ECE ILLINOIS

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

ECE 417 Guest Lecture
Video Compression in MPEG-1/2/4

Min-Hsuan Tsai

Apr 02, 2013

What is MPEG and its standards

- MPEG stands for Moving Picture Expert Group
  - Develop standards for video/audio compression and transmission
- A partial list of the MPEG standards:
  - MPEG-1 (Systems, Video, Audio)
  - MPEG-2 (Systems, Video, Audio, DSM-CC)
  - MPEG-4 (Visual, Audio, AVC, File Format, AFX, Open Font Format)
  - MPEG-7 (Visual Description, Audio Description, Image and Video Signature, MPEG Query Format)

MPEG-Video

- The following parts of MPEG standards are about video compression
  - MPEG-1 part 2 (video)
  - MPEG-2 part 2 (video)
  - MPEG-4 part 2 (visual)
  - MPEG-4 part 10 (AVC)

MPEG-Video Compression

- The need for video compression
  - Using 8 bits per color channel

	lines	pixels	color channels	frame rate (frame/s)	data rate (Mbps)
CD	288	352	3	25	61
DVD	480	720	3	29.97	249
HDTV	720	1280	3	60	1327

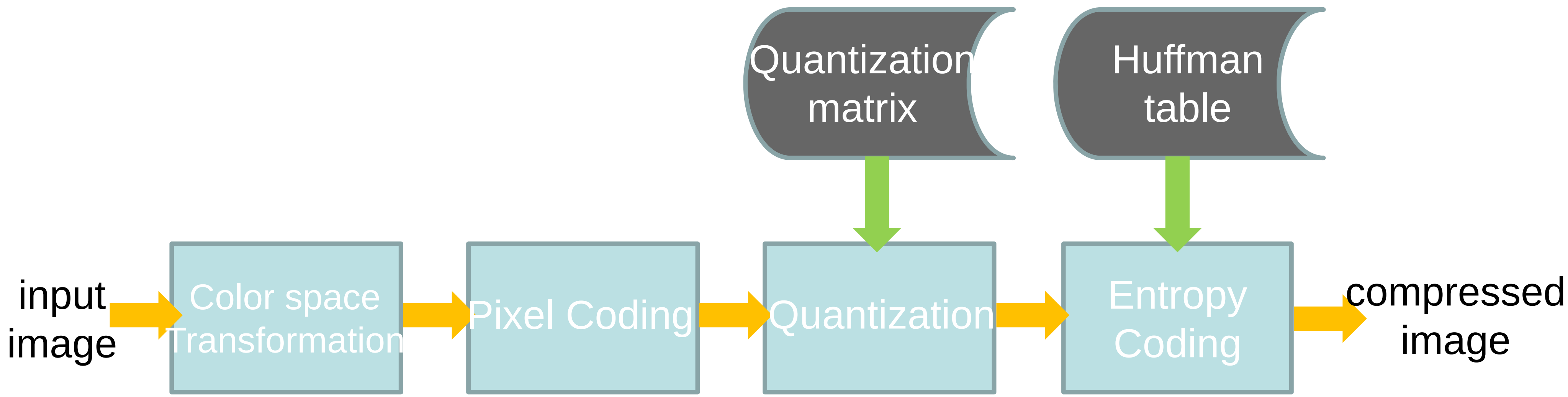
  - Some interesting bit-rates
    - TV broadcasting channel: ~20 Mbps
    - DVD: 5-20 Mbps
    - Ethernet: 10/100/1000 Mbps

MPEG-Video Compression

- Intra-frame Compression
  - Exploiting *spatial* redundancy (same as JPEG)
- Inter-frame Compression
  - Exploiting *temporal* redundancy
    - Motion estimation/ compensation

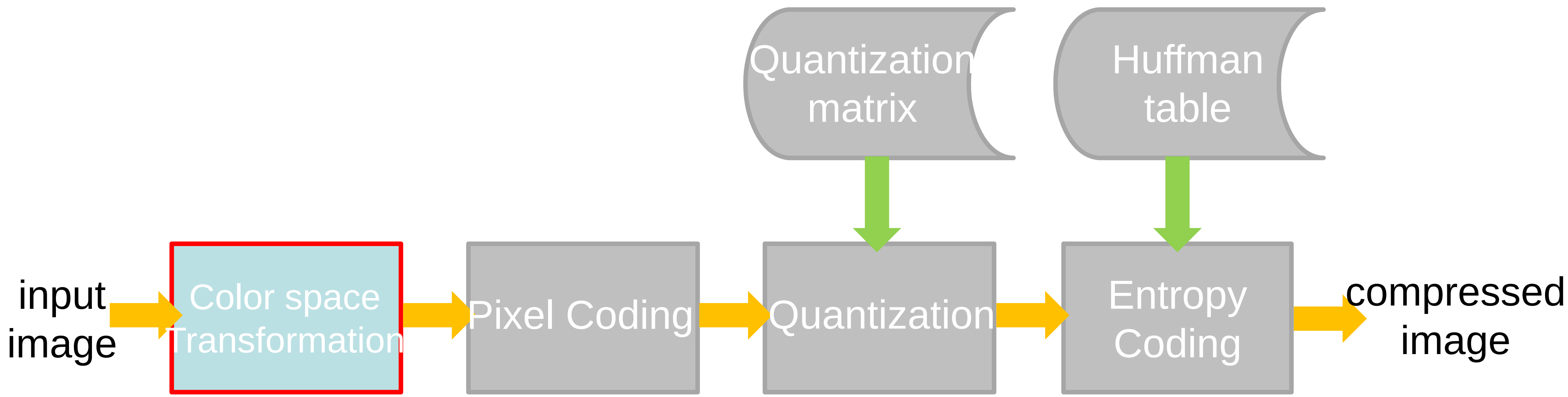
MPEG-Video Compression

- Intra-frame compression
  - Exploiting *spatial* redundancy (same as JPEG)



MPEG-Video Compression

- Intra-frame compression
  - Exploiting *spatial* redundancy (same as JPEG)



Color space transformation

- Convert RGB frames to YUV (YCbCr) color space
  - Luminance information is more important to the perception of color in the human visual system

Original image

Luminance

Chrominance (blue)

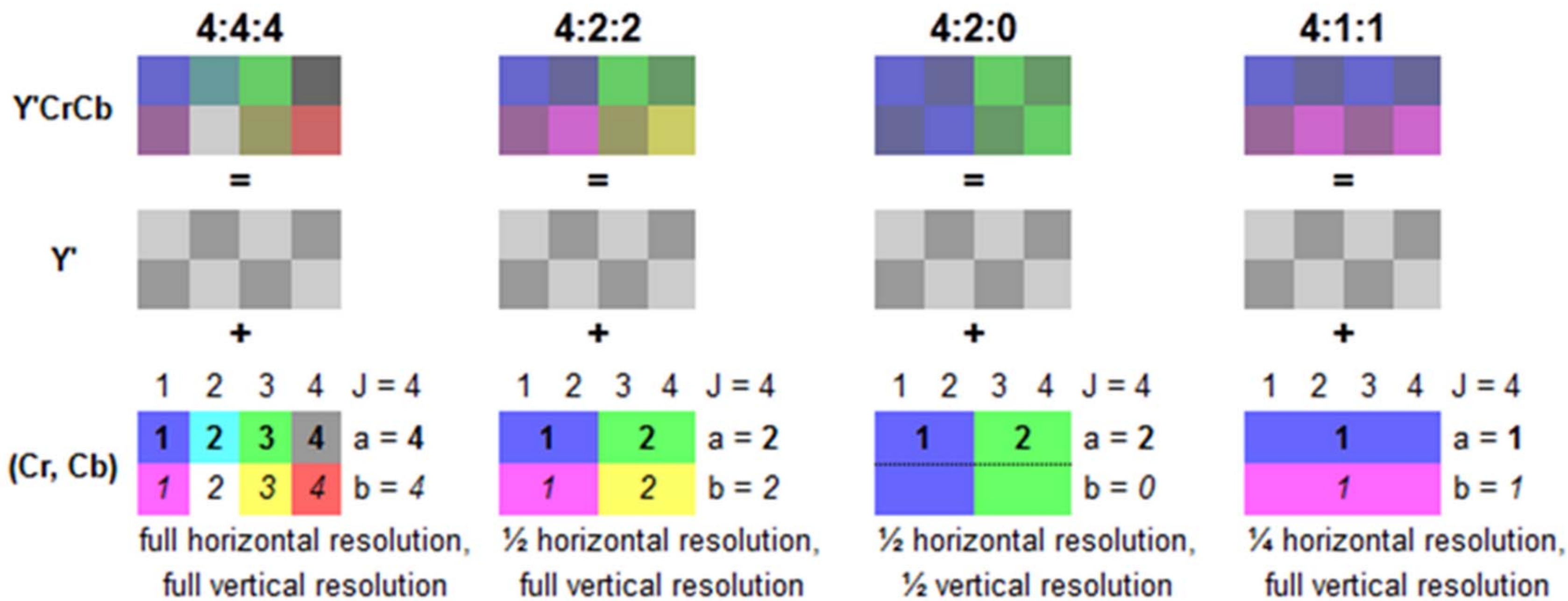
Chrominance (red)

  - Remove more chrominance info than luminance info (downsampling on chrominance components)

(Images courtesy of Wikipedia)

Color space transformation

- Chrominance subsampling (J:a:b)



J: horizontal sampling reference (width of the conceptual region). Usually, 4.
a: number of chrominance samples (Cr, Cb) in the first row of *J* pixels.
b: number of (additional) chrominance samples (Cr, Cb) in the second row of *J* pixels.

(Images courtesy of Wikipedia)

Color space transformation

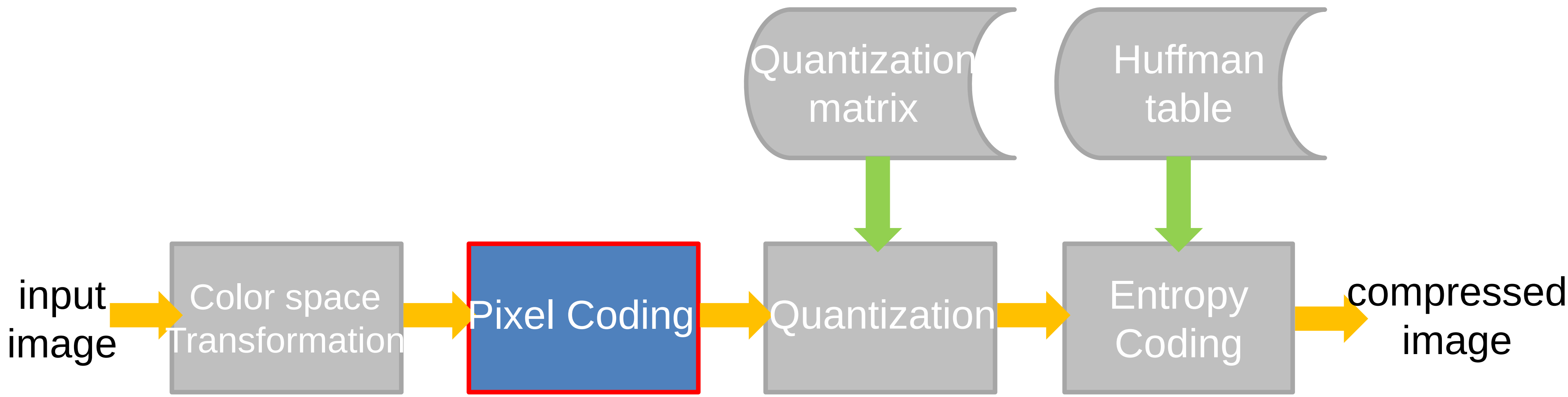
- Chrominance subsampling

4:4:4

4:2:0

MPEG-Video Compression

- Intra-frame compression
  - Exploiting *spatial* redundancy (same as JPEG)



Pixel coding

- Pixel coding using the DCT
  - Human eyes are insensitive to high frequent color changes
  - DCT converts each frame into a frequency-domain representation which has a nice strong *energy compact* property:
    - Most of the signal information tends to be concentrated in a few low-frequency components of the DCT
    - Thus DCT is often used for lossy data compression

Discrete Cosine Transformation

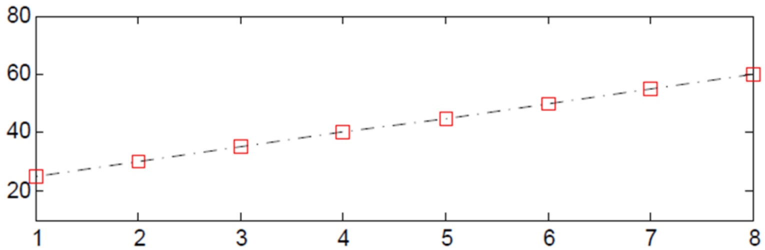
- DCT expresses a signal (x_n, n=0,…,N-1) in terms of a sum of *sinusoids* with different *freq* and *amplitudes*

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right] \qquad k = 0, \dots, N-1.$$

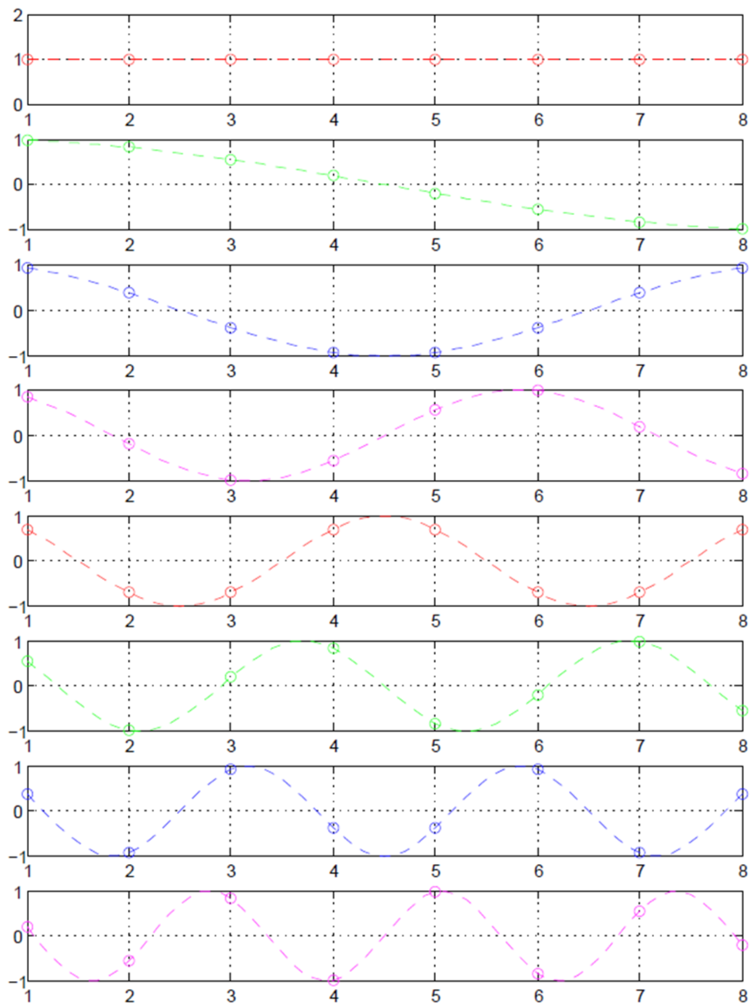
  – $\cos\left[\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right] \qquad k = 0, \dots, N-1.$ forms N basis functions

- DCT transforms the signal to a set of independent parts to reduce the entropy, so that the transformed signal can be quantized to decrease the number of bits to represent the original signal

1-D DCT



42.5	-16.1	0	-1.68	0	-0.50	0	-0.13



* 42.5

* -16.1

* 0

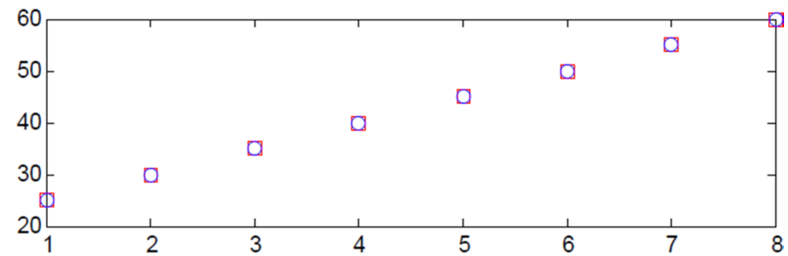
* -1.68

* 0

* -0.50

* 0

* -0.13



Perfect reconstruction
(on the dots)

1-D DCT

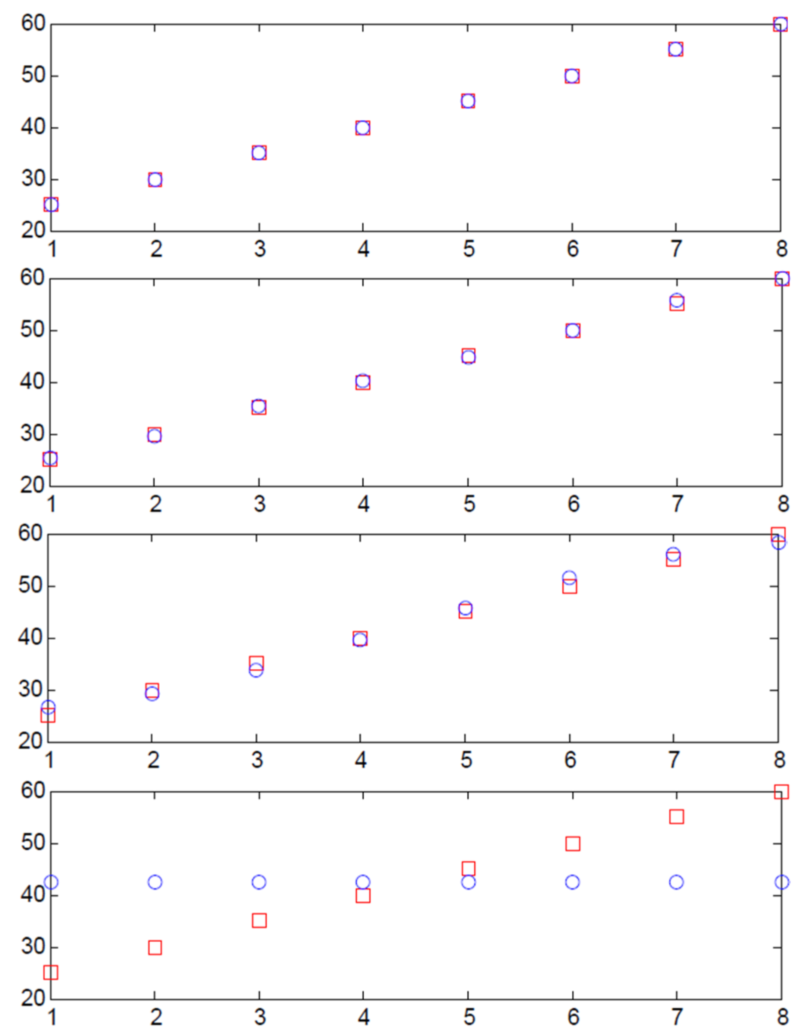
- Lossy compression

42.5	-16.1	0	-1.68	0	-0.50	~~0~~	~~-0.13~~

42.5	-16.1	0	-1.68	~~0~~	~~-0.50~~	~~0~~	~~-0.13~~

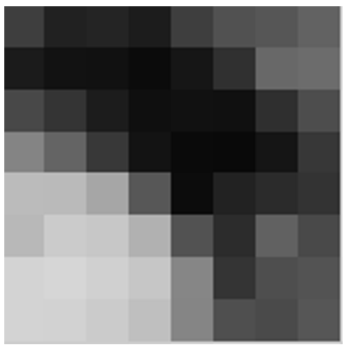
42.5	-16.1	~~0~~	~~-1.68~~	~~0~~	~~-0.50~~	~~0~~	~~-0.13~~

42.5	~~-16.1~~	~~0~~	~~-1.68~~	~~0~~	~~-0.50~~	~~0~~	~~-0.13~~



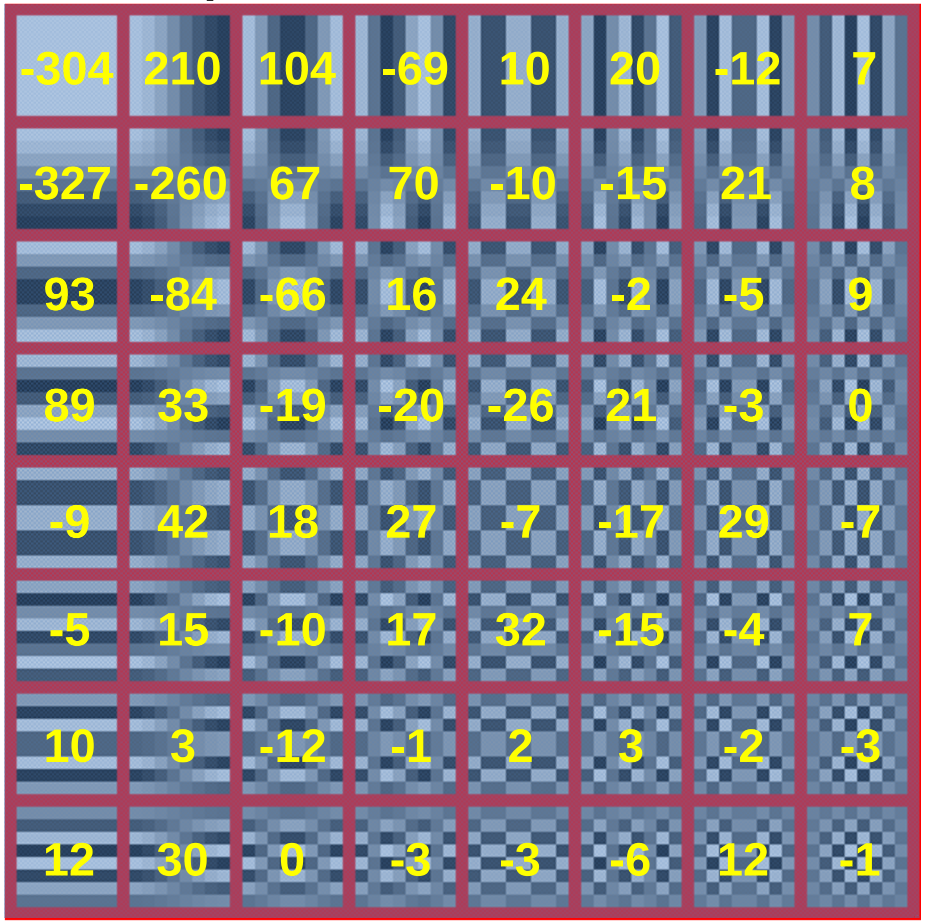
2-D DCT

- On a 8-bit image sized 8x8 (a *block*), the DCT basis functions are



63	33	36	28	63	81	86	98
27	18	17	11	22	48	104	108
72	52	28	15	17	16	47	77
132	100	56	19	10	9	21	55
187	186	166	88	13	34	43	51
184	203	199	177	82	44	97	73
211	214	208	198	134	52	78	83
211	210	203	191	133	79	74	86

DCT*

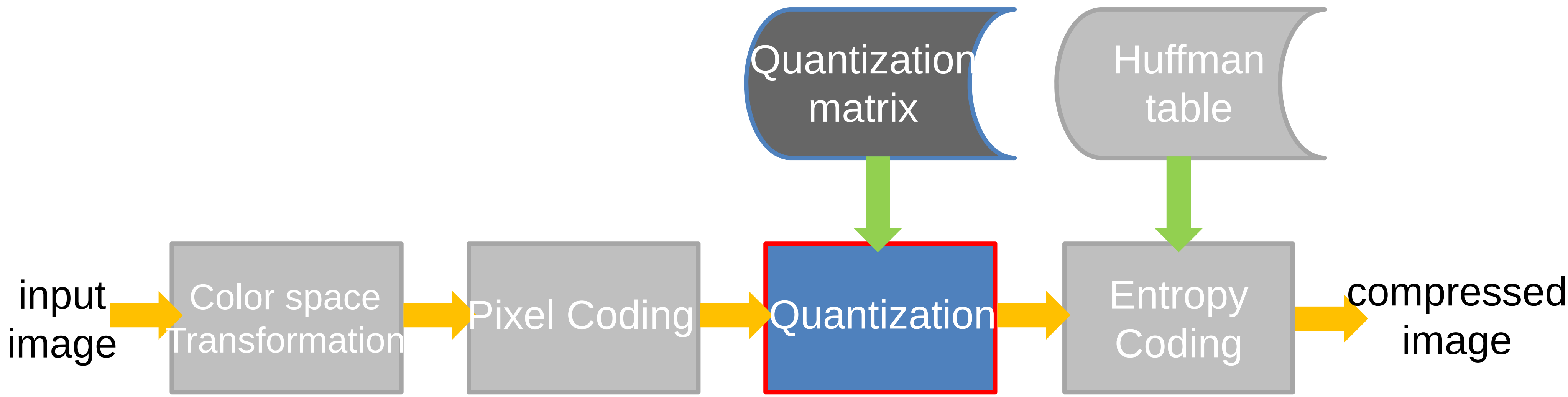


* with nearest integer truncation

(Image courtesy of Wikipedia)

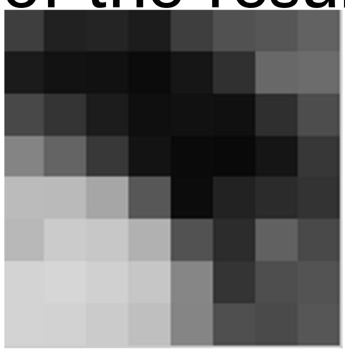
MPEG-Video Compression

- Intra-frame compression
  - Exploiting *spatial* redundancy (same for JPEG)



Quantization

- DCT tends to aggregate most of the signal in one corner of the result



I =

63	33	36	28	63	81	86	98
27	18	17	11	22	48	104	108
72	52	28	15	17	16	47	77
132	100	56	19	10	9	21	55
187	186	166	88	13	34	43	51
184	203	199	177	82	44	97	73
211	214	208	198	134	52	78	83
211	210	203	191	133	79	74	86

DCT* →

G =

-304	210	104	[illegible]	10	20	-12	7
-327	-260	67	70	-10	-15	21	8
93	[illegible]	-66	16	24	[illegible]	-5	9
[illegible]	33	-19	-20	[illegible]	21	-3	0
-9	42	18	[illegible]	-7	-17	29	-7
-5	15	[illegible]	17	32	-15	[illegible]	7
10	3	-12	-1	2	3	-2	-3
12	30	0	-3	-3	-6	12	-1

Most important

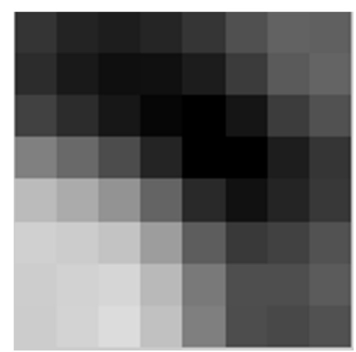
Semi important

Least important

↓ Inverse DCT*

M+S+L

M+S



M

* with nearest integer truncation

Quantization

- A better quantization scheme: quantization matrix (with loss parameter p)

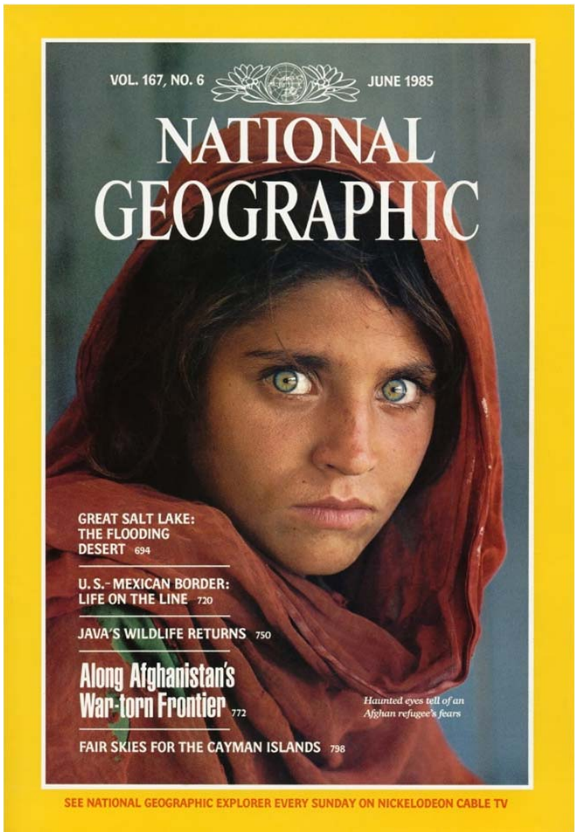
$$Q^Y(p) = p^* \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

$$G' = \text{round}(G./\, Q^Y(1))$$

$$G = \begin{bmatrix} -304 & 210 & 104 & -69 & 10 & 20 & -12 & 7 \\ -327 & -260 & 67 & 70 & -10 & -15 & 21 & 8 \\ 93 & -84 & -66 & 16 & 24 & -2 & -5 & 9 \\ 89 & 33 & -19 & -20 & -26 & 21 & -3 & 0 \\ -9 & 42 & 18 & 27 & -7 & -17 & 29 & -7 \\ -5 & 15 & -10 & 17 & 32 & -15 & -4 & 7 \\ 10 & 3 & -12 & -1 & 2 & 3 & -2 & -3 \\ 12 & 30 & 0 & -3 & -3 & -6 & 12 & -1 \end{bmatrix} \longrightarrow G' = \begin{bmatrix} -26 & -3 & -6 & 2 & 2 & -1 & 0 & 0 \\ 0 & -2 & -4 & 1 & 1 & 0 & 0 & 0 \\ -3 & 1 & 5 & -1 & -1 & 0 & 0 & 0 \\ -3 & 1 & 2 & -1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Note: Superscript Y indicates the quantization matrix is for luminance

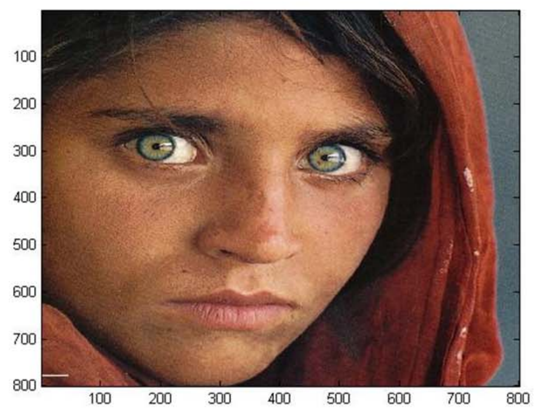
Quantization vs quality



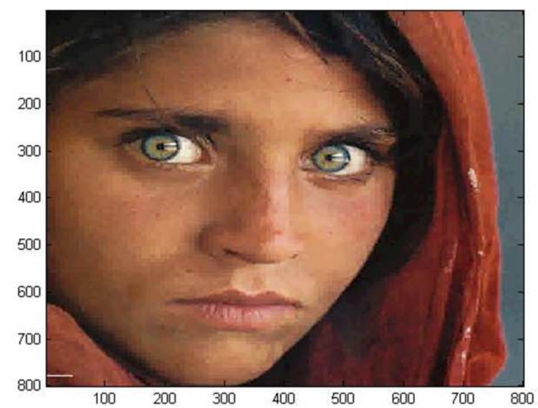
Original image

P=1

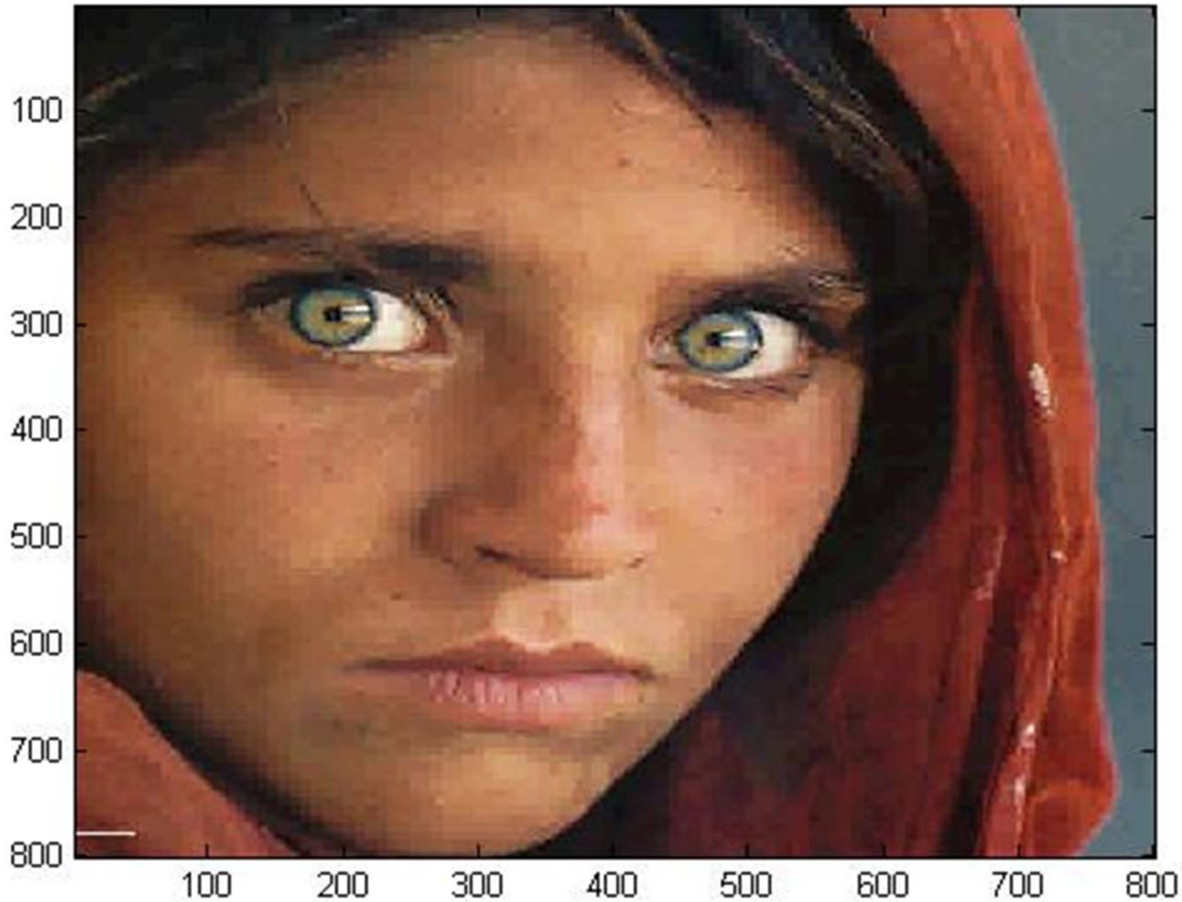
P=4



P=8



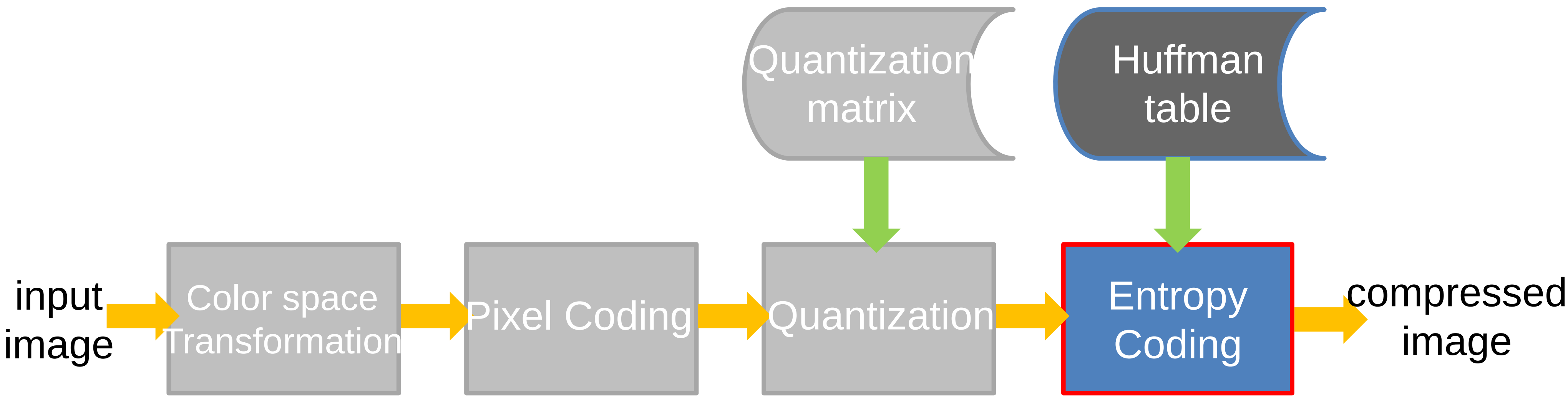
Closer look



P = 8

MPEG-Video Compression

- Intra-frame compression
  - Exploiting *spatial* redundancy (same for JPEG)



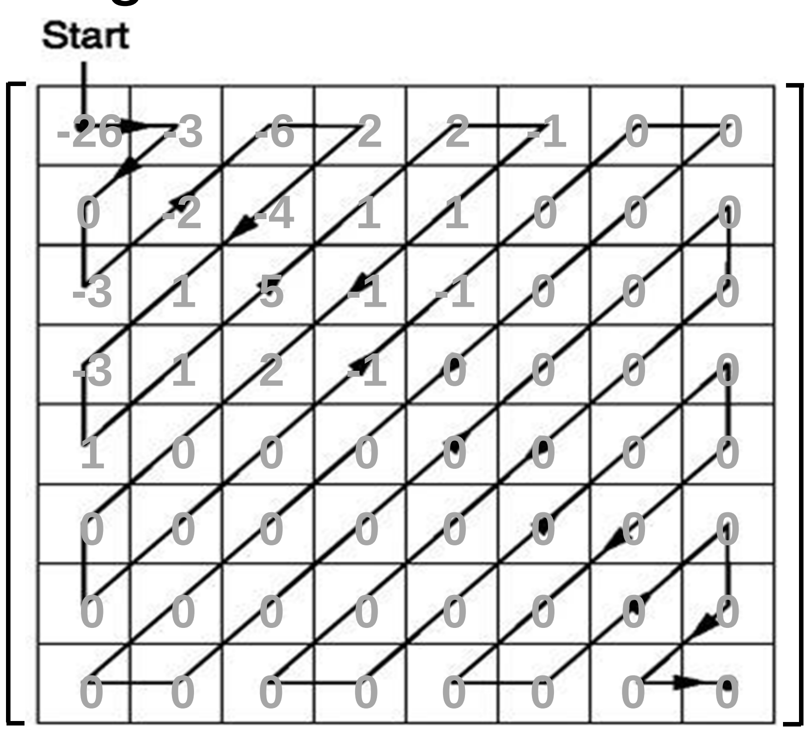
Encoding (I)

- Run-length encoding (RLE)
  - To represent a sequence of data by a single data value and count
  - E.g., aaaaabbaaaaaaabbbbbbabbbbbaaaabbb can be represented by 5a2b7a6b1a5b4a3b
  - Save lots of storage for the data sequence where the same data value occurs in many consecutive elements
    - For 2-D cases, need to smartly vectorize G* for RLE (how?)

How to vectorize G*

- Such that the resulting sequence can be better compressed w.r.t. RLE
- Zigzag Sequencing

G* =



Encoding (II)

- Entropy coding
  - Given a set of symbols and their weights (e.g., probabilities), find a *prefix-free* and *variable-length* code with minimum expected #bits needed
    - Prefix-free code: the bit string representing some particular symbol is never a prefix of the bit string representing any other symbol
    - Variable-length code (VLC)

	a	b	c	d	Cost (#bits)
probability	0.05	0.6	0.3	0.05	
Fixed-length	00	01	10	11	2.00
Variable-length	111	0	10	110	1.50

Encoding (II)

- Entropy coding
  - Any prefix free code is uniquely decodeable
    - 111010110100111
    - 111010110100111
    - abcdcba

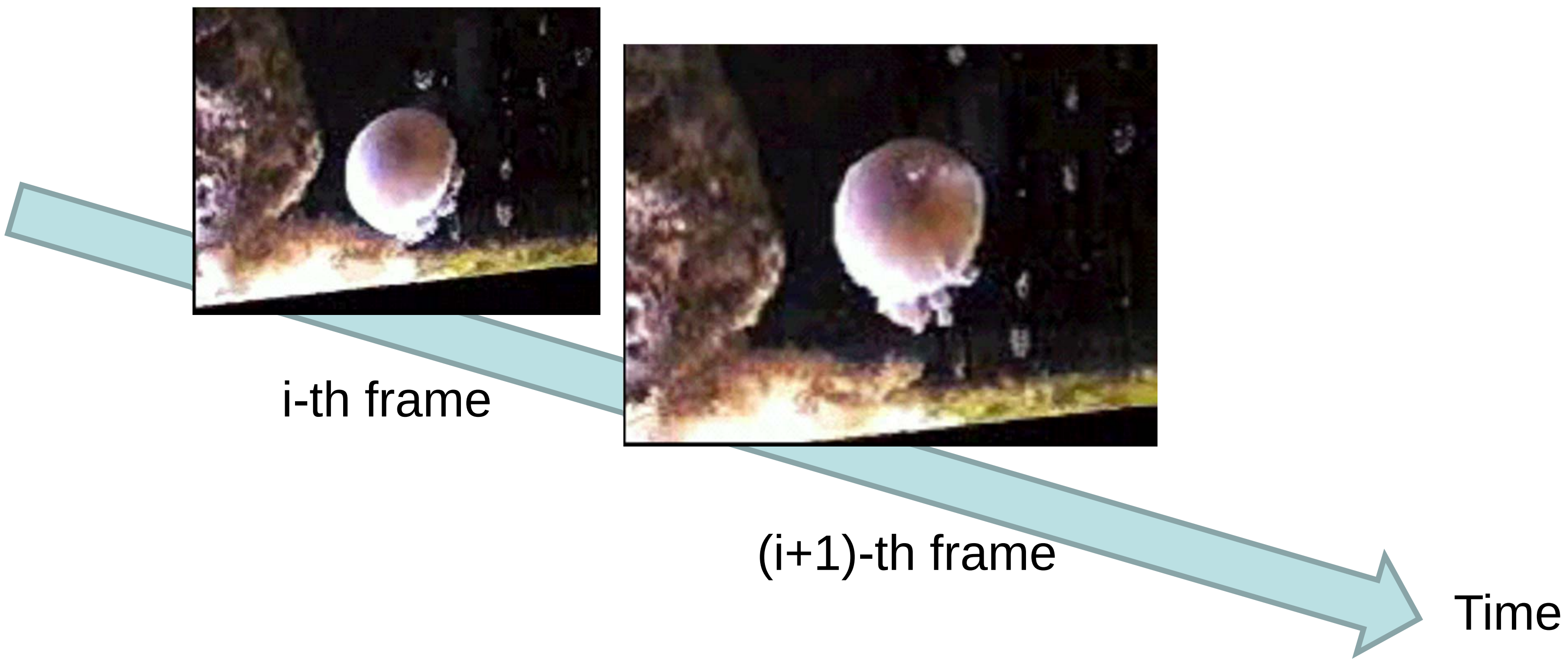
Symbol	a	b	c	d
Code	111	0	10	110

  - Huffman coding is a greedy algorithm for producing a variable-length prefix code with
    - Optimality guaranteed[1]
    - Efficient construction time (O(nlogn))

[1] When the symbols are independently distributed

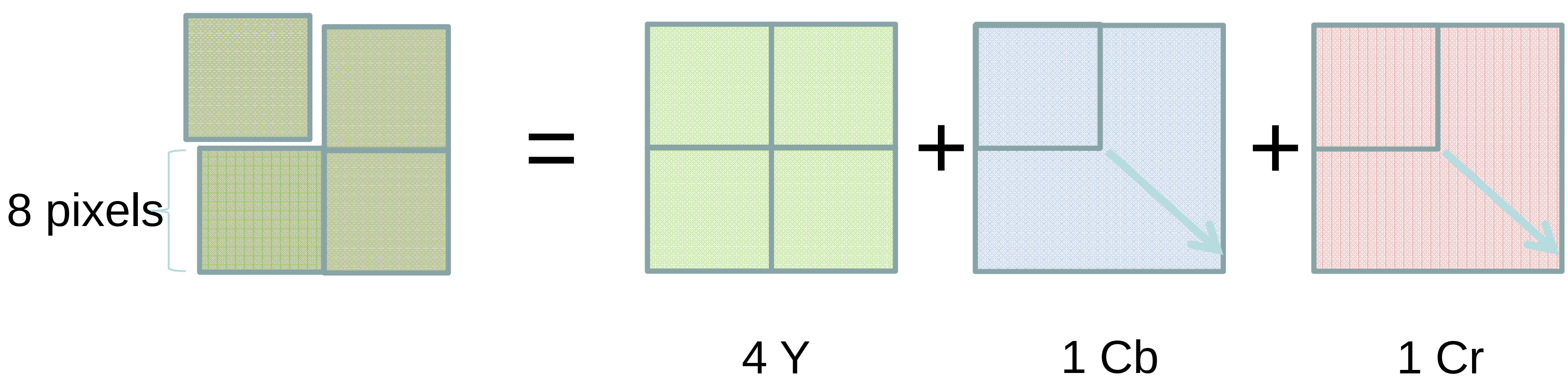
MPEG-Video Compression

- Inter-frame compression
  - Exploiting *temporal* redundancy



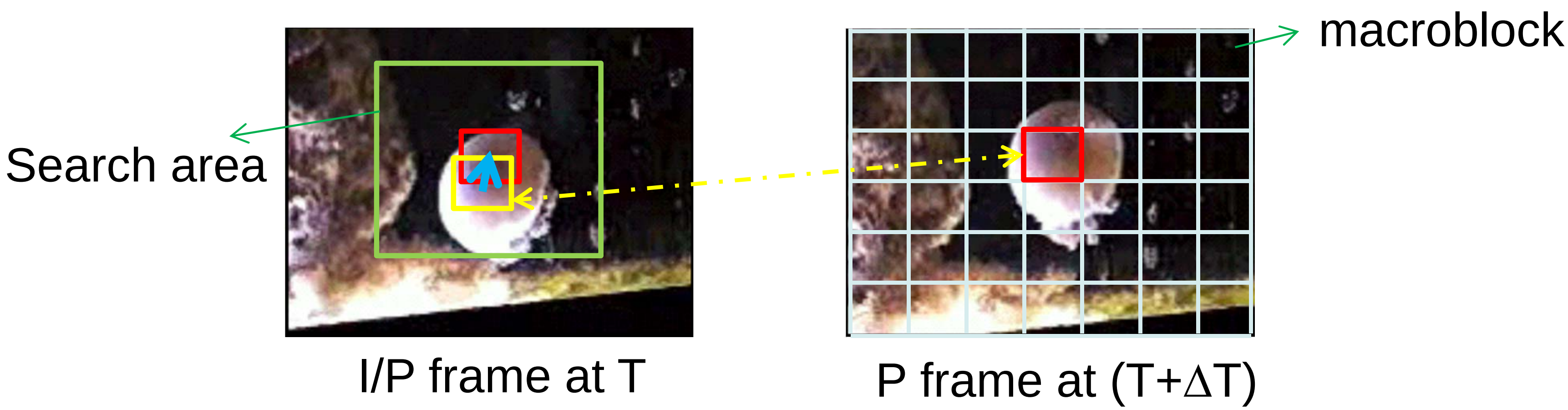
MPEG-Video Compression

- Exploiting temporal redundancy
  – Macoblocks: composed of several (usually a multiple of 4) blocks of pixels
    - Typical macroblock for a 4:2:0 color space



Motion estimation

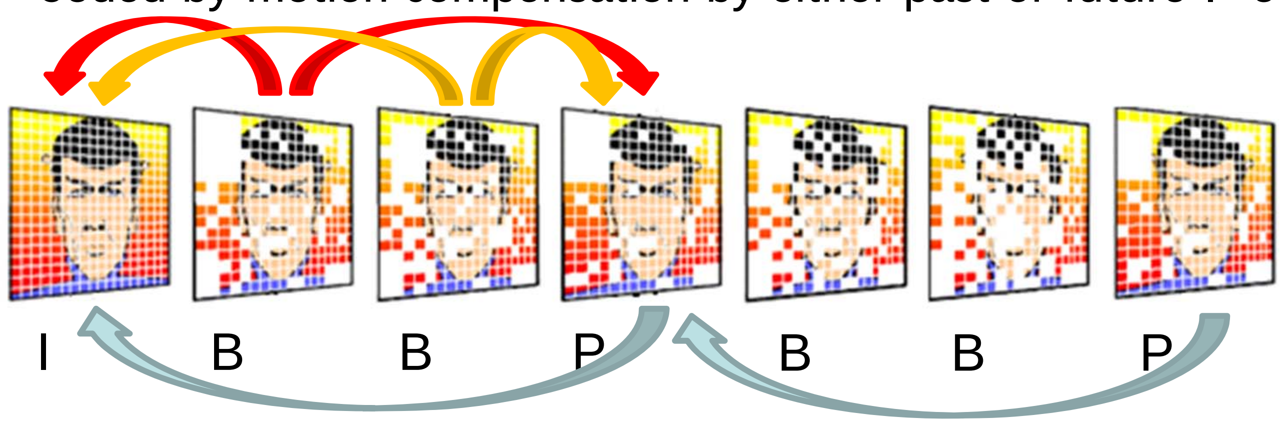
- Motion estimation (at the macroblock level)



  - Motion vector
    - Used to indicate where exactly to get the data in the reference frame(s)
    - The motion vectors are up to sub-pel accuracies. E.g., half-pel in MPEG-1/2 and quarter-pel in MPEG-4
    - Motion vector would be zero for a static background
    - MPEG does not specify the motion estimation algorithm

MPEG-Video frames

- Types of video frames
  - I-frame (intra)
    - Compressed using only *intra-frame coding*
    - Moderate compression but faster random access
  - P-frame (predicted)
    - Coded with motion compression using past I- or P-frames
    - Can be used as reference pictures for additional motion compensation
  - B-frame (bidirectional)
    - Coded by motion compensation by either past or future I- or P-frames

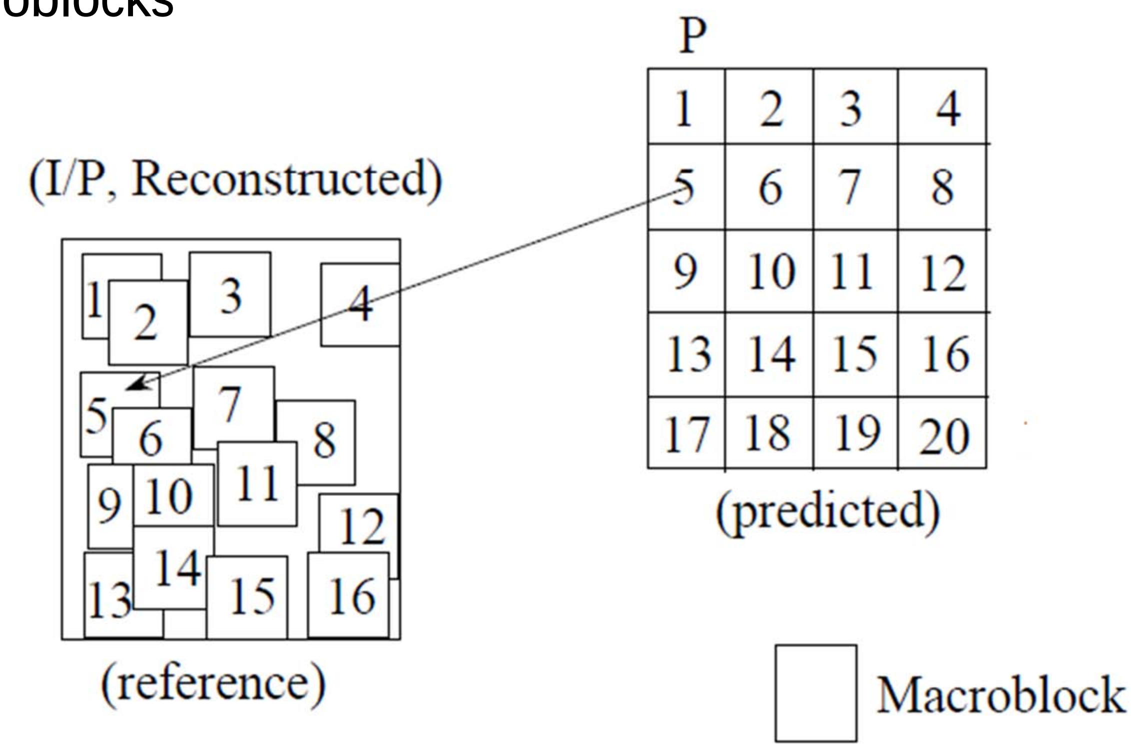


Motion estimation

- Inter-frame predictive coding (P-frames)
  - For each macroblock the motion estimator produces the best matching macroblock
  - The two macroblocks are subtracted and the difference is DCT coded
- Inter-frame interpolative coding (B-frames)
  - The motion vector estimation is performed twice
  - The encoder forms a prediction error macroblock from either or from their average
  - The prediction error is encoded using a block-based DCT
- The encoder needs to reorder pictures because B-frames always arrive late

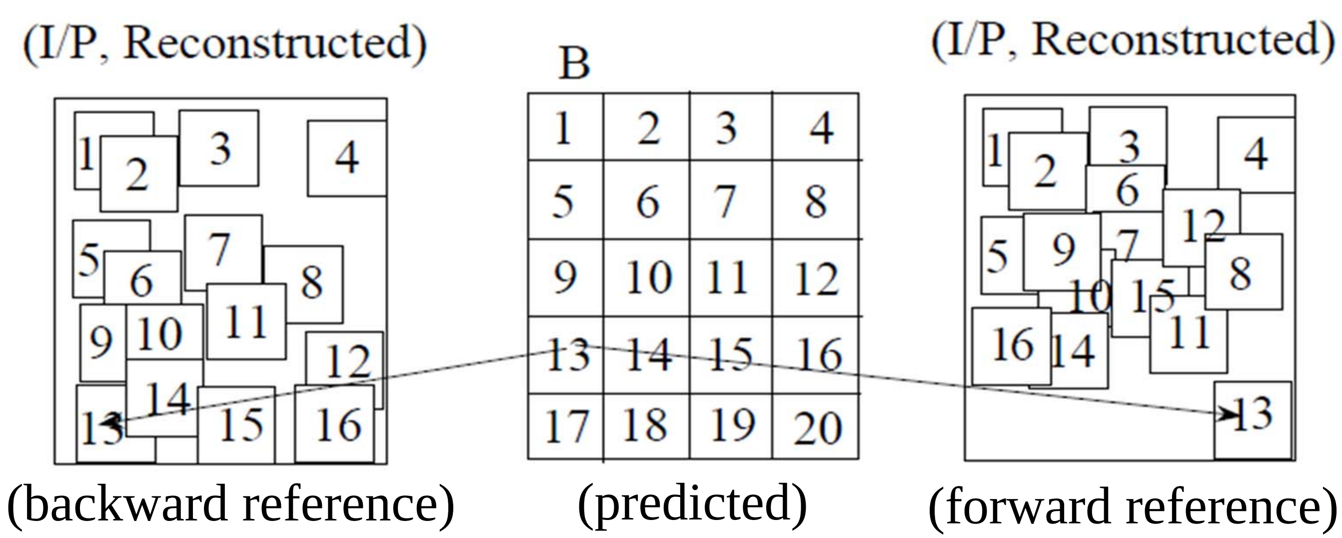
Motion compensation

- P-frames
  - Macroblocks can be reconstructed by
    - Forward prediction (with a motion vector)
    - Intra coded macroblocks



Motion compensation

- B-frames
  - Macroblocks can be reconstructed by
    - Bi-directional prediction (with two motion vector)
    - Forward prediction (with a motion vector)
    - Backward prediction (with a motion vector)
    - Intra coded macroblocks



Group of pictures (GOP)

- A group of successive pictures within a coded bit stream
- The structure of GOP specifies the order in which different types of pictures are arranged
  - Always starts with an I-frame, followed usually by a number of P-frames and B-frames
  - Characterized by two parameters: *GOP size M* (distances between anchor frames) and *GOP length N* (distances between I-frames)
  - Can be as small as a single I-frame (e.g., IMAX)
  - usually allow only 1 I-frame in a GOP
  - E.g., IBBPBBPBBPBB (M=3, N=12)

Structure of GOP

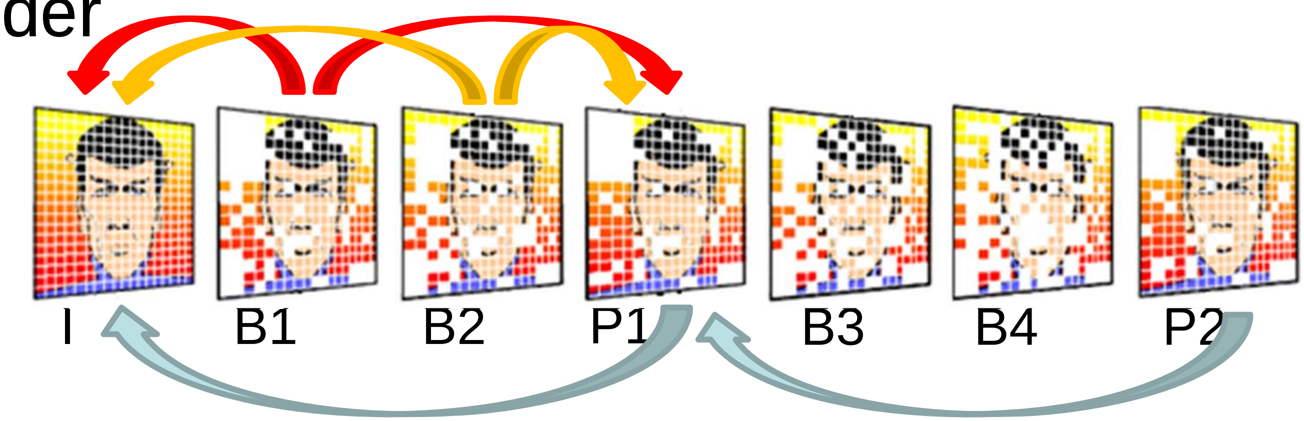
- The structure of GOP determines the compression ratio
  - For MPEG-1 NTSC with the GOP structure (M=3, N=12), there are 1 I-frame, 3 P-frames and 8 B-frames: IBBPBBPBBPBB

Type	Size	Compress
I	18K	7:1
P	6K	20:1
B	2.5K	50:1
Avg	4.8K	27:1

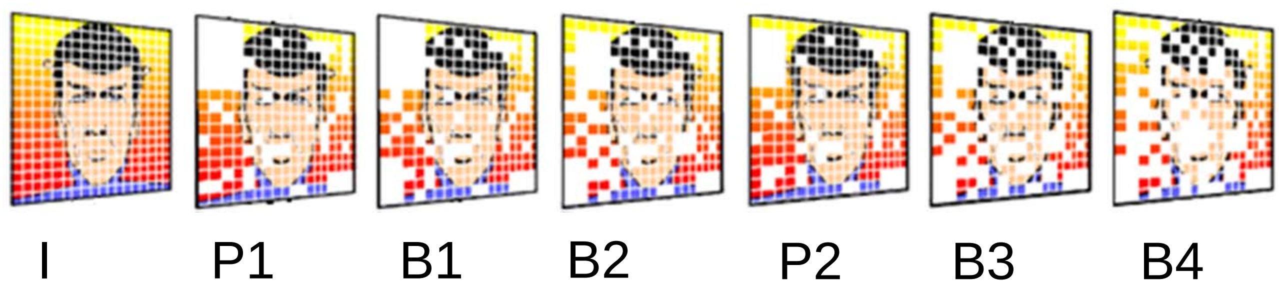
- Other examples of GOP length used in common MPEG formats:
  - MPEG-2 for DVD: (M=3, N=18) for NTSC or (M=3, N=15) frames for PAL.
  - 1080-line HDV: (M=3, N=15)
  - 720-line HDV: (M=3, N=6)
  - IMX: Uses only I-frames

Ordering frames for decoding

- Presentation order

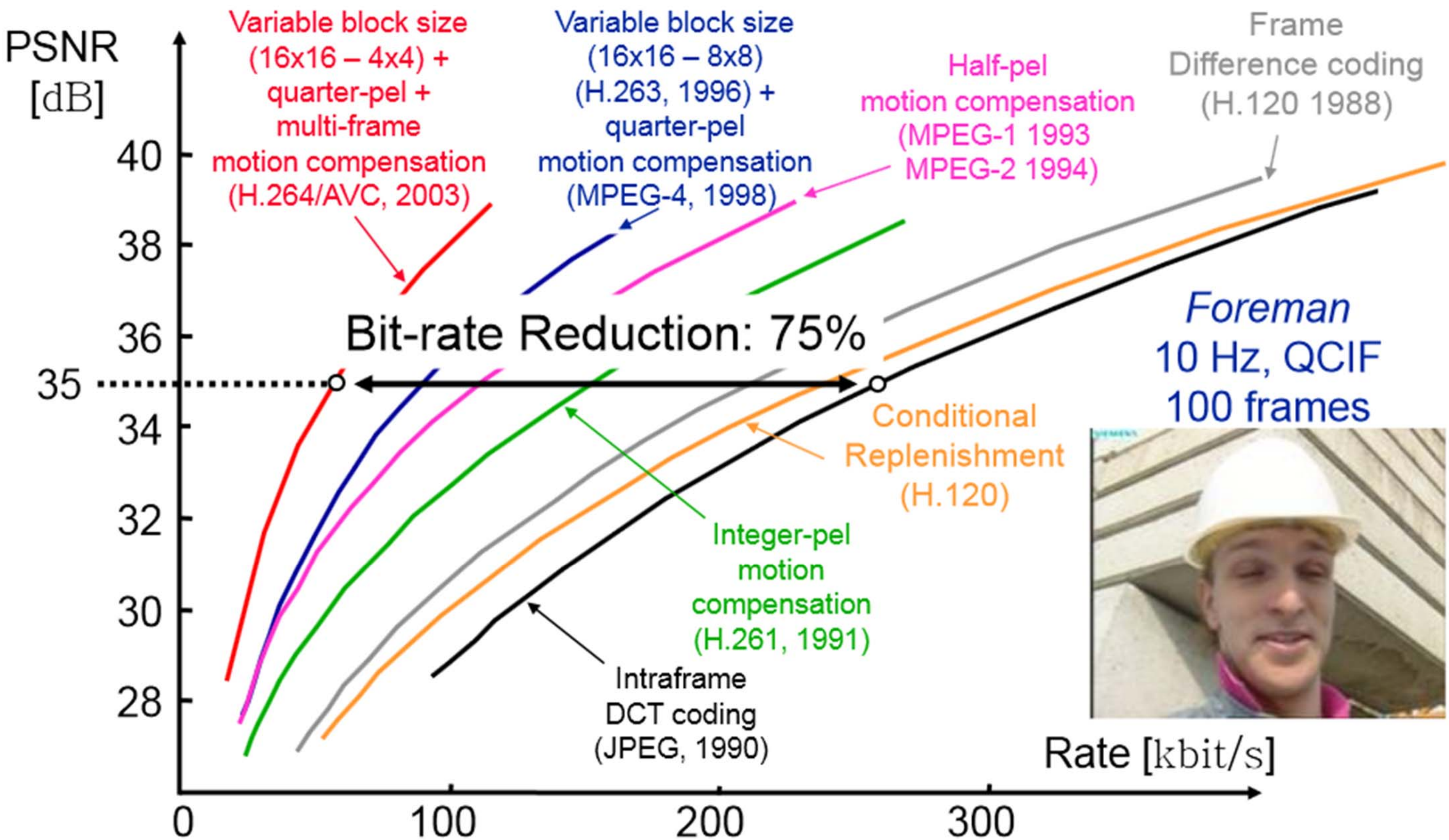


- Coding order



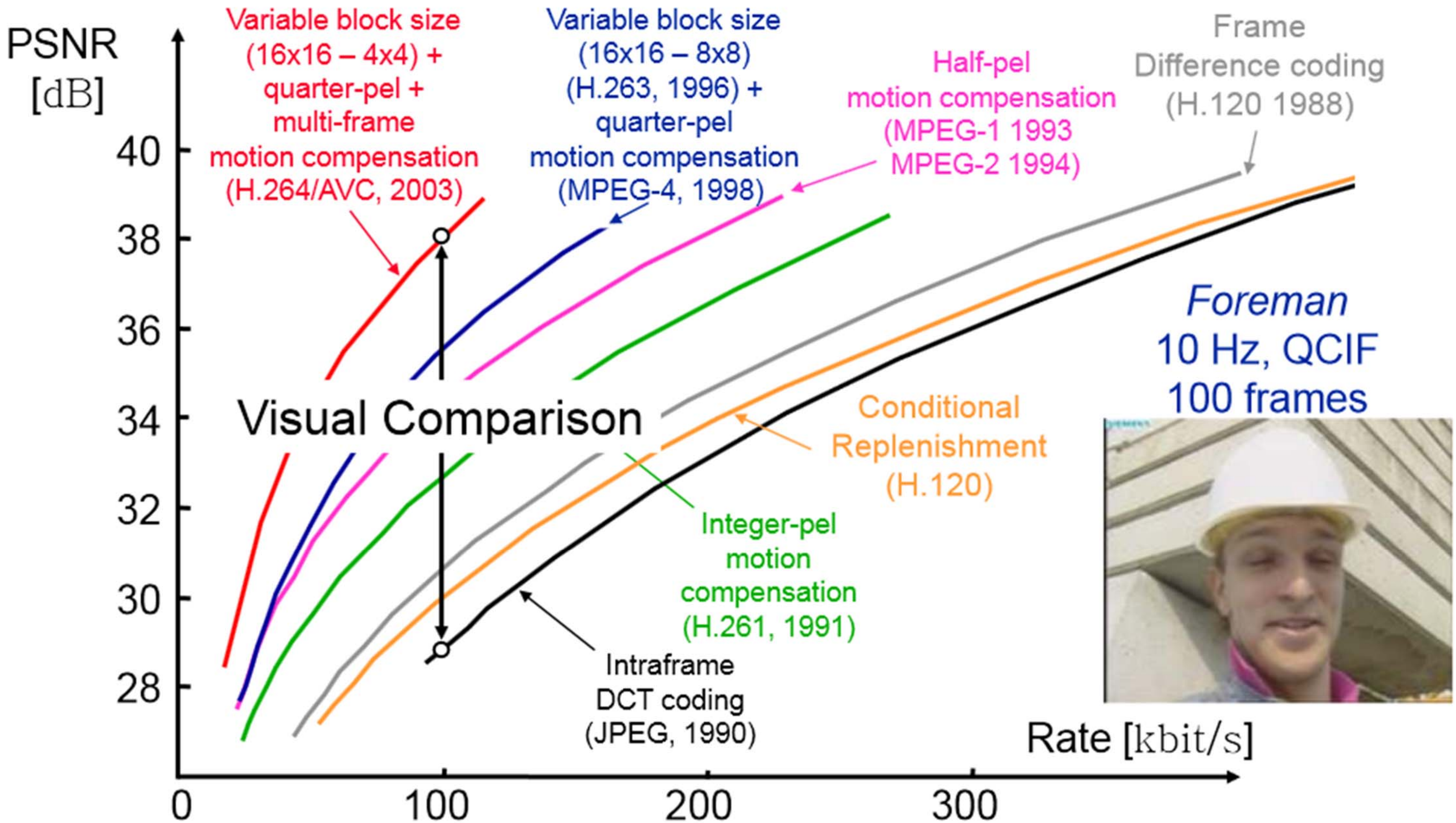
Decode Time Stamps (DTS)	0	1	2	3	4	5	6
Presentation Time Stamps (PTS)	0	3	1	2	6	4	5

Milestones for video compression



(slide courtesy of Wiegand and Girod)

Milestones for video compression

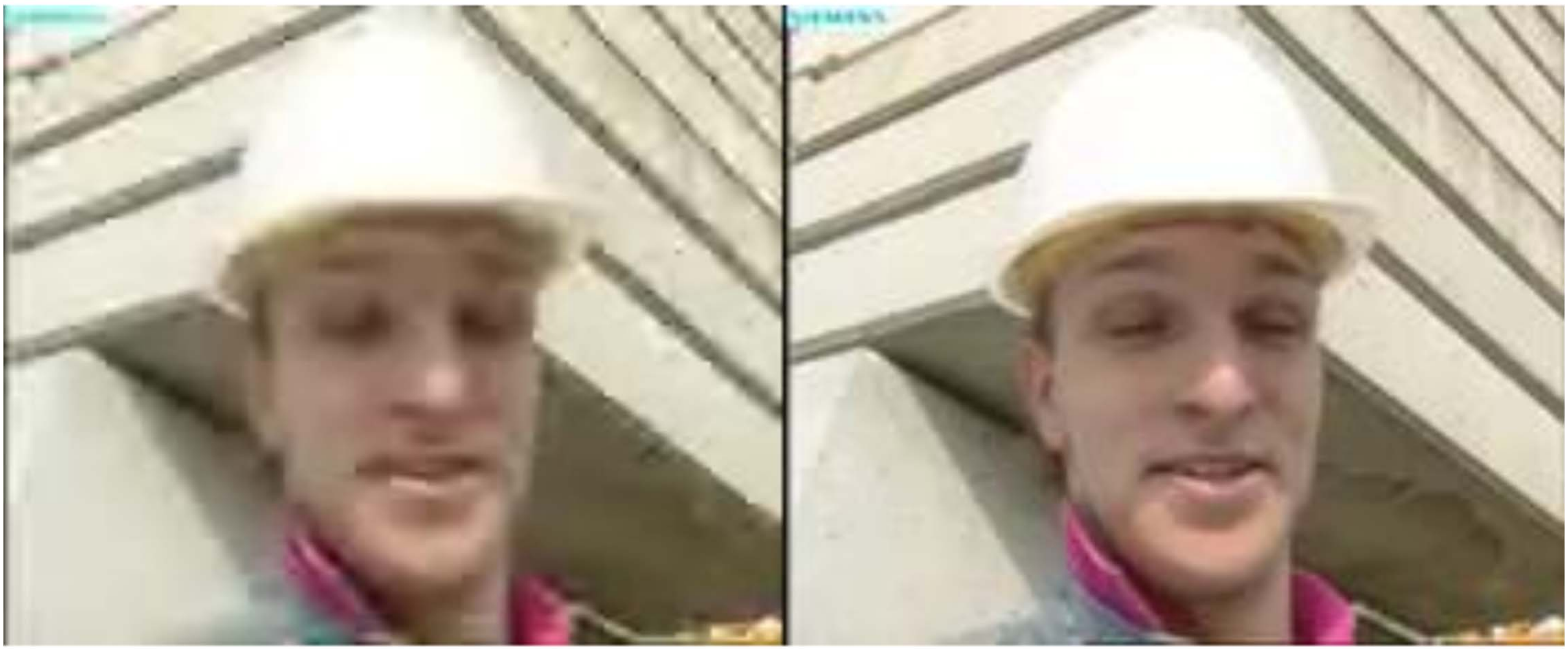


(slide courtesy of Wiegand and Girod)

Visual comparison

Foreman, QCIF, 10 Hz, 100 kbit/s

JPEG H.264/AVC



(slide courtesy of Wiegand and Girod)