Digital television

Audio and Video compressing - MPEG

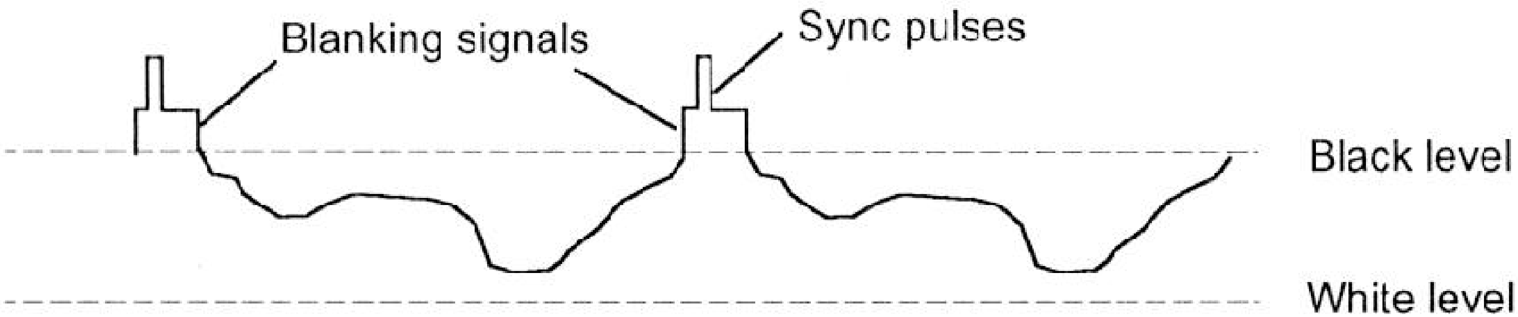
* MPEG Motivation / History

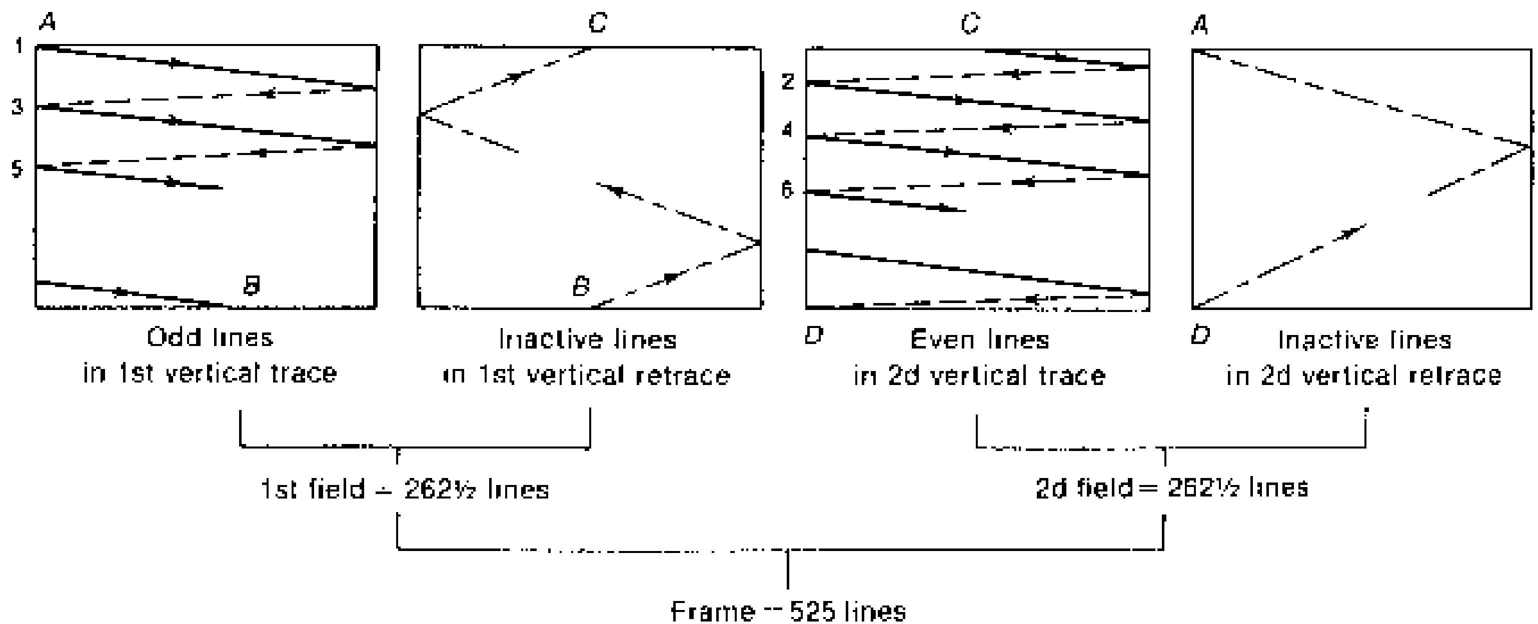
* Overall structure

* Fundamental blocks: DCT, Motion estimation,
  Entropy coding

* MPEG stream

MPEG motivation





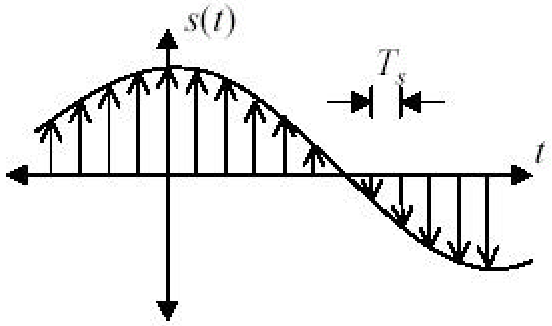
Digital TV: Sampling

$s[k] = s(kT_s)$

$k \in \{..., -1, 0\ 1, 2, 3,...\}$

$1/f_s = T_s$ sampling interval

Nyqvist rate $2\ f_{max}$



Digital TV: Sampling

Sampling of video signal (composite video):

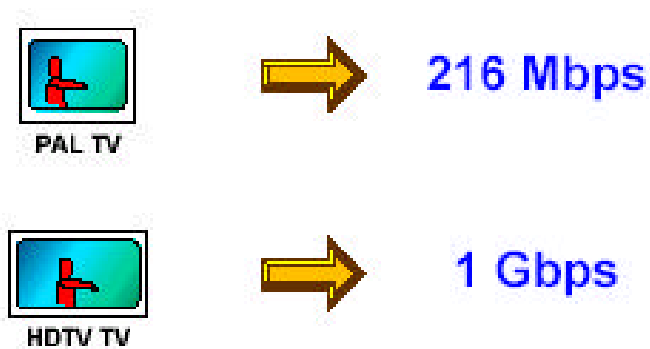
Normally ”4fsc” 4 * colorsubcarrier frequency

-> 14.318 MHz at 12 bits = 171 Mbit/s

Sampling components (Y / Cb / Cr):

Normally 13.5 MHz luminance frequency

At 4:2:2 sampling totally 27 MHz, 8 bit ->



Available: 5 Mbit/s

Compression ”40:1”

MPEG: History

JPEG – during the 80s

H.120 – 2Mbit/s – DCPM - videoconferencing
H.261, 1989, – 384 kbit/s-1Mbit/s – DCT - videoconferencing
MPEG-1 (ISO/IEC-11172), 1989-1996 – VHS quality at 1,5 Mbit/s
MPEG-2 (ISO/IEC-13818), 1991-2000 – video broadcasting
MPEG-4 (ISO/IEC-14496), 1995-2002 – objects, wavelets
MPEG-7 2002 – “content-based coding”

MPEG-1

•This standard is named after the ISO committee that created it. It has a bandwidth of about 188 KBps, and consists of three layers:

- •MPEG video, encoded at 138 KBps
- •MPEG system (responsible, among other things, for A/V sync), encoded at 34 KBps
- •MPEG audio, the Third Layer, hence MP3, encoded at 128 kilobits per second (16 KBps)

MPEG-2

Improvements for broadcasting

Possibilty to perform parallel decoding

Compression in general

- Lossless

decompressed stream
identical to original

Ex. Run-length encoding
Huffman-encoding
(ZIP, GIF)

- Lossy

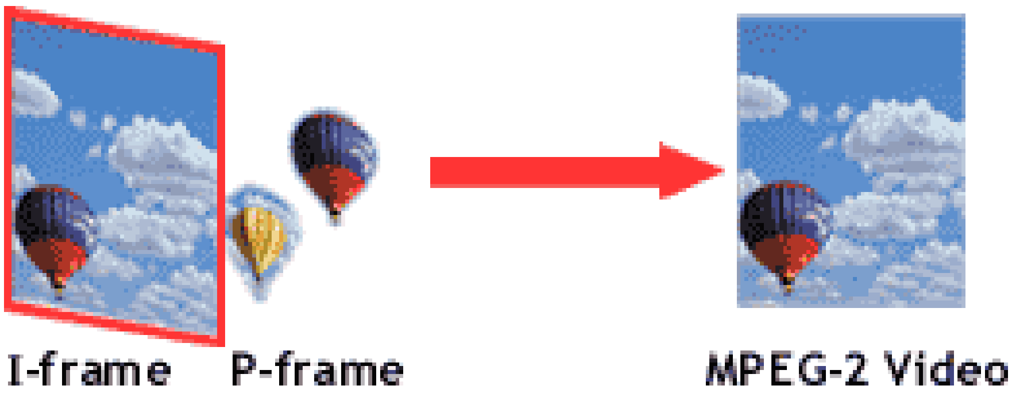
there is loss in data,
taking advt. of limits
of eye and ear

Ex. Interpolation,
subsampling, quantization
(YUV, Wavelets, Fractals)

- Hybrid coding

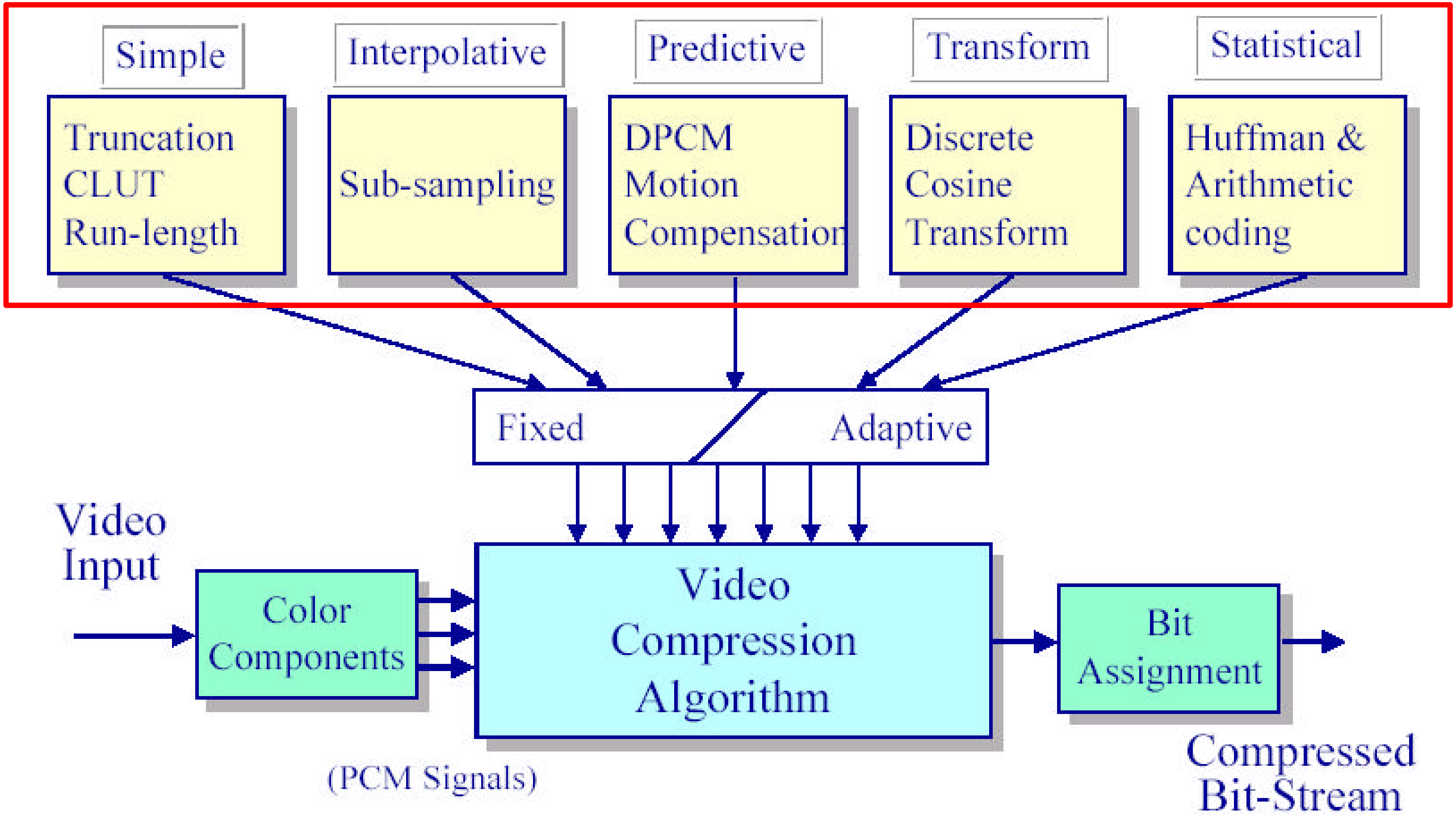
Jpeg, Mpeg,
Last step entropy

MPEG compression techiniques

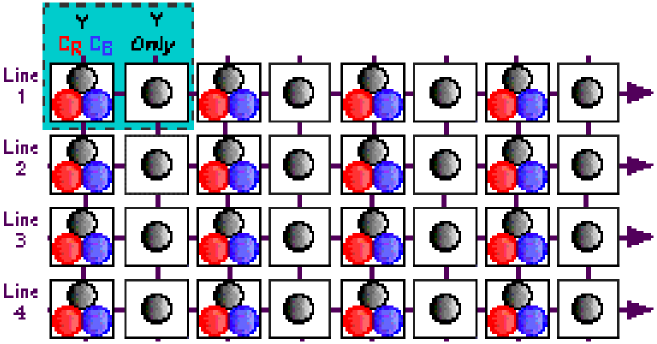


- Temporal, spatial redundancy
- Impercitible details

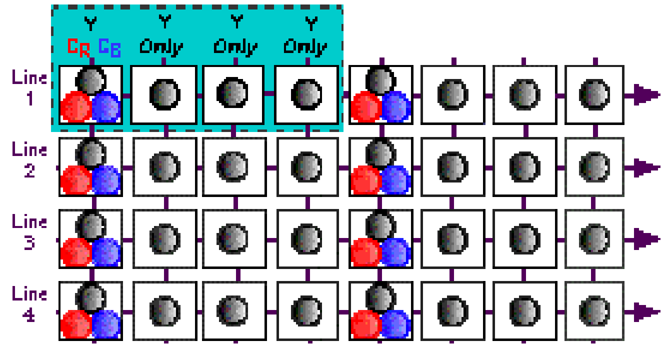
Video compression - methods



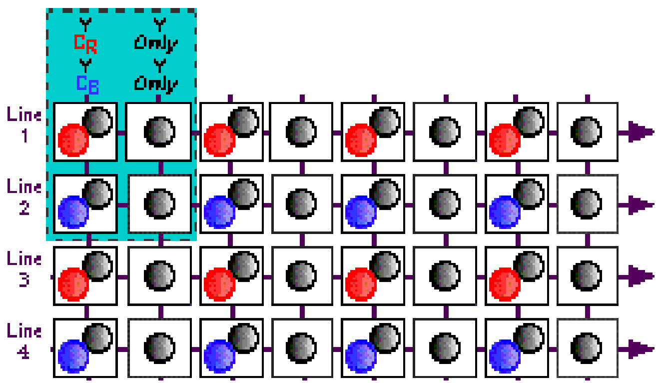
Video Sampling Formats



4:2:2 Sampling (Digital Betacam)
High profile MPEG-2



4:1:1 Sampling (NTSC DV)

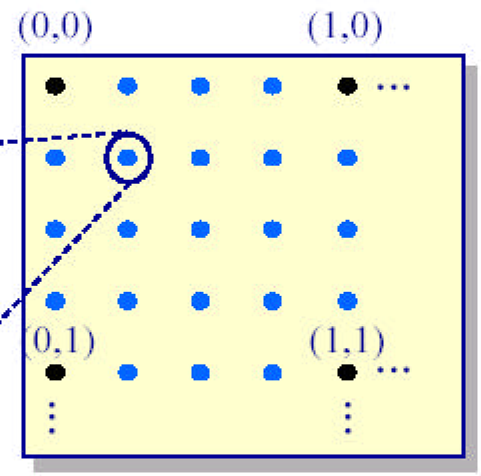


4:2:0 Co-sited (PAL DV)
Normal profile MPEG-2

Sample interpolation
Chrominance

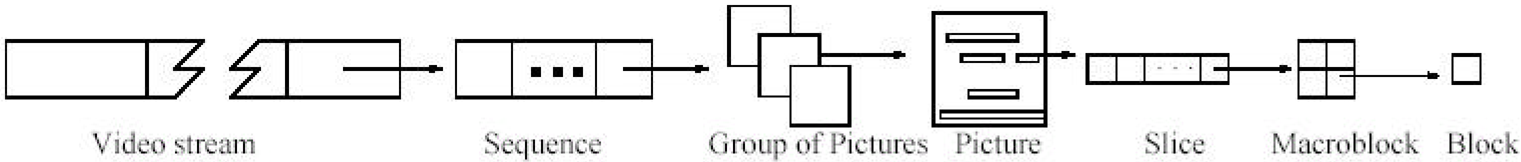
- Bi-linear interpolation:

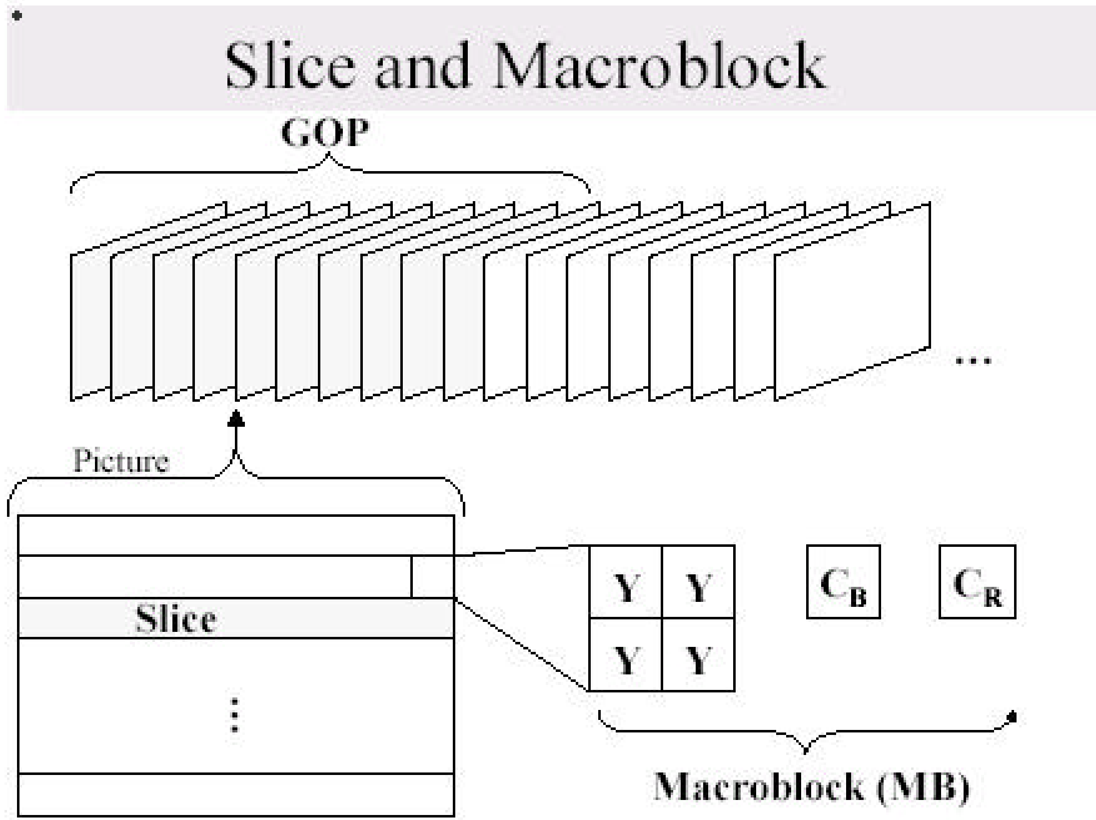
$$U(1,1) = U(0,0) \times 0.75 + U(1,0) \times 0.25 + U(0,1) \times 0.75 + U(1,1) \times 0.25$$



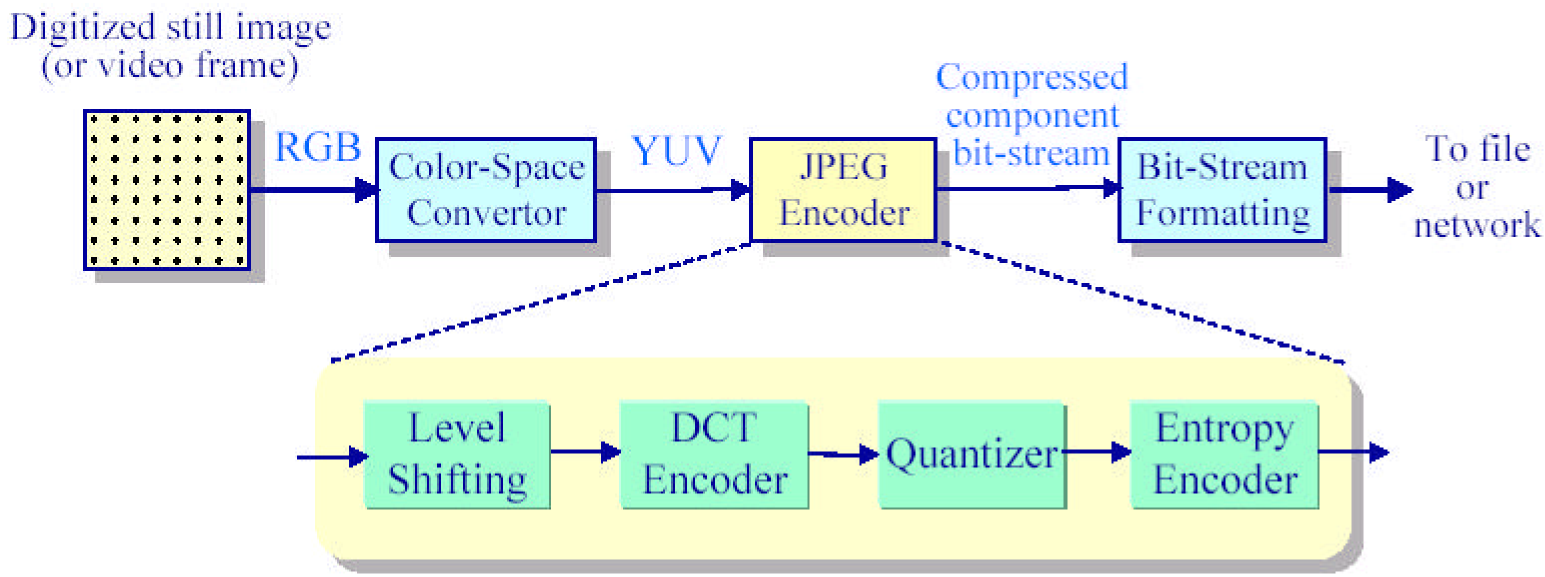
Sub-sampled
U or *V* component

MPEG fundamental blocks

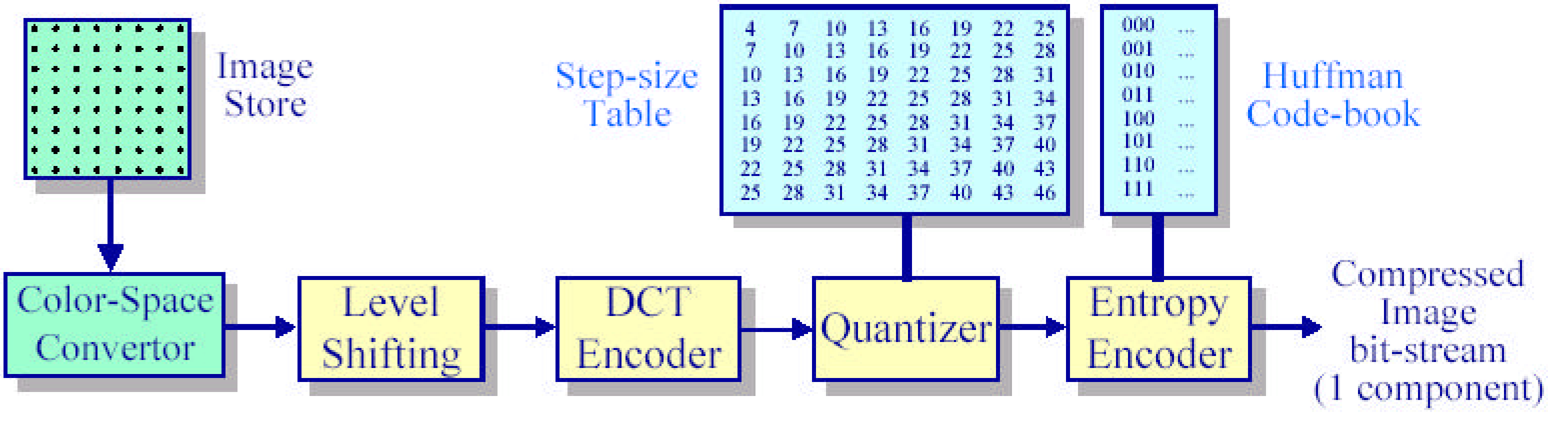




JPEG / Still coding



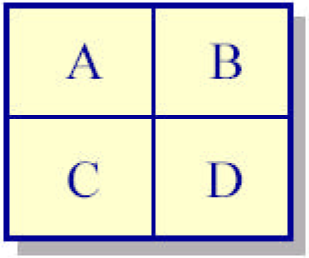
Intra-frame coding: Overall



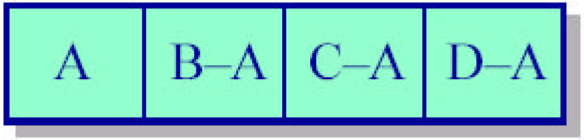
- Compression comes from:
  - Chrominance subsampling
  - DCT coefficient quantization
  - Difference coding DC coefficients
  - Statistical & run-length coding of AC coefficients
- Qualitative results:
  - 0.25 - 0.5 bpp — ok for some applications
  - 0.5 - 0.75 bpp — ok for many
  - 0.75 - 1.5 bpp — excellent
  - 1.5 - 2.0 — indistinguishable

Transformation based compression

2 x 2 array of pixels



1-D array of differences



Encode differences with less precision!!

Common transformations: Fourier, DCT, Wavelets
MPEG2 uses DCT

Discrete cosinus transform

- The 1-dimensional transform:

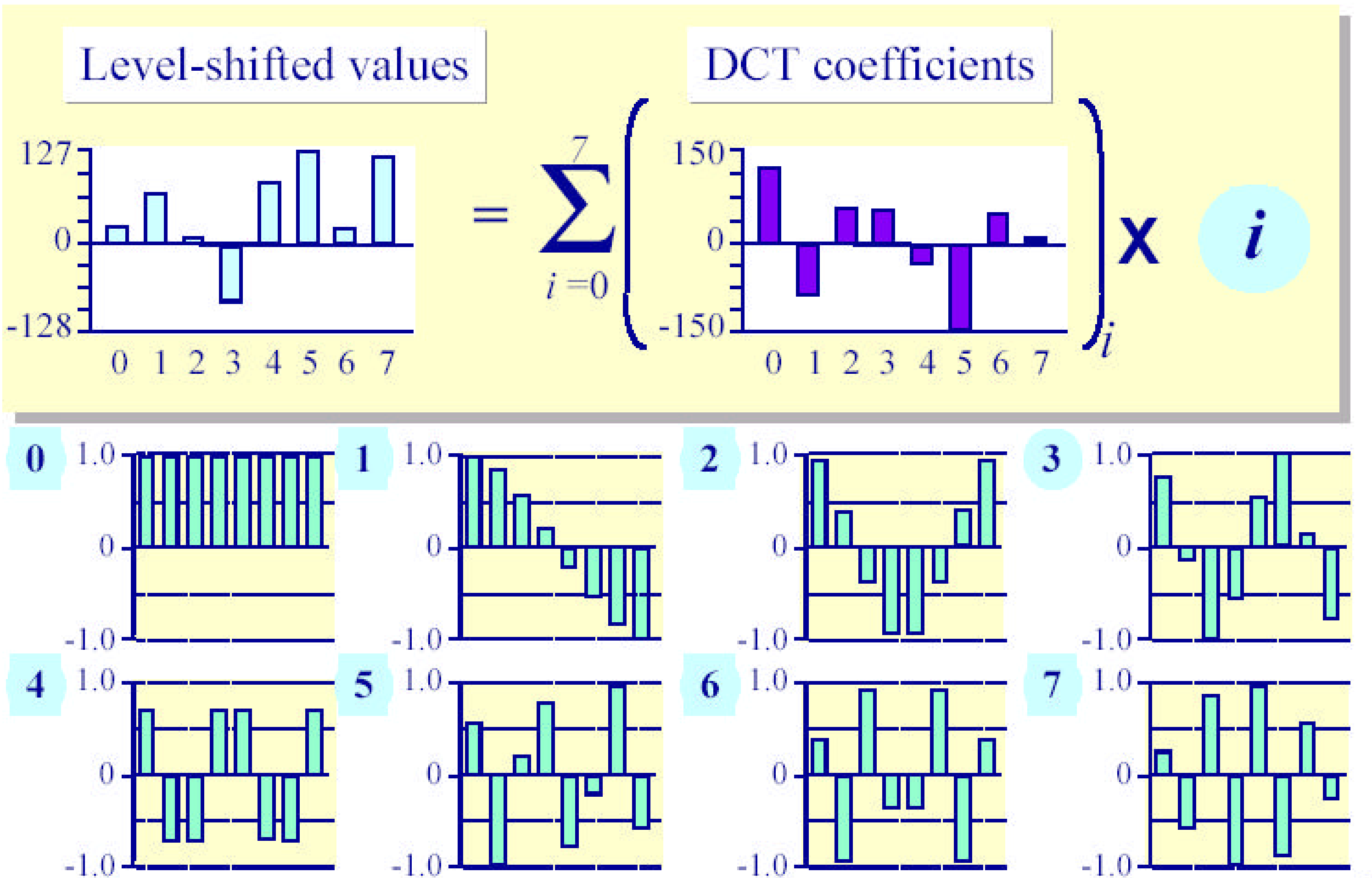
$$F(\mu) = \frac{C(\mu)}{2} \sum_{x=1}^{7} f(x) \cos \frac{(2x+1)\mu\pi}{16}$$

  - » $F(\mu)$ is the DCT coefficient for $\mu = 0..7$
  - » $f(x)$ is the x^{th} input sample fot $x = 0..7$
  - » $C(\mu)$ is a constant (equal to $2^{-0.5}$ if $\mu = 0$ and 1 otherwise)
- The 2-dimensional (spatial) transform:

$$F(\mu,v) = \frac{C(\mu)}{2}\frac{C(v)}{2} \sum_{y=1}^{7} \sum_{x=1}^{7} f(x,y) \cos \frac{(2x+1)\mu\pi}{16} \cos \frac{(2y+1)v\pi}{16}$$

Discrete cosinus transform

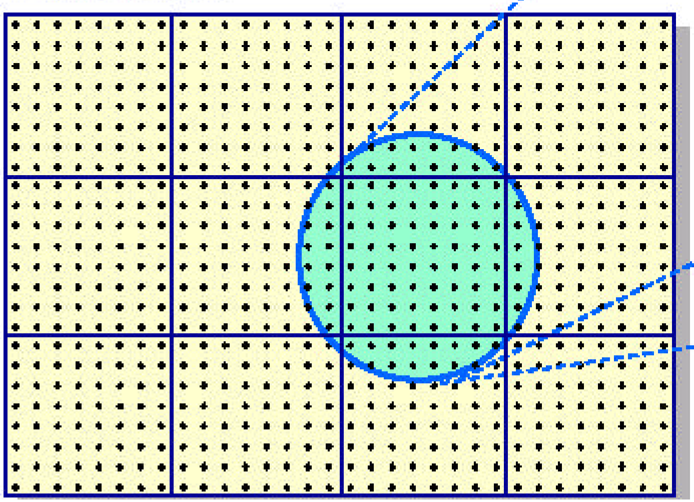
Represents input as sum of basic functions



DCT on frames

- Apply the DCT in *x* and *y* dimensions simultaneously to 8x8 pixel blocks
  - » Code coefficients individually with fewer bits

Video Frame

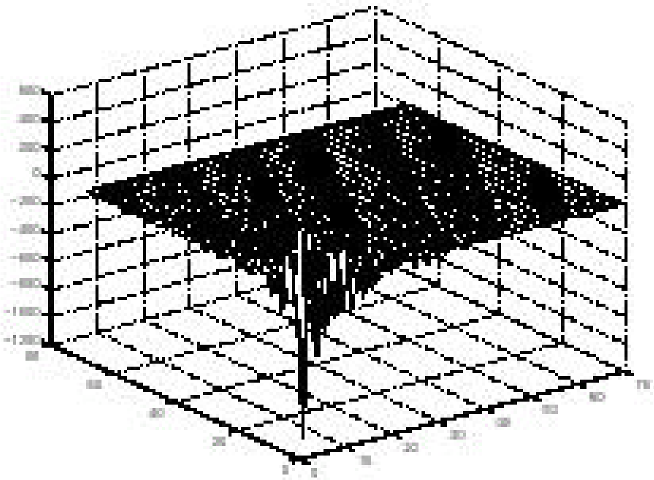


172	-18	15	-8	23	-9	-14	19
21	-34	24	-8	-10	11	14	7
-9	-8	-4	6	-5	4	3	-1
-10	6	-5	4	-4	4	2	1
-8	-2	-3	5	-3	3	4	6
4	-2	-4	6	-4	4	2	-1
4	-3	-4	5	6	3	1	1
0	-8	-4	3	2	1	4	0

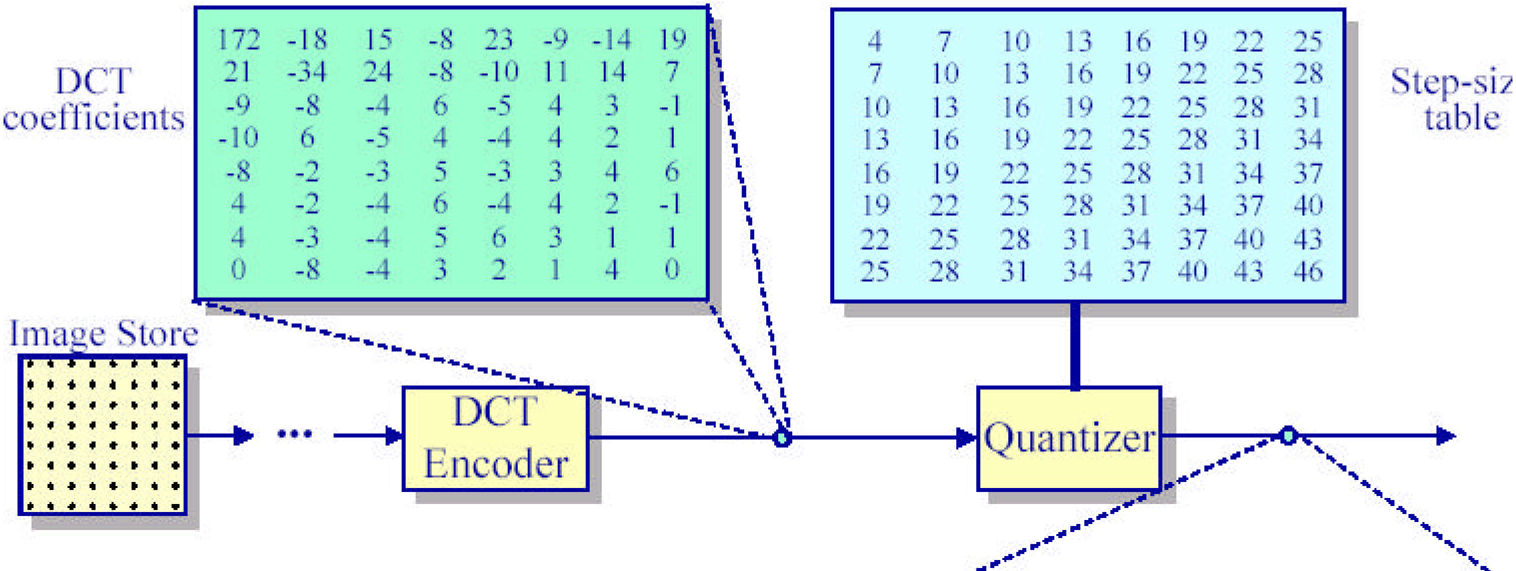
DCT Coefficients

79

DCT on frames: Example



Quantization



DCT coefficients:

172	-18	15	-8	23	-9	-14	19
21	-34	24	-8	-10	11	14	7
-9	-8	-4	6	-5	4	3	-1
-10	6	-5	4	-4	4	2	1
-8	-2	-3	5	-3	3	4	6
4	-2	-4	6	-4	4	2	-1
4	-3	-4	5	6	3	1	1
0	-8	-4	3	2	1	4	0

Step-siz table:

4	7	10	13	16	19	22	25
7	10	13	16	19	22	25	28
10	13	16	19	22	25	28	31
13	16	19	22	25	28	31	34
16	19	22	25	28	31	34	37
19	22	25	28	31	34	37	40
22	25	28	31	34	37	40	43
25	28	31	34	37	40	43	46

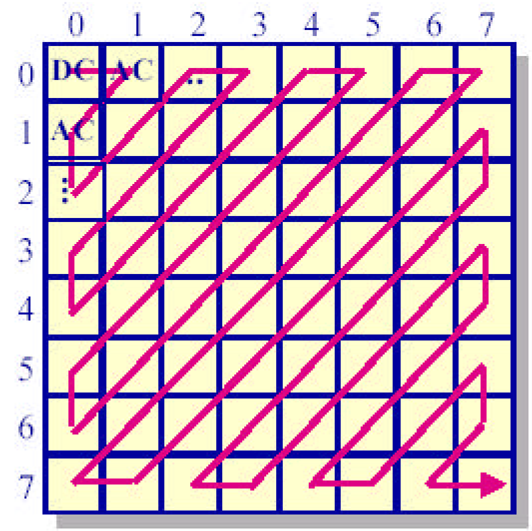
43	3	2	1	1	0	0	0
3	3	2	1	0	0	0	0
1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

- Step-size table can be scaled to control degree of compression
  - » Scaling factor called the *"q-factor"*

* Variable in MPEG stream

Zig-zag walking

In order to increas effectiveness of run-length encoding



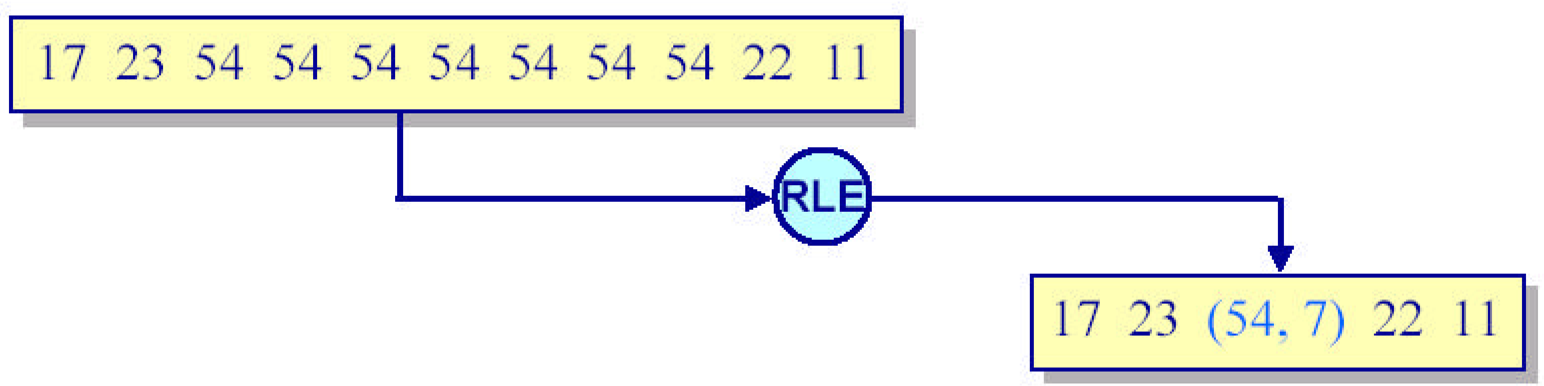
43	3	2	1	1	0	0	0
3	3	2	1	0	0	0	0
1	1	0	0	0	0	0	0
1	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

43,3,3,1,3,2,1,2,1,1,1,0,0,1,1,0,0,....

After RLE:

43, (3,2), 1, 3, 2, 1, 2, (1,3), (0,2), (1,2),(0,49

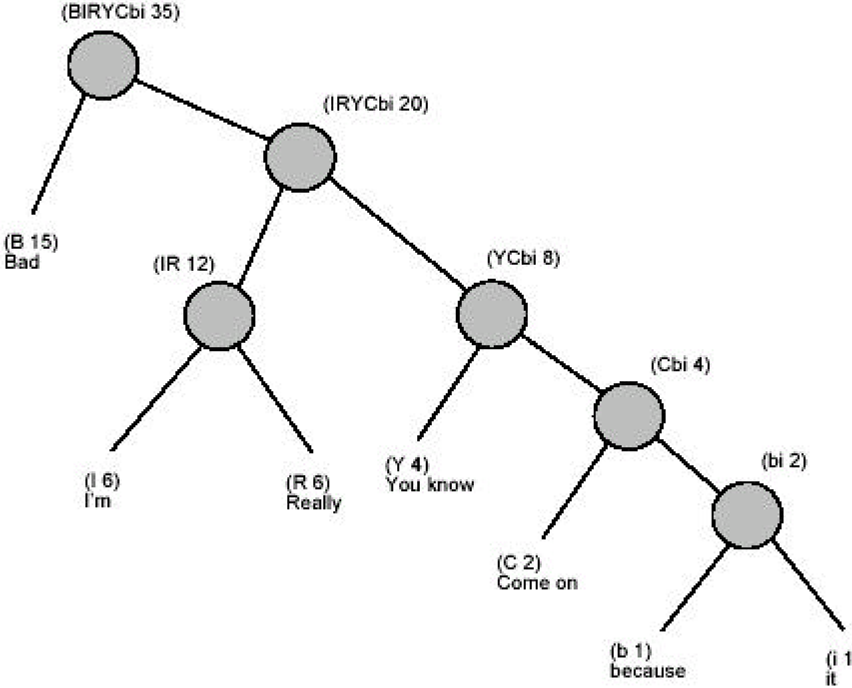
Run-length encoding



Huffman encoder - principles

```
Because I'm bad, I'm bad-- come on
Bad, bad-- really, really bad
You know I'm bad, I'm bad--
you know it
Bad, bad-- really, really bad
You know I'm bad, I'm bad-- come on, you kn
Bad, bad-- really, really bad
```

Phrase	Symbol	Frequency	Code Length	Code
Because	b	1	5	00001
I'm	I	6	3	011
Bad	B	15	1	1
Come on	C	2	4	0001
It	i	1	5	00000
Really	R	6	3	010
You know	Y	4	3	001



Transformation based compression: Wavelets

Orig:64x64
8bpp

2.88 bpp

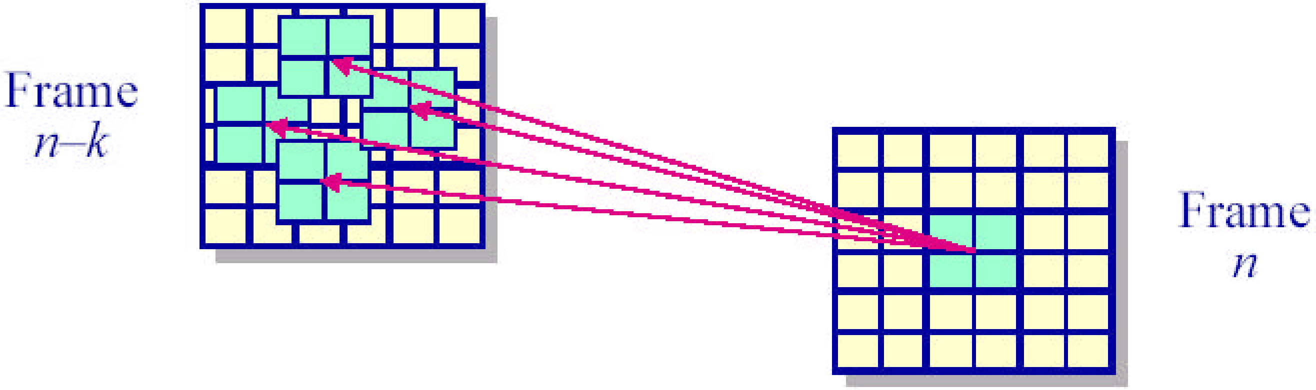
1.68 bpp

0.88 bpp

0.4 bpp

0.16 bpp

MPEG motion compensation

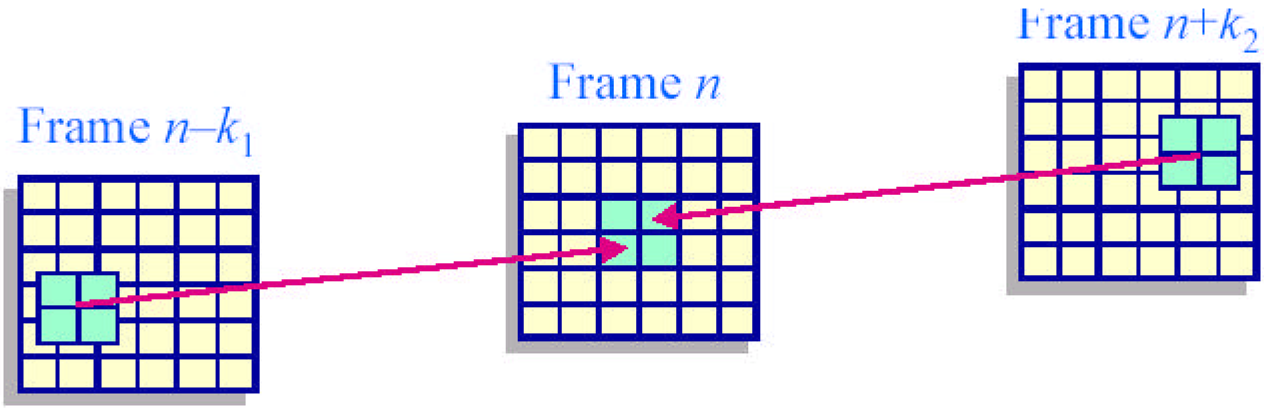


- Performed on macroblock (16x16 pixels)
- The prediction search space is not specified in the standard
- Objective: Find an motion vector *w* that minimizes some cost function, e.g.

$$\sum_{j=0}^{15}\sum_{i=0}^{15} f(frame_n[16b_x+i,\ 16b_y+j]-frame_{n-1}[16b_x+w_x+i,\ 16b_x+w_y+j])$$

MPEG motion compensation

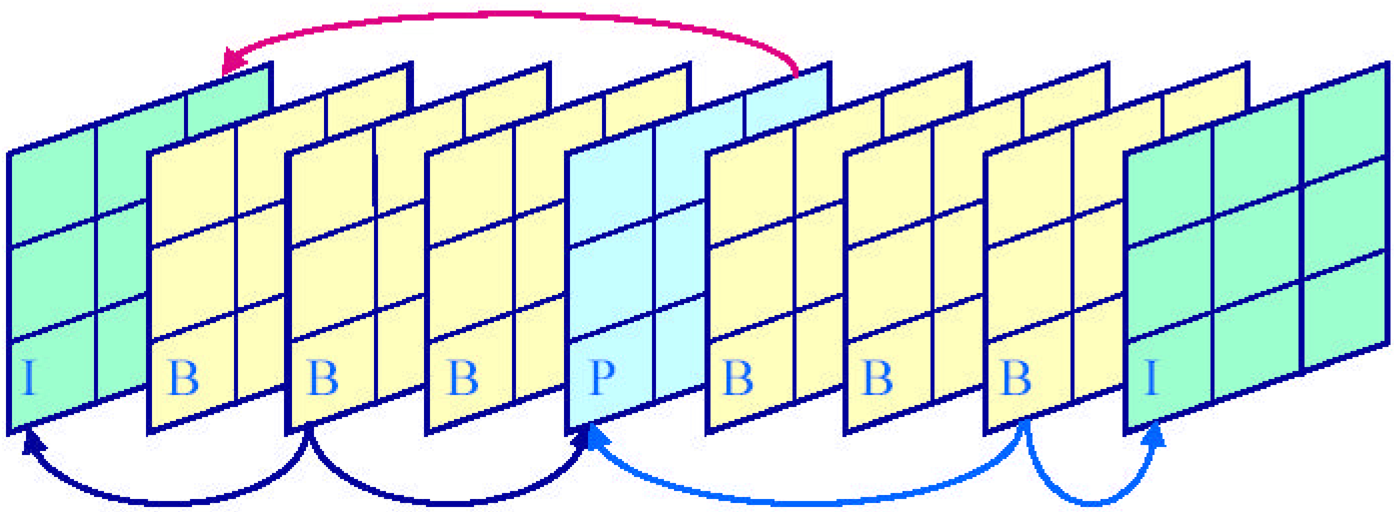
Bi-directional prediction



- Not only prediction, also interpolation
- Both past and future frames are used to predict the current frame

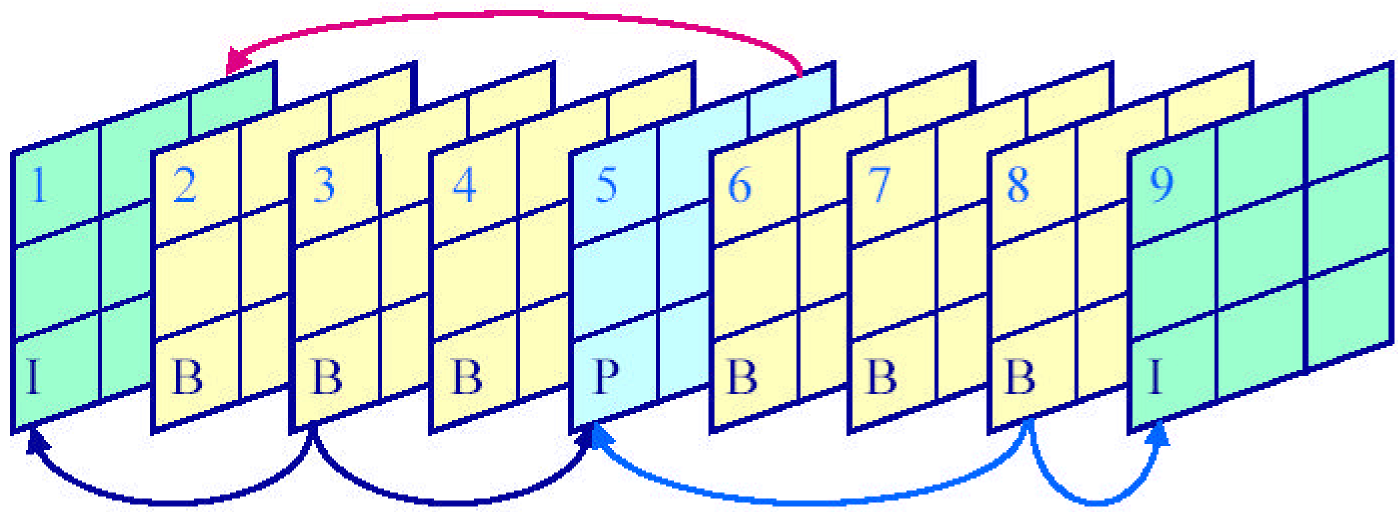
MPEG motion compensation

Bi-directional prediction



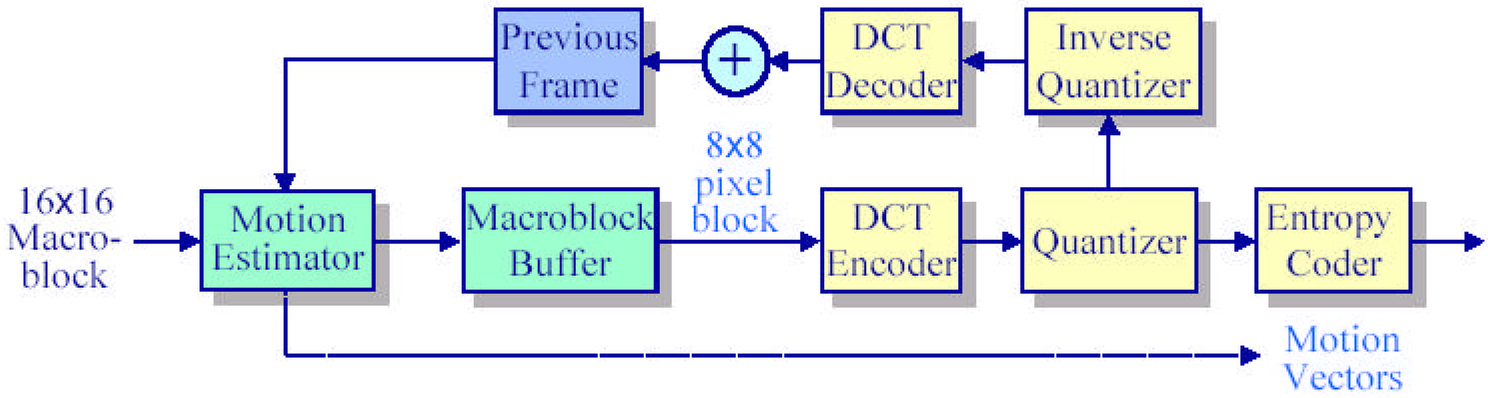
- 3 types of pictures in a MPEG stream
  - I – Intracoded pictures (still images)
  - P – Predicted pictures from previous I or P pictures
  - B – Bi-directinally pictures (interpolated pictures)

MPEG motion compensation
Bi-directional prediction

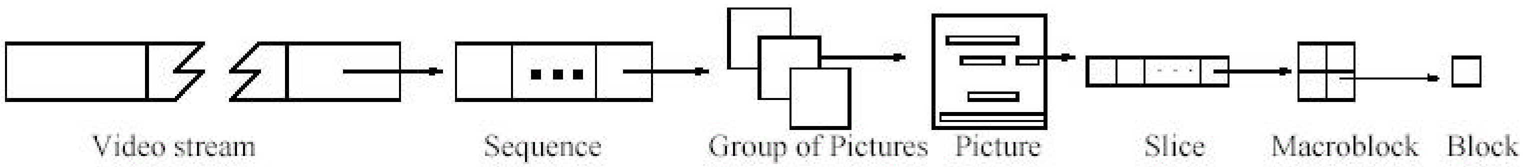


- Directional prediction implies that frames cannot be encoded or transmitted in the order they are scanned
- Encoding & transmission order:
  - » I_1 P_5 B_2 B_3 B_4 I_9 B_6 B_7 B_8 P_{13} B_{10} B_{11} B_{12} I_{17} B_{14} B_{15} B_{16} ...

Structure of MPEG encoding

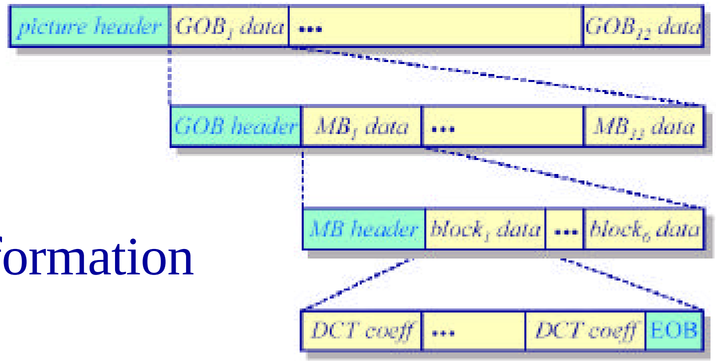


MPEG fundamental blocks



MPEG-2 Stream

» Sequence Layer
 * decoding parameters (bit-rate, buffer size, picture resolution, framerate, ...)



» Group of Pictures Layer
 * a random access point

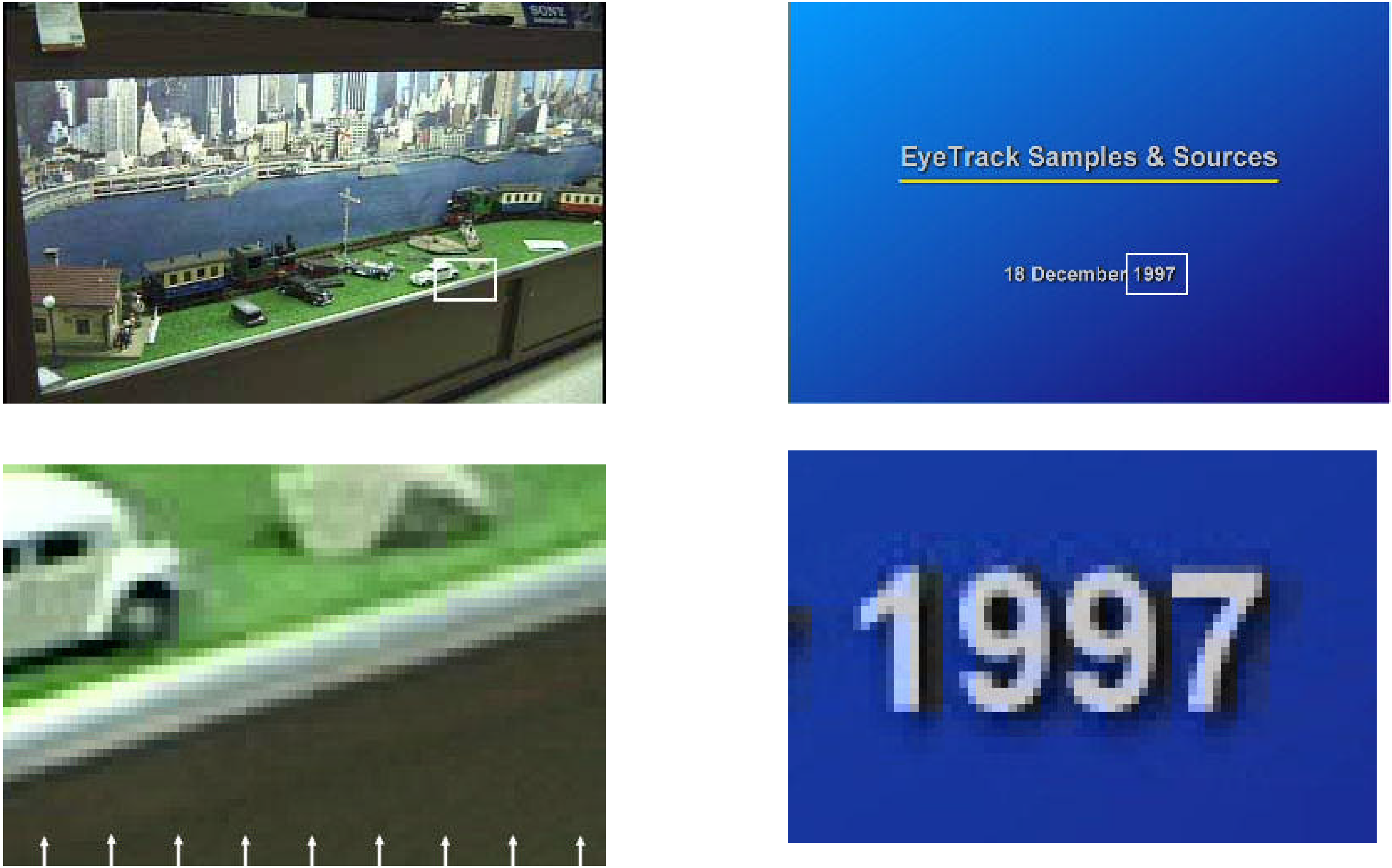
» Picture Layer
 * picture type and reference picture information

» Slice Layer
 * position and state information for decoder resynchronization

» Macroblock Layer
 * coded motion vectors

» Block Layer
 * coded DCT coefficients, quantizer step size, *etc*.

Compression artifacts



MPEG standardization

What about research???

• pre-processing: (or "how to fit a square peg in a round hole?").

• motion estimation (or "how to efficiently find a good prediction.")

• macroblock decision models (efficient, but does it also optimise the ripple effect on subsequent macroblocks ?)

• rate control and buffer management in editing environments (MPEG: video only exists within a sequence. Real world: decoder is displaying picture from previous sequence, while reconstructing a picture from the new sequence)

• implementation complexity reduction ("let's run it on a Pentium!").