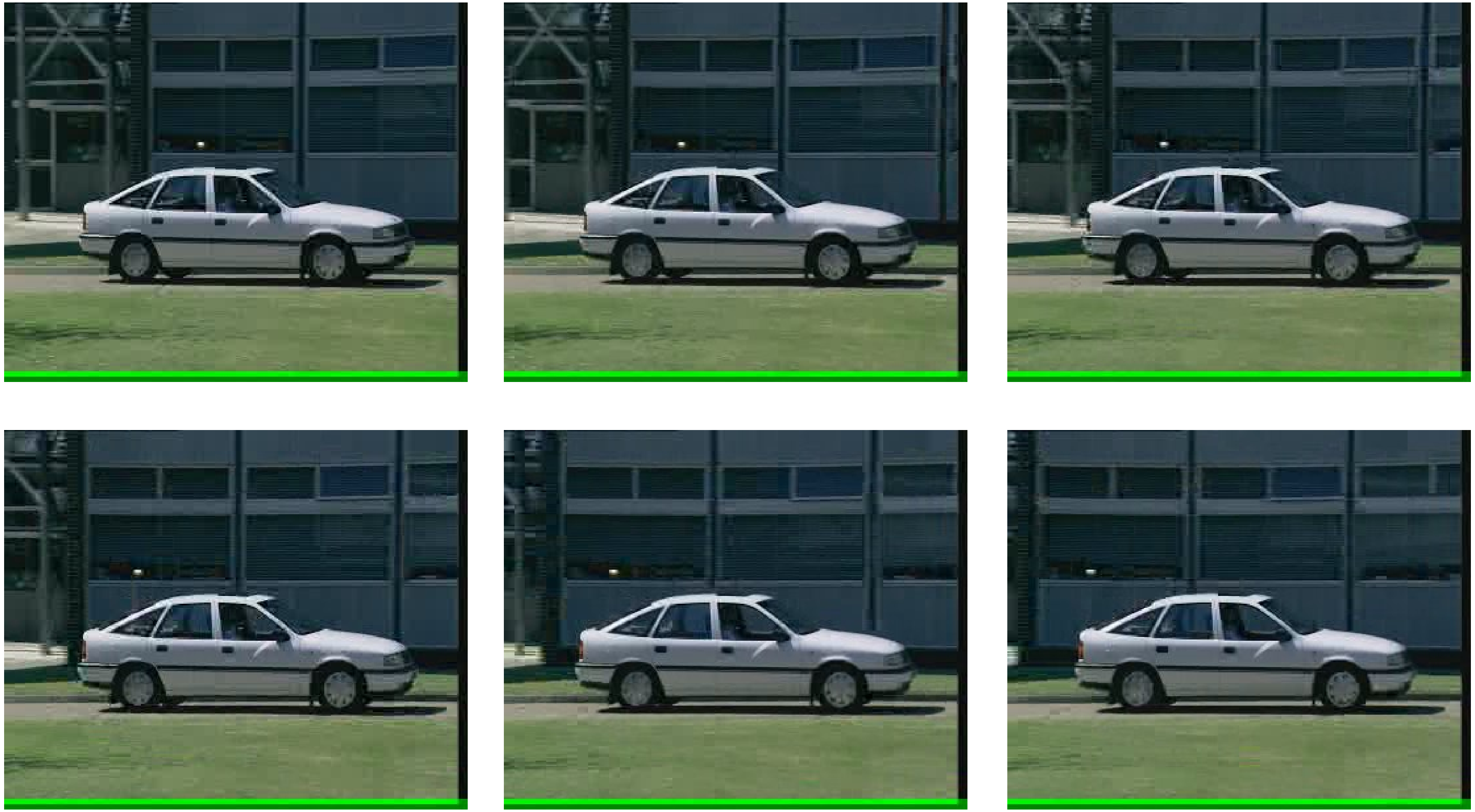


Interframe coding



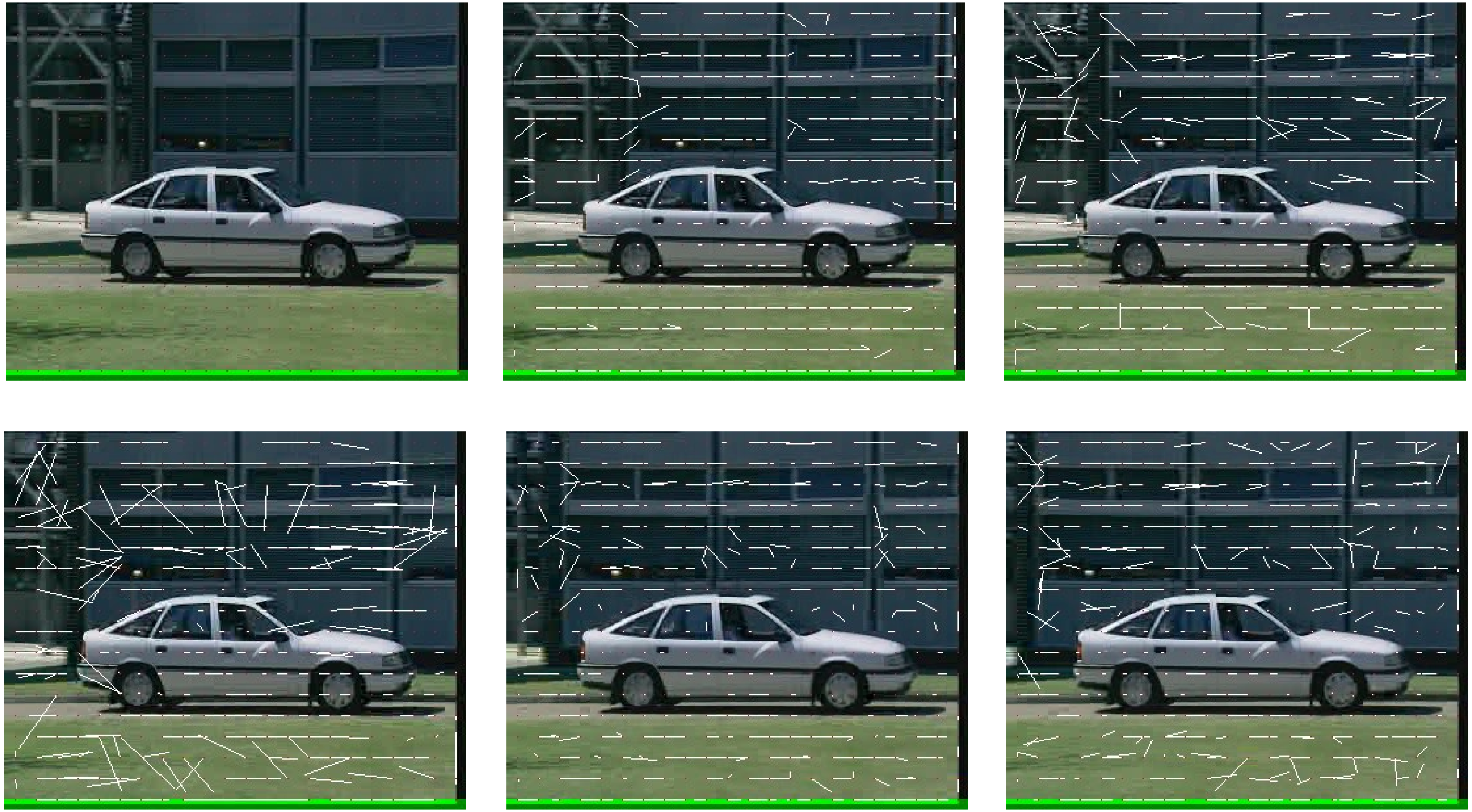
Example of coding results

- video sequence “Vectra”, MPEG-2 coding, 0,768 Mbit/s



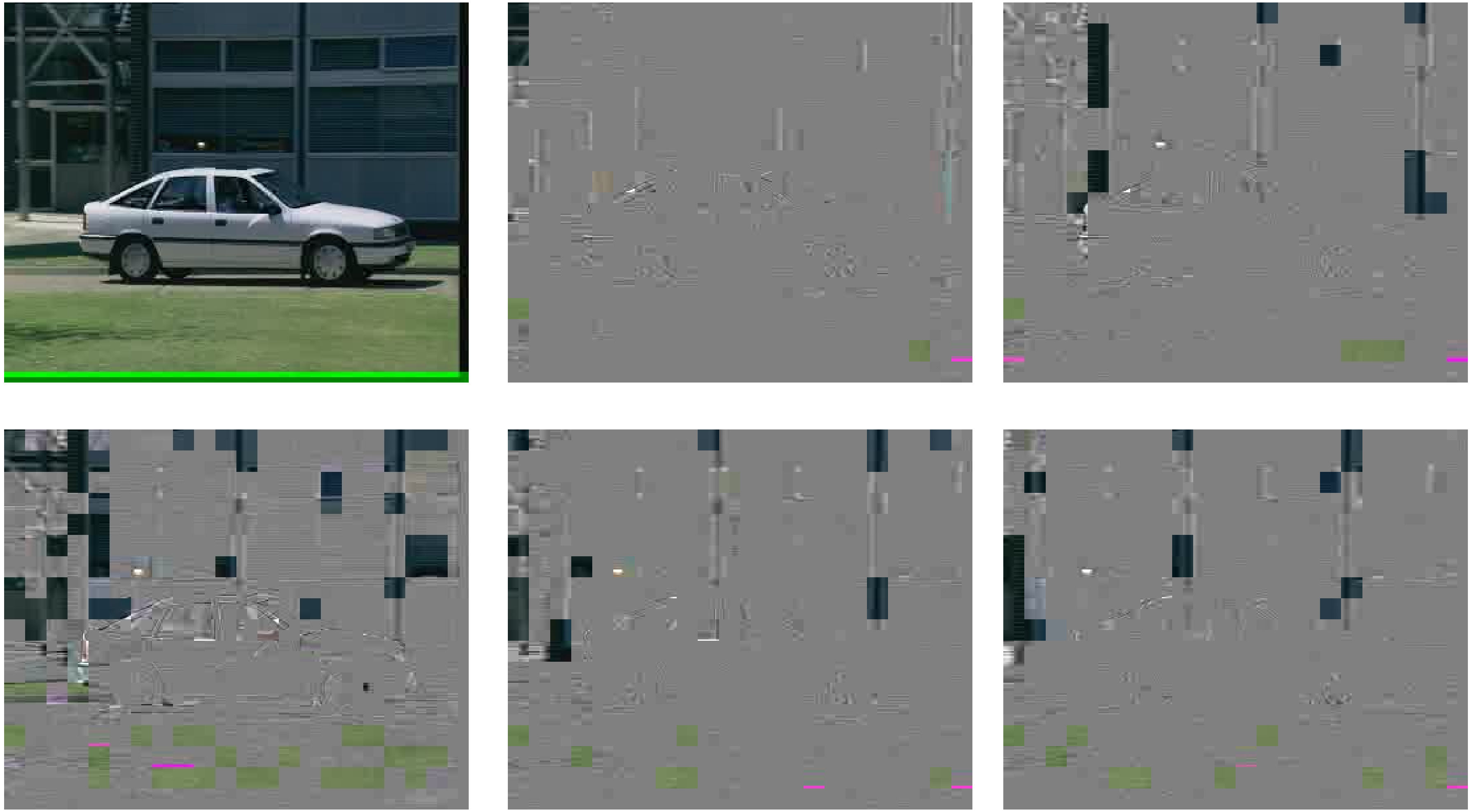
Example of coding results

- motion vectors



Example of coding results

- difference frame



Standards for image and video compression

- standardization bodies

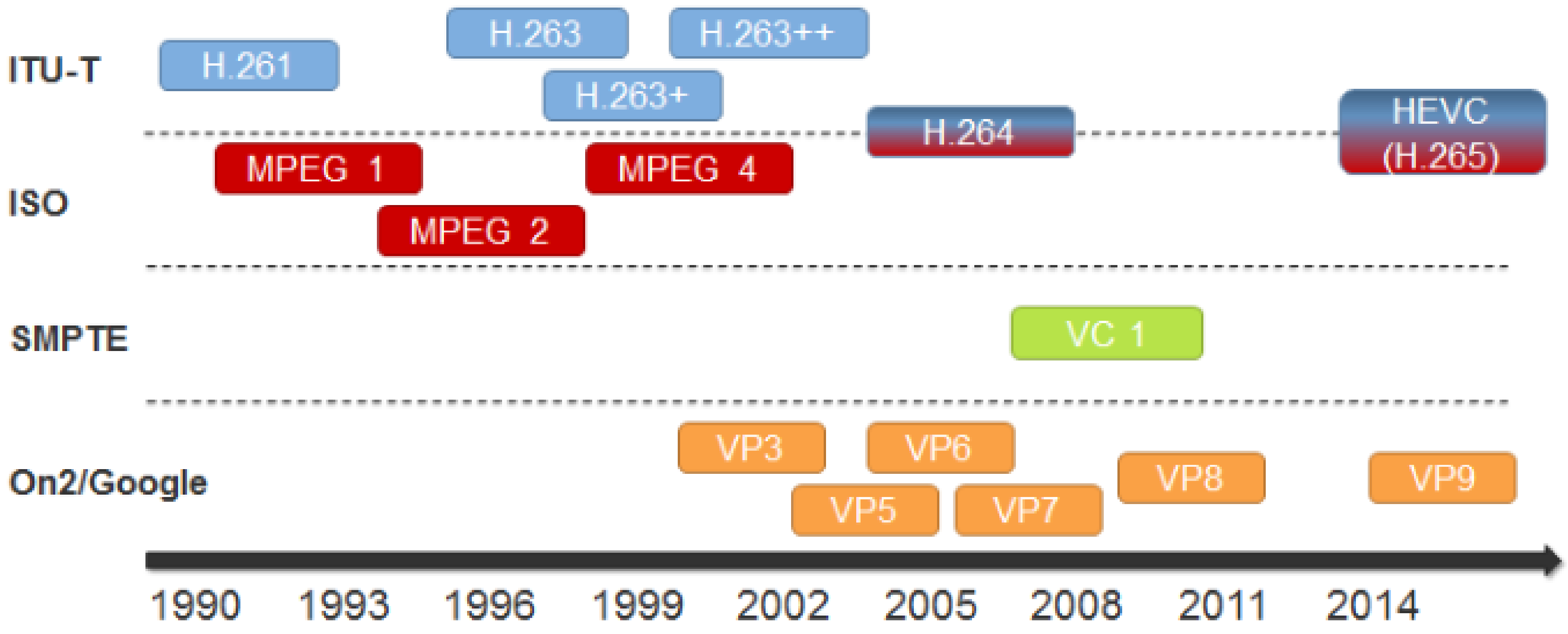
CCITT	International Telegraph and Telephone Consultative Committee
ITU	International Telecommunication Union
JPEG	Joint Photographic Experts Group
MPEG	Moving Picture Experts Group
ISO	International Standards Organization
IEC	International Electrotechnical Commission

Standards for image and video compression

Name	Year	Application
H.261	1990.	Videoconferencing, 64 kbit/s -1,92 Mbit/s
MPEG-1	1991.	CD-ROM, 1,5 Mbit/s
MPEG-2 (H.262)	1994.	Digital televsion, 2-15 Mbit/s
H.263	1995.	Vry low bit-rate transmission (ispod 64 kbit/s)
H.263+(version 2)	1998.	New options added to H.263
MPEG-4 Part 2	1999.	Coding of multimedia objects
MPEG-4 (version 2)	2000.	New options added to MPEG-4 Part 2
H.263++	2000.	New options added to H.263+
MPEG-4 Part 10 (H.264/AVC)	2004.	Different applications, modification of the coding process in relation to MPEG-4 Part 2
MPEG-H Part 2 (H.265/HEVC)	2013.	Increased compression efficiency, image formats up to 8k (8192 × 4320), compression ratio up to 1000:1

Standards for image and video compression

- development of compression standards

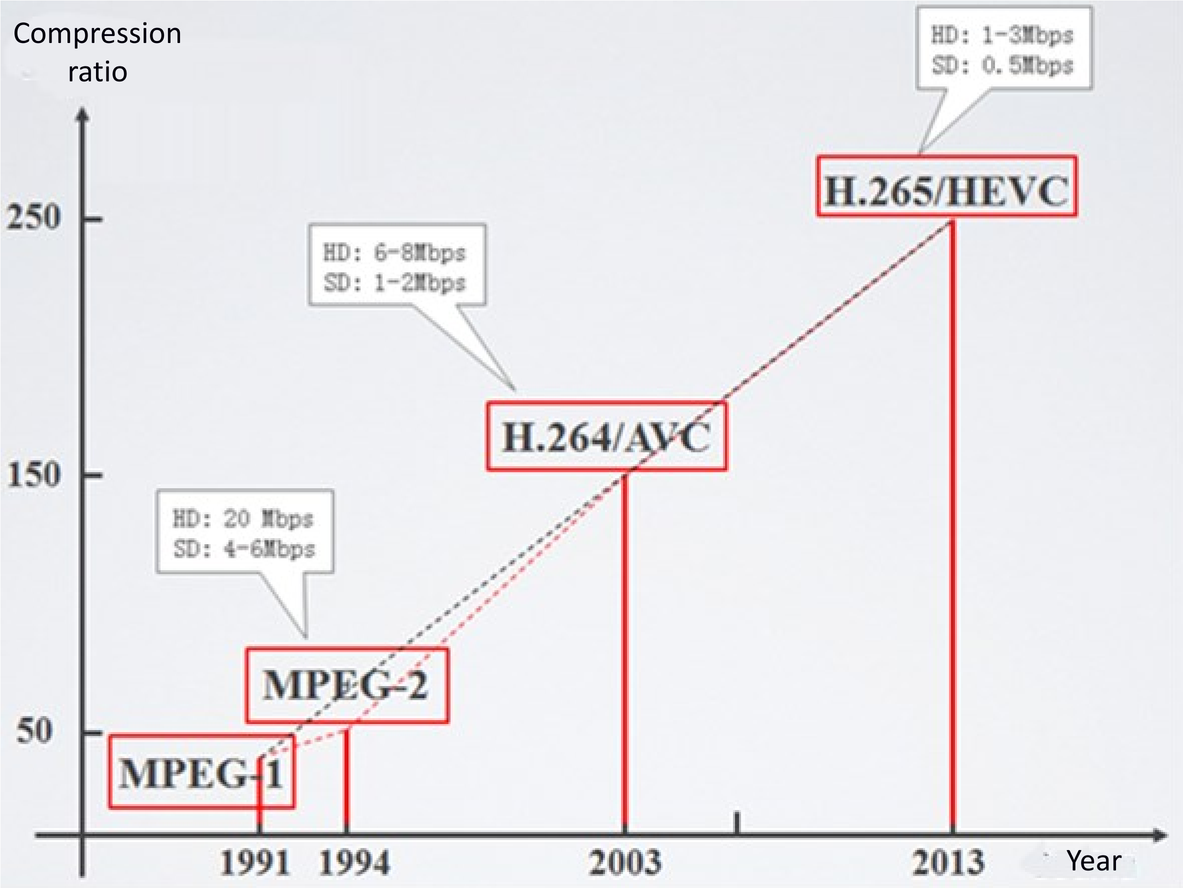


Standards for still image compression

- based on the application of transform coding
  - ISO/IEC IS 10918: *Digital Compression and Coding of Continuos Tone Still Images* - JPEG (*Joint Photographic Experts Group*)
    - DCTtemelji se na primjeni diskretne kosinusne transformacije
    - 0,25-2 bpp
  - ISO/IEC IS 15444-1: *JPEG2000 Image Coding System-Part 1: Core Coding System* - JPEG2000
    - DWT - Discrete Wavelet Transform
    - at higher compression ratios it achieves better image quality compared to JPEG

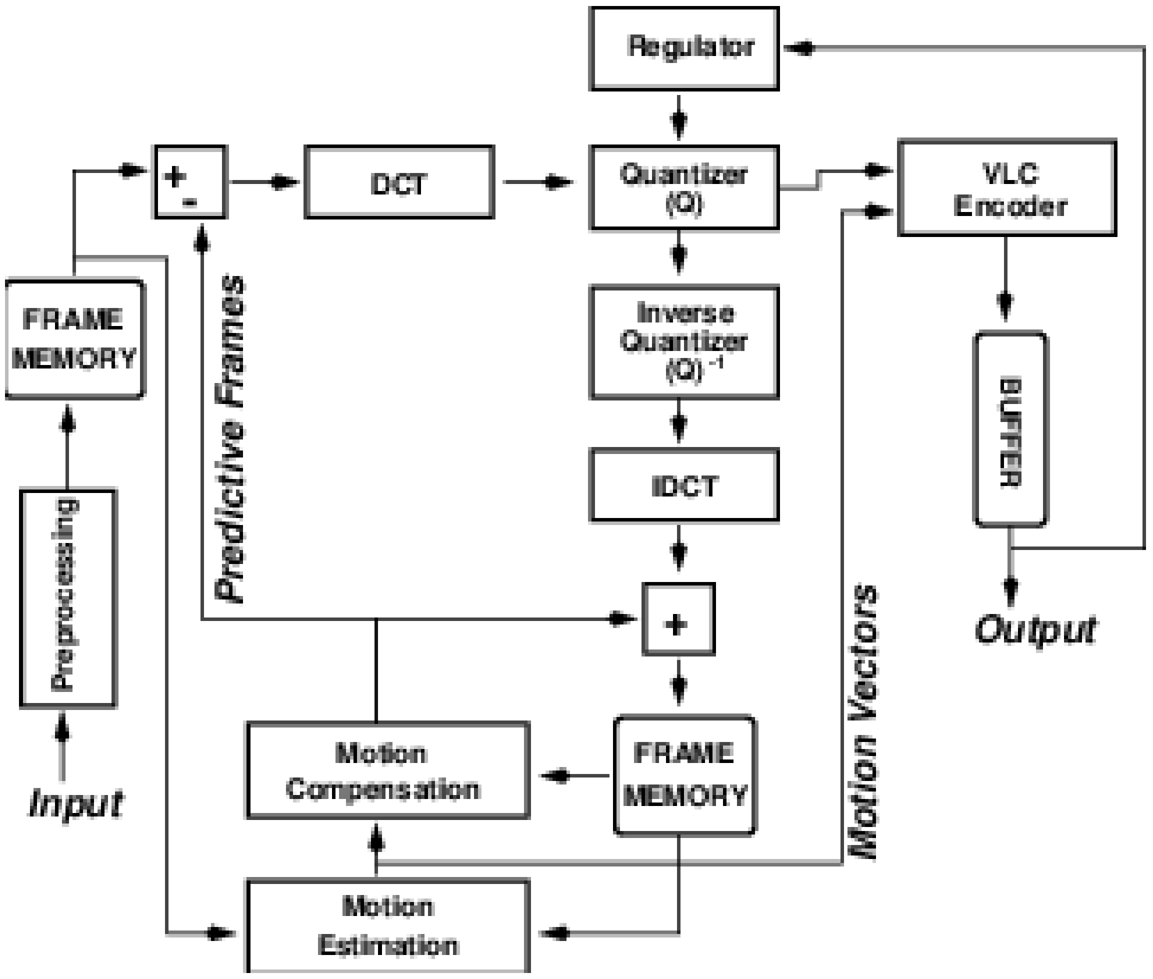
Standards for image and video compression

- required bit rates for SDTV and HDTV with different compression methods



Standards for video compression

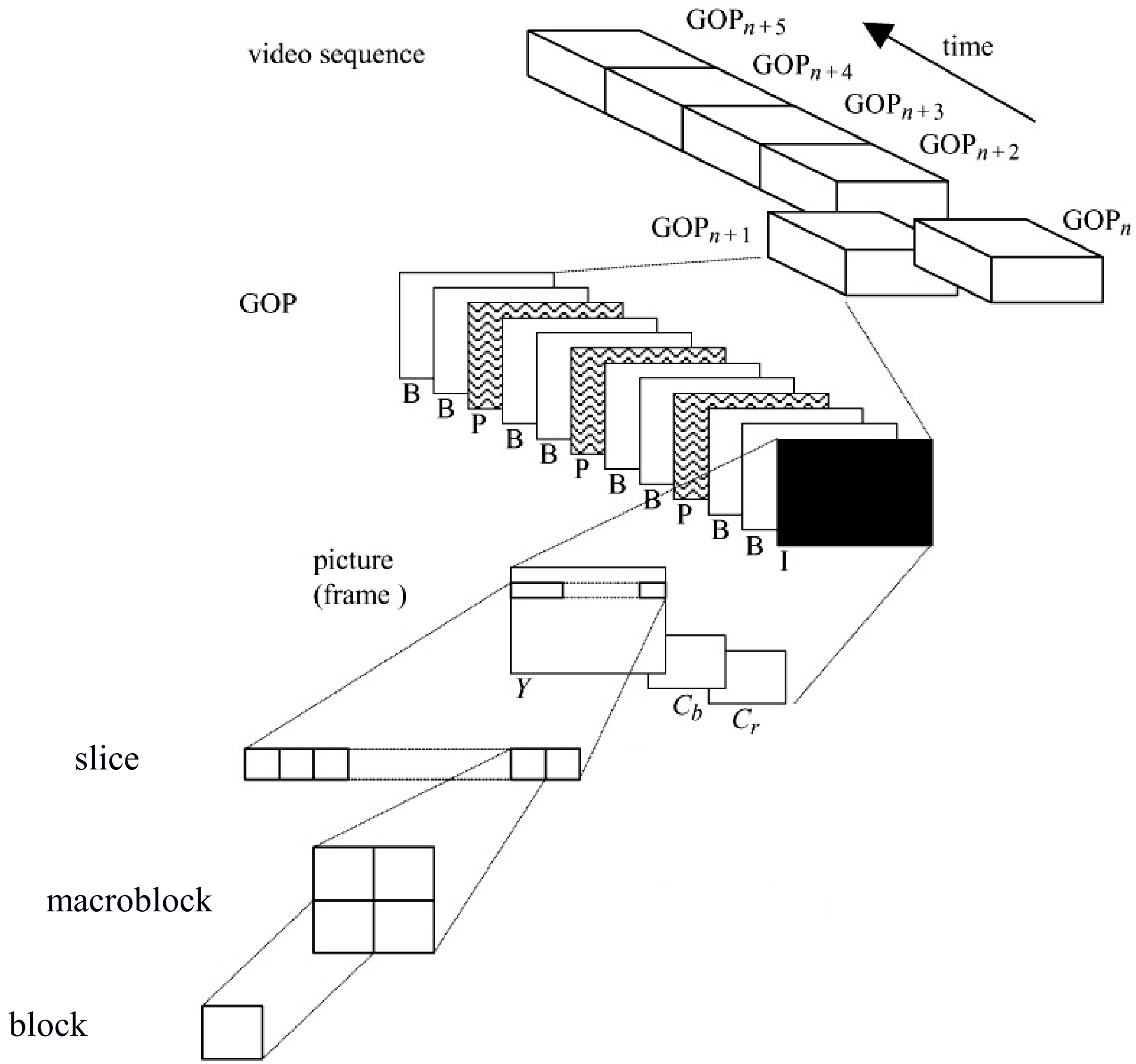
- motion compensated predictive coding (MC+DCT)
  - DCT - removes spatial redundancy in difference frame
  - DPCM with motion estimation (ME) and compensation - removes temporal redundancy



Video coding model

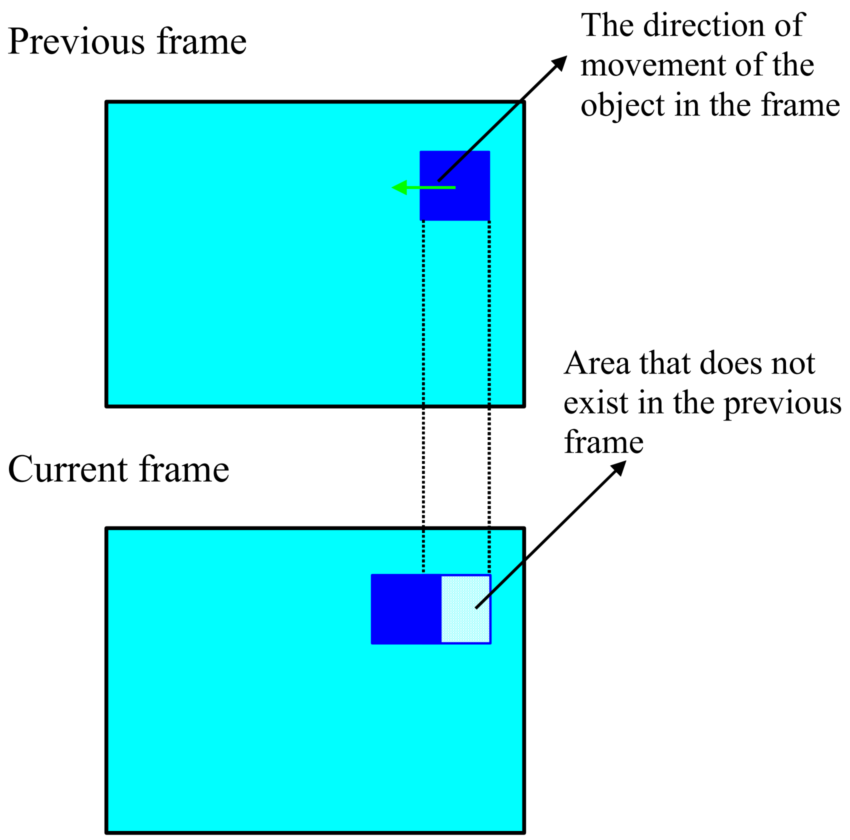
- six layers
  - block
    - smallest encoding unit (usually 8x8 or 4x4 pixels)
  - macroblock
    - basic coding unit for motion estimation and compensation (usually 16x16 pixels, but can also be 16x8, 8x16, 4x8, 8x4, etc.)
  - slice
    - horizontal sequence of macroblocks
  - picture (frame)
    - basic coding unit
  - Group of Pictures (GOP)
    - sequence of one or more frames
  - video sequence
    - succession of groups of pictures

Video coding model



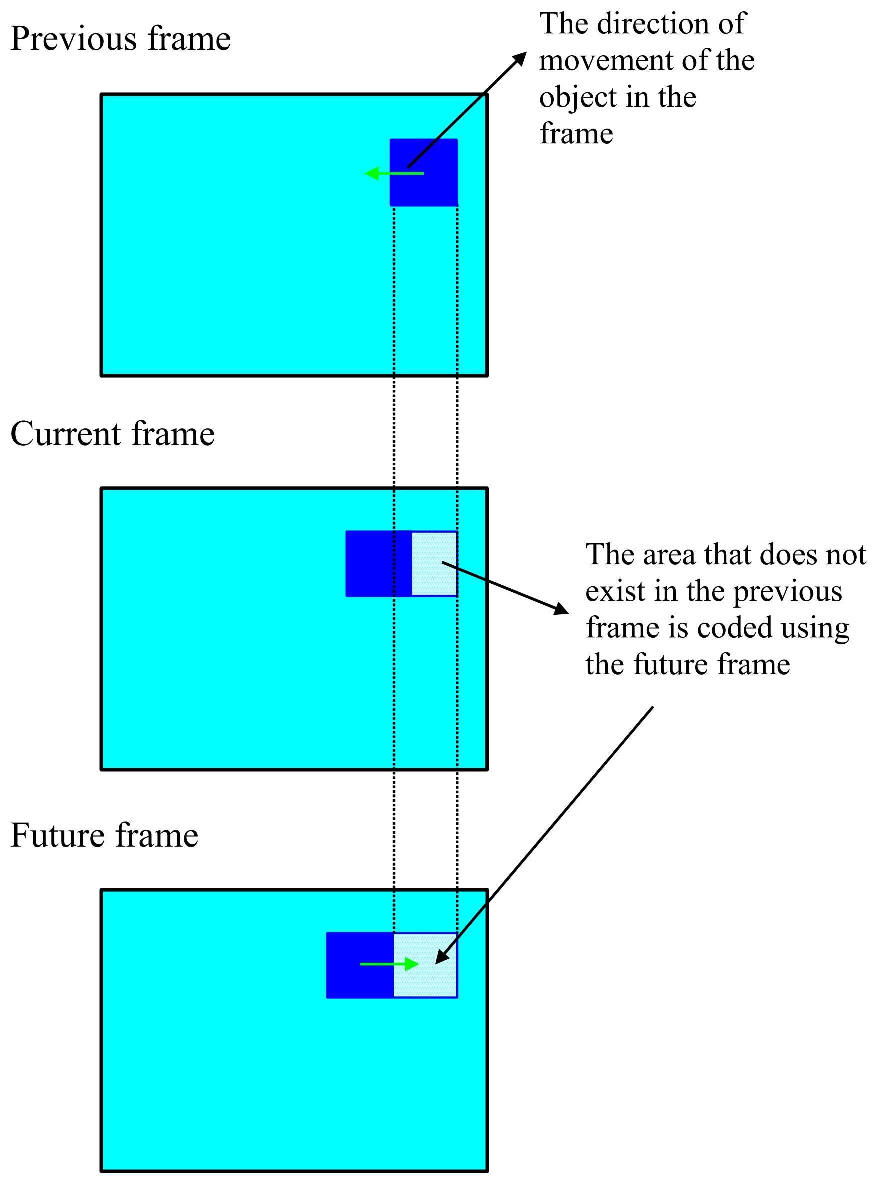
Video coding model

- P-frame
  - one-way motion prediction
  - motion prediction is performed only from the previous images
  - areas that do not exist in previous frame, exist in the difference frame and must be coded



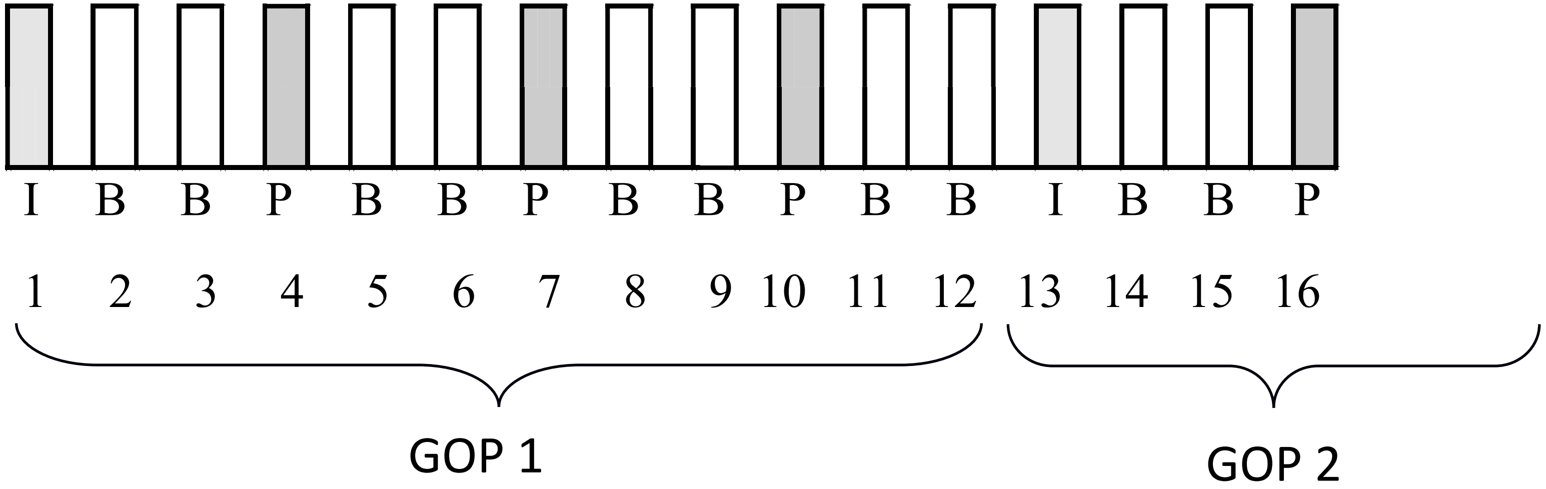
Video coding model

- B-frame
  - two-way motion prediction
  - for the B-image, motion prediction is performed from both previous and future I- or P-frame
  - the future image is used to encode areas that exist in the current farme but do not exist in the previous frame
  - higher degrees of compression are achieved compared to one-way motion prediction
  - two-way motion prediction requires changing the frame order in a group of pictures
  - the future frame must come to the decoder before than current frame
  - the future frame is needed in the process of decoding the current frame, which is coded as a B-image
  - B-frames introduces delay in coding process



Video coding model

- the display order in group of pictures consisting of 12 frames

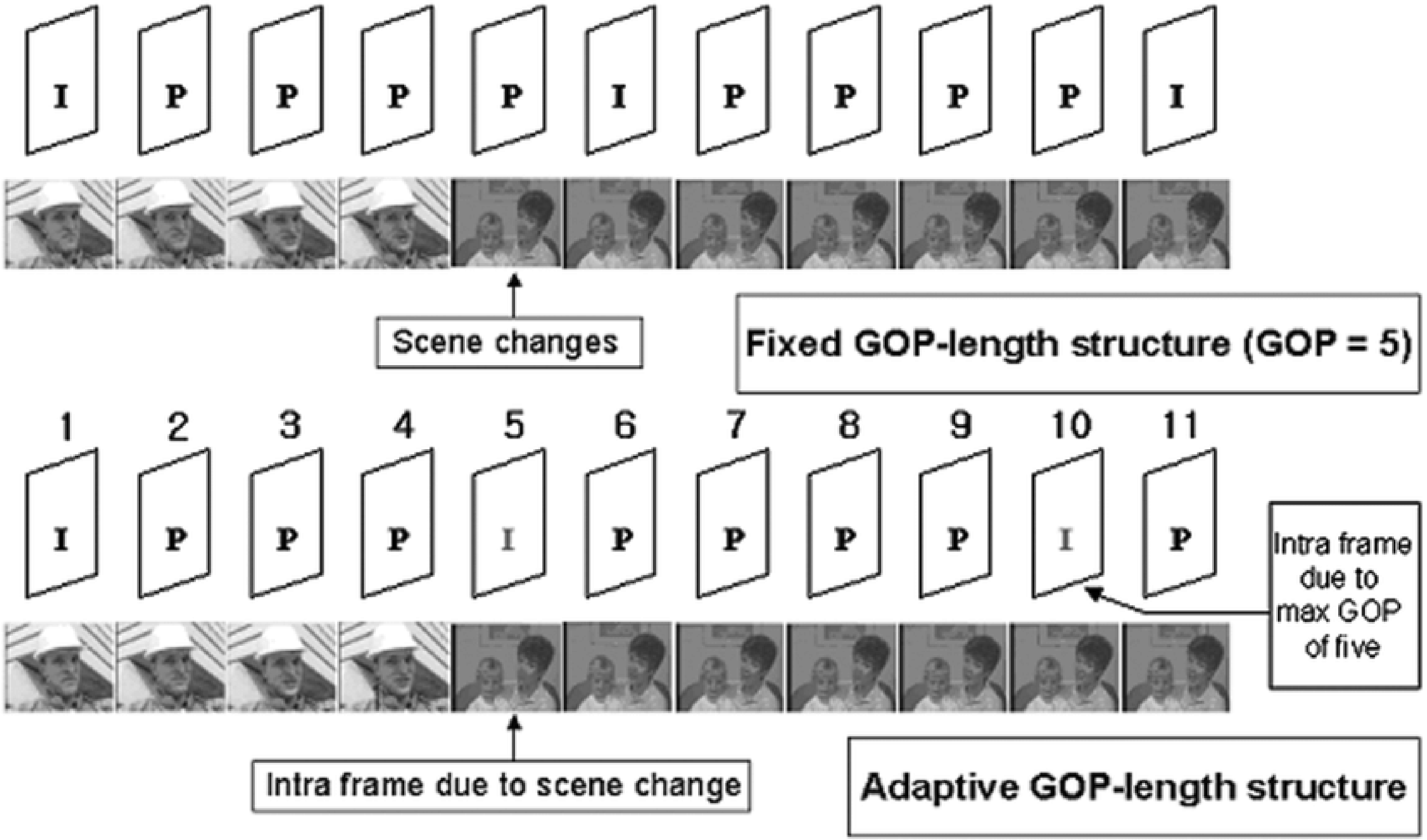


- decoding of B-images requires the presence of both past and future I- and P-frames
- the transmission order of the frames in the GOP is adjusted to the decoding process and differs from the display order

1	4	2	3	7	5	6	10	8	9	13	11	12	16	14	15
I	P	B	B	P	B	B	P	B	B	I	B	B	P	B	B

Video coding model

- in adaptive GOP, I-frame occures with scene change



Video coding model

- the dependence of image quality, bit-rate, and GOP structure
  - constant picture quality for reduced bit-rate can be preserved by changing the GOP structure
  - GOP composed of only I-frames requires higher bit-rate than a GOP composed of I- and B-frames for the same picture quality
  - a further reduction in bit-rate while maintaining the same quality level can be achieved by changing the GOP structure to IBBP
  - more B-frames in GOP means lower bit-rate while maintaining the same quality level