Part II – Video

- General Concepts
- MPEG1 encoding
- MPEG2 encoding
- H.264 encoding
- MPEG4 encoding

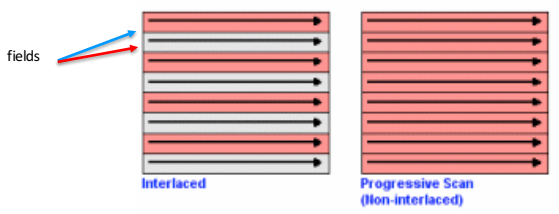
Video General Concepts

Digital video

- Digital video is a sequence of frames produced by a digital camera, consecutively transmitted and displayed so to provide a continuum of actions. This is obtained by adjusting the frequency of frames to the properties of the visual human system
- There are advantages with digital video:
  - Digital video can be copied with no degradation in quality. No matter how many generations of a digital source is copied
  - Digital video can be manipulated and edited on a computer-based device. More and more, consumer-grade computer hardware and software are available.
  - Recording digital video is very inexpensive. Digital video increased in quality with the introduction of MPEG-1 and MPEG-2 standards (adopted for use in television transmission and DVD media)
- Digital television (including higher quality HDTV) started to spread in most developed countries in early 2000s.
- Digital video is increasingly diffused in film industry. Paramount has been the first to produce only digital, from 2013
- Digital video is also used in modern mobile phones. Digital video is used for Internet distribution of media, including streaming video and peer-to-peer movie distribution.

Digital video cameras

- Digital video cameras come in two different image capture formats: interlaced and progressive scan
  - ***Interlaced cameras*** record the image in alternating sets of lines: the odd-numbered lines first, and then the even-numbered lines. One set of odd or even lines is referred to as a *field*, and a consecutive pairing of two fields of opposite parity is called a *frame*.
  - ***Progressive scan cameras*** record each frame as distinct, with all scan lines being captured at the same moment in time.

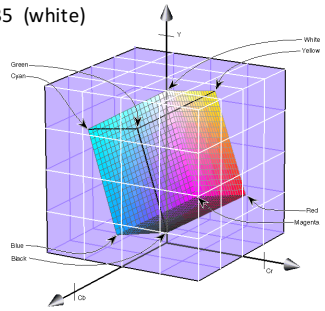


Digital video color spaces

- Video color is displayed in RGB (monitors use RGB). Although RGB color components could be used to represent color information in video however these signals are expensive to record, process and transmit.
- Digital video is therefore transmitted and stored using YCbCr. or Y'CbCr *color spaces* that distinguish instead *brightness* and *chrominance* information

- With YCbCr and Y'CbCr the values are scaled and offsets are added:
  for Y (Y') component: from 16 (black) to 235 (white)
  for Cb Cr components: from 16 to 240

$$
\begin{aligned}
Y' &= 16 + \frac{65.738 \cdot R'_D}{256} + \frac{129.057 \cdot G'_D}{256} + \frac{25.064 \cdot B'_D}{256} \\
C_B &= 128 - \frac{37.945 \cdot R'_D}{256} - \frac{74.494 \cdot G'_D}{256} + \frac{112.439 \cdot B'_D}{256} \\
C_R &= 128 + \frac{112.439 \cdot R'_D}{256} - \frac{94.154 \cdot G'_D}{256} - \frac{18.285 \cdot B'_D}{256}
\end{aligned}
$$



Digital video encoding

- Brightness and chrominance of images can be carried either combined in one channel as in ***composite encoding*** (brightness and chrominance information are mixed together in a single signal) or in separate channels as ***component encoding.***
- In Digital video component color encoding is used (brightness and chrominance of images are carried in separate channels)

Digital video bitrate

.

- For digital video we use the term ***bitrate,*** counting the number of bits that are conveyed or processed per unit of time (measured in bits per second) :
  - 16 Kbit/s videophone quality (talking heads)
  - 128-364 Kbit/s videoconferencing quality with video compression
  - 1.25 Mbit/s video CD quality with MPEG1 compression
  - 5 Mbit/s DVD quality with MPEG2 compression
  - 8-16 Mbit/s HDTV quality with MPEG4 compression
  - 29.4 Mbit/s HD DVD quality

- A theoretical upper bound for the bitrate in bits/s for a certain spectral bandwidth in Hertz is given by the *Nyquist law* for low-pass and bandpass cases:

Low-pass:	Bitrate ≤ 2 · bandwidth (Nyquist rate)
Band-pass:	Bitrate ≤ Bandwidth

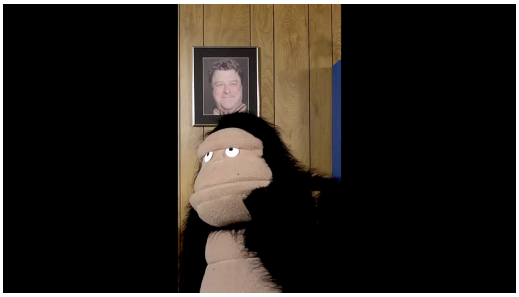
Digital video aspect ratio

- The *aspect ratio* of an image describes the proportional relationship between its height and width.

- Current standards for digital video *aspect ratio* are:
  - **4:3** (1.33:1) for standard television has been in use since the invention of moving picture cameras and many computer monitors used to employ the same aspect ratio.
  - **16:9** the international standard format of HDTV, non-HD digital television. Many digital video cameras have the capability to record in 16:9.
  - **Square video and vertical video** new video formats more suited to mobile devices.

Square video was popularized by Instagram and then supported by Facebook and Twitter.

Vertical video (9:16 format) was popularized by Snapchat and is also now being adopted by Twitter and Facebook.



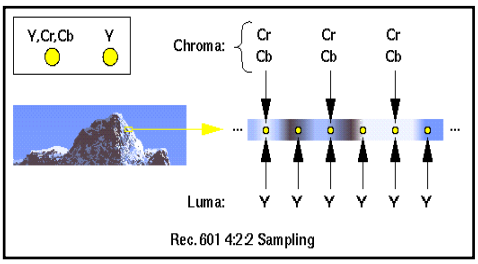
Video sampling

- Sampling is a mechanism for data compression in video. It applies to luminance and chroma information in each video frame. Because the human visual system is less sensitive to the position and motion of color than luminance, bandwidth can be optimized by storing more luminance detail than color detail.

- Sampling is expressed with three values: x,y,z
  - x = relative number of luminance (Y) samples (sampling reference usually 4)
  - y = number of chroma (CbCr) samples for odd lines (in the first row of x pixels)
  - z = number of chroma (CbCr) samples for even lines (in the second row of x pixels)

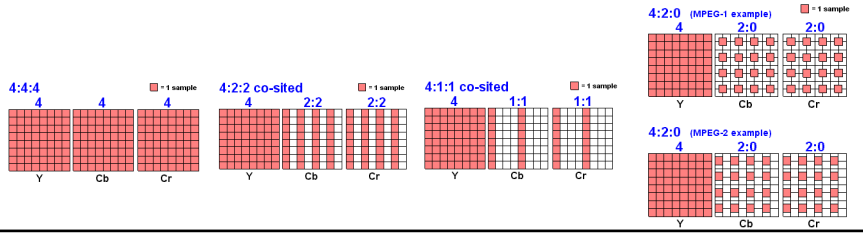
  Es. 4:2:2 means that every 4 samples of luminance, there are 2 chroma samples both in the odd and the even lines. It compresses frames as it drops data. 4:2:0 provides higher compression

For each line



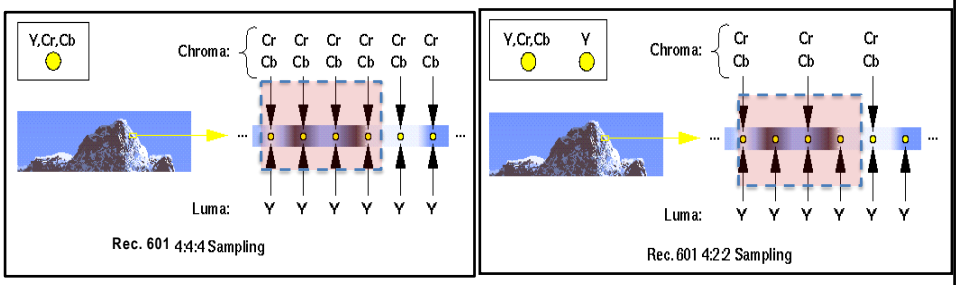
Digital video sampling rates

- Digital video can be stored and transmitted with different sampling rates:
  - **4:4:4 (Cb/Cr same as Luminance)** Cb and Cr are sampled at the same full rate as the Y. MPEG-2 supports 4:4:4 coding. When video is converted from one color space to another, it is often resampled to 4:4:4 first.
  - **4:2:2 (Cb/Cr 1/2 the Luminance samples)** Cb and Cr are sampled at half the horizontal resolution of Y and taken at the same time as Y. It is considered very high quality and used for professional digital video recording. It is an option in MPEG-2.
  - **4:1:1 (Cb/Cr 1/4 the Luminance samples)** Cb and Cr are sampled at one quarter the horizontal resolution of Y and taken at the same time as Y.
  - **4:2:0 (Cb/Cr 1/4 the Luminance samples)** The zero in 4:2:0 means that Cb and Cr are sampled at half the horizontal and vertical resolution of Y. MPEG-1 and MPEG-2 use 4:2:0, but the samples are taken at different intervals. H.261/263 also uses 4:2:0.



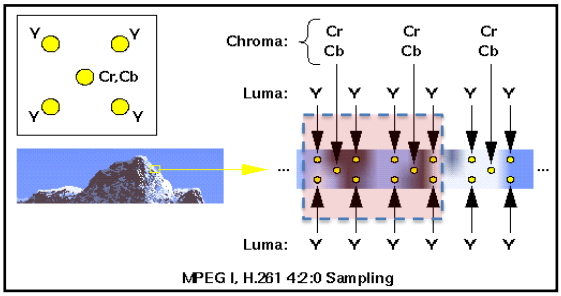
Digital video format ITU-R BT.601

- Standard ITU-R BT.601 for digital video (also referred as CCIR Recommendation 601 or Rec. 601) defines, independently from the way in which the signal is transmitted, the color space to use, the pixel sampling frequency. Distinct modes of color sampling are defined:
  - 4:4:4 a pair of Cr Cb every Y
  - 4:2:2 a pair of Cr Cb every two Y
  - 4:2:0 a pair of Cr Cb every two Y in alternate lines



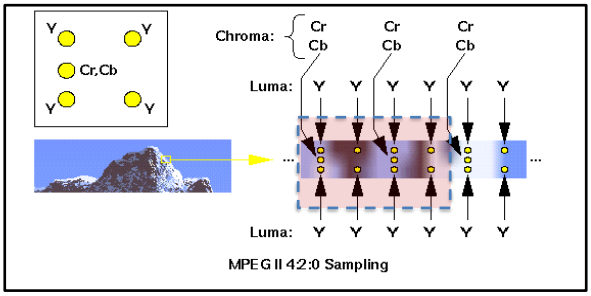
Digital video format MPEG 1

- In MPEG1:
  - 4:2:0 sampling
  - Bitrate: ~ 1.5 Mbit/s, non interlaced
  - Frame size: 352x240 or 352x288



Digital video format MPEG 2

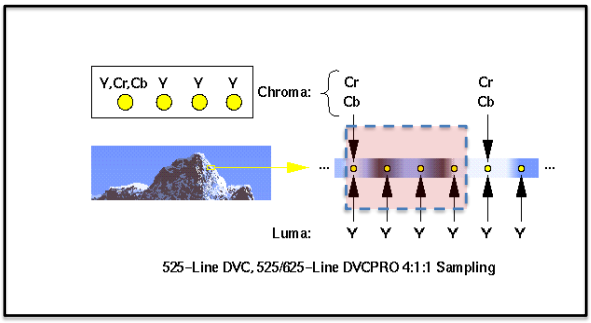
- MPEG2 was defined to provide a better resolution than MPEG1 and manage interlaced data. Based on fields instead of frames. Used for DVD and HDTV:
  - 4:2:0 sampling
  - Bitrate 5 Mbit/s.
  - Frame size: 720x480



MPEG II 4:2:0 Sampling

Digital video format DV

- DV standard is used for registration and transmission of digital video over cables. It employs *digital video component* format to separate luminance and chrominance.
  - Color sampling (typical): 4:1:1 (NTSC, PAL DVC PRO)
  - Horizontal resolution for luminance is 550
  - Horizontal resolution for chroma is about 150 lines (about ¼)
- Many standards: DV25, DV50, DV100



525-Line DVC, 525/625-Line DVCPRO 4:1:1 Sampling

Other digital video formats

- Other formats for (professional) digital video are:
  - D1 (CCIR 601, 8bit, uncompressed)
  - D2 (manages 8 bit color)
  - D3 (used by BBC…)
  - D5 (10bit, uncompressed) / D5 HD
  - D9
  - Digital BetaCam (HDCAM / HDCAM SR for HD format, with 4:2:2 and 4:4:4 RGB)
  - …

Video compression

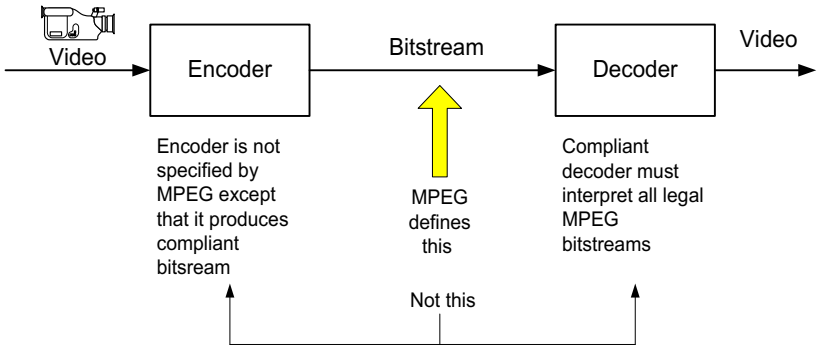
- Video compression algorithms attempt to reduce the amount of information that is contained in video while preserving quality. They can be lossy and lossles but typically are lossy, starting with color subsampling
- Algorithms can be symmetric or not symmetric, in terms of (de)compression time/complexity. Typically video compression algorithms for video distribution are highly asymmetric
- Compression can be spatial or/and temporal
  - remove spatially redundant data (as in JPEG)
  - remove temporally redundant data (the basis for good video compression)

Motivation for compression

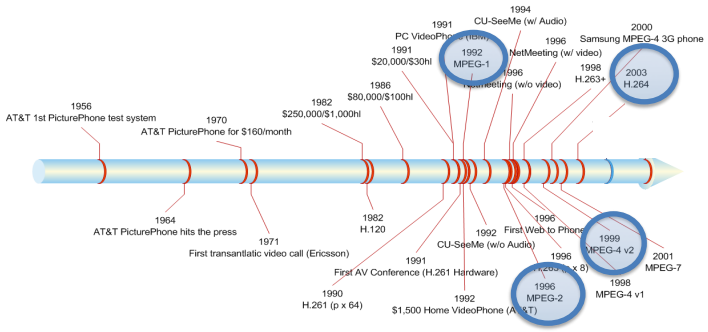
- An example: suppose we have a video with a duration of 1 hour (3600sec), a frame size of 640x480 (WxH) pixels at a color depth of 24bits (8bits x 3 channels) and a frame rate of 25fps.
- This video has the following size:
  - pixels per frame = 640 * 480 = 307,200
  - bits per frame = 307,200 * 24 = 7,372,800 = 7.37Mbits
  - bit rate = 7.37 * 25 = 184.25Mbits/sec
  - video size = 184Mbits/sec * 3600sec = 662,400Mbits = 82,800Mbytes = 82.8Gbytes
- Compressing video aims at reducing the average bits per pixel (bpp):
  - with chroma subsampling we reduce from 24 to 12-16 bpp
  - with JPEG compression we reduce to 1-8 bpp
  - with MPEG we go below 1 bpp

MPEG compression

- MPEG compression methods have been defined defined according to ISO standard
  - The MPEG defines the protocol of the bitstream between the encoder and the decoder
  - The decoder is defined by implication. The encoder is left to the designer



Progress of MPEG standard (1990- onwards)

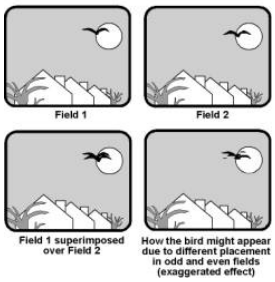


- MPEG-1: "Coding of moving pictures and associated audio for digital storage media"
  - VHS Quality at 1.5 MBits/s
  - Basis of Video-CD
  - MP3 (MPEG-1 Layer 3)
- MPEG-2: "Generic coding of Moving Pictures and Associated Audio"
  - Broadcasting and storage
  - Bitrates: 4-9 MBits/s
  - Satellite TV, DVD
- MPEG-3?
  Aimed to do High Definition TV (HDTV)
  Folded into MPEG-2
- MPEG-4: "Coding of audio-visual objects"
  Started as very low-bitrate project
  Turned out to be much more:
  - Coding of media *objects*
  - 64kbps to 240Mbps (Part 10/H.264)
  - Synthetic/Semi-synthetic objects
  - Intellectual Property Management

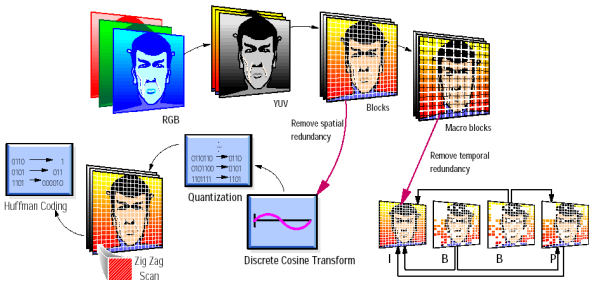
Part II - MPEG 1

MPEG1

- MPEG1 is an ISO standard (ISO/IEC 11172) developed to support VHS quality video at bitrate of ~1.5 Mbps. MPEG1 defines the syntax of encoding a stream video and the method for decoding. However the encoder can be implemented in different ways.
- MPEG1 was developed for *progressive video* (non interlaced) so it manages only frames (progressive scan): input is given according to SIF Standard Image Format and is made of 1 field
- If we have interlaced video, two fields can be combined into a single frame, and hence encoded with MPEG1. However in this case there are artifacts due to the motion of the objects. MPEG2 is a better choice in this case, since it manages fields natively.



- MPEG1 is based on the principle that an encoding of the differences between adjacent still pictures is a fruitful approach to compression. It assumes that:
  - A moving picture is simply a succession of still pictures.
  - The differences between adjacent still pictures are generally small
- Main features of MPEG1
  - *intra-frame coding*: transform-domain-based compression (based on blocks, similar to JPEG with 2D DCT, quantization and run-length encoding)
  - *Inter-frame coding*: block-based motion compensation (based on macroblocks, considers a group of blocks of pixels common to two or more successive frames and replaces it by a pointer i.e. a *motion vector* that references the macroblock)



CPB Constrained Parameters Bitstream

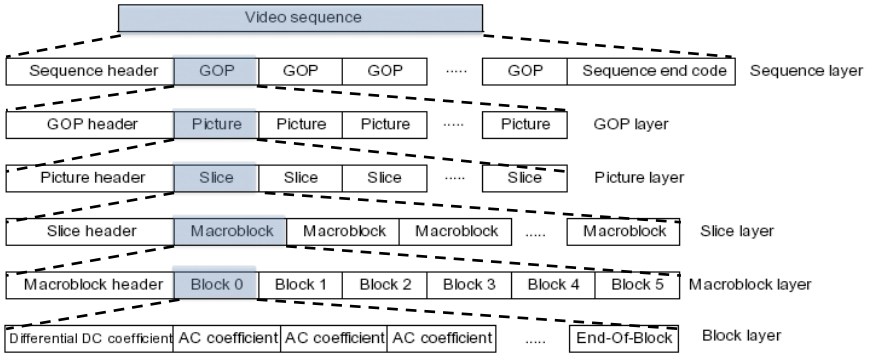
- The usual MPEG1 video resolution is: 352x240 or 320x240 at a bitrate of ~1.5 Mbps. This modality is referred to as *Constrained Parameters Bitstream* or CPB (1 bit of the stream indicates if CPB is used) and is the minimum video specification for a decoder to be MPEG compliant

Resolution	Frames per Second
352×240	29.97
352×240	23.976
352×288	25
320×240[1]	29.97
384×288[1]	25

- MPEG1 can also provide compressed video at broadcast quality with a bandwidth up to 4 Mbps - 6 Mbps. Similar quality is obtained in MPEG-2 with 4 Mbps bandwidth, thanks to fields.

6 layers

- Sequence:
  - unit for random access
- GOP:
  - unit for video random access (the smallest unit of independent coding)
- Picture (frame):
  - primary coding unit
- Slice:
  - syncronizzation unit
- Macroblock:
  - motion compensation unit
- Block:
  - unit for DCT processing

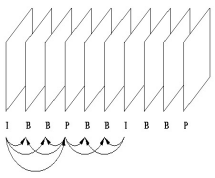


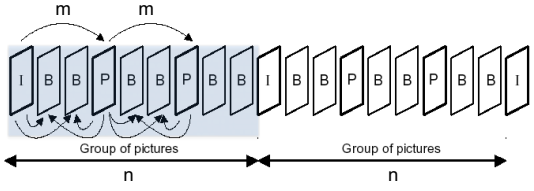
GOP

- A video sequence is decomposed in Groups of Pictures (GOPs). Frames have different typology: I (intra-coded), P (Predictive), B (Bi-directional), D (DC) frame. Frame types:
  - I, P, B occur in repetitive patterns within a GOP; there are predictive relationships between I, P and B frames. Relative number of I, P and B pictures can be arbitrary. It depends on the nature of the application
  - D frames contain DC coefficients only. They are low quality representations used as thumbnails in video summaries

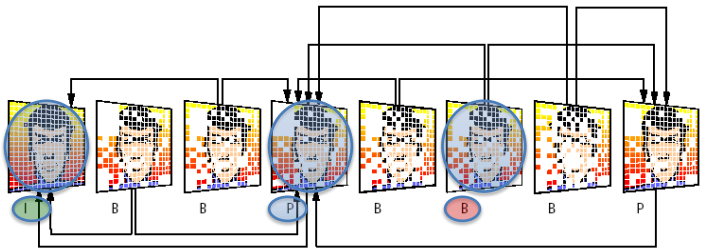
- Distance between I, P e B frames can be defined when coding. The smaller GOP is the better is fidelity to motion and the smaller compression (due to I frames)
- A GOP is *closed* if can be decoded without information from frames of the preceding GOP (ends with I,P or B with past prediction). Max GOP lenght are 14-17

Typically m=3, n=9





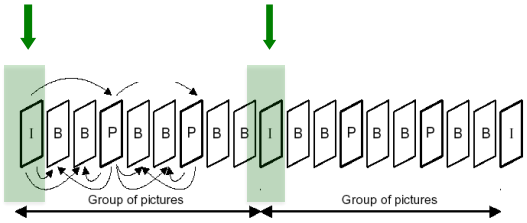
Frames



- I-frame: contains the full image
- P-frame is based on preceding I o P-frame
- B-frame uses past or future I o P frames

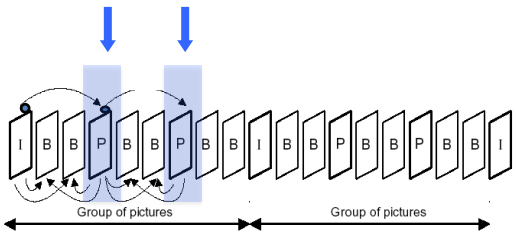
I-frames

- *Intra – coded frames* are so called because they are decoded independently from any other frames. They are identical to JPEG frames.
- Intra-Coded frame are coded with no reference to other frames. They act as *anchor* for P or B frames. Minimize propagation of errors and permit random access. Relatively smaller amount of compression is expected to be achieved at I pictures compared to P and B pictures.
- I-frame compression is very fast but produces large files (three times larger than normally encoded MPEG video).



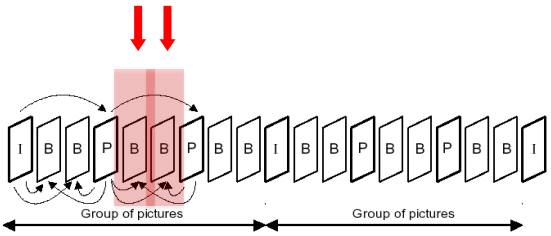
P-frames

- *Predictive-Coded frames* are coded with *forward motion prediction* from preceding I o P frame.
- Improve compression by exploiting the temporal redundancy. They store the difference in image from the frame immediately preceding it. The difference is calculated using *motion vectors*.



B-frames

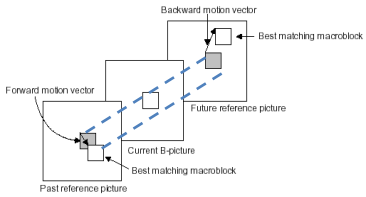
- *Bi-directional-Coded frames* are coded with *bidirectional (past and future) motion compensation* using I and P frame (no B frame).
- Motion is inferred by averaging past and future predictions. B pictures are expected to provide relatively the largest amount of compression under favorable predict
- Harder to encode introduces delay in coding: the player must first decode the next I or P frame sequentially after the B frame before it can be decoded and displayed. This makes B frames computationally complex and requires large data buffers.

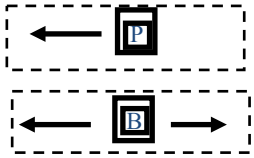


Macroblocks

- Each video *frame* contains *macroblocks* that are the processing unit of MPEG1 compression. Macroblocks are set of 16x16 pixel and are necessary for purposes of the calculation of motion vectors and error blocks for motion compensation.

- Main types of macroblocks:
  - I *macroblocks*: encoded independently of other macroblocks (by 2D Discrete Cosine Transform as in JPEG blocks)
  - P *macroblocks*: encode not the region but the motion vector and error block of the previous frame (forward predicted macroblock)
  - B *macroblocks*: same as above except that the motion vector and error block are encoded from the previous (forward predicted macroblock) or next frame (backward predicted macroblock)





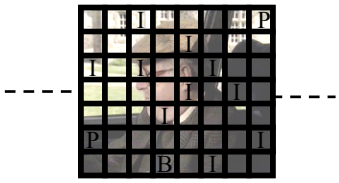
Frames and macroblocks

- In MPEG1 compression is performed at the macroblock level. Therefore frames can contain different types of macroblocks

- P and B frames contain encoding of residual error after prediction:
  - P frames: contain *Intra-coded* (I) *macroblocks* or *forward-predicted* (P) *macroblocks*
  - B frames: contain *Intra-coded* (I), *forward* (P) or *backward-predicted* (B) *macroblocks*

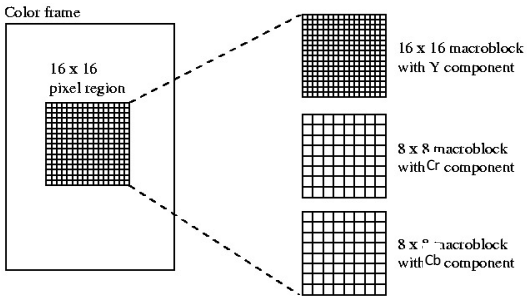
- I and D frames don't have motion prediction. Therefore they contain *Intra-coded* (I) *macroblocks* with blocks that contain direct encoding from the image samples (D frames are only DC encoded)

B Frame with macroblocks



Macroblock components

- Each macroblock is encoded separately.

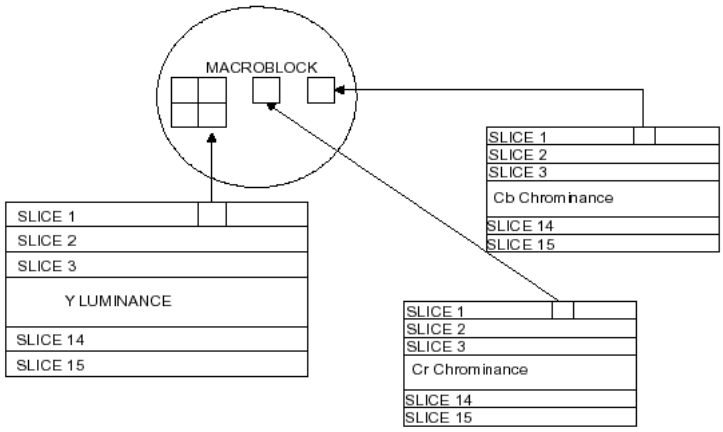


4:2:0 sampling

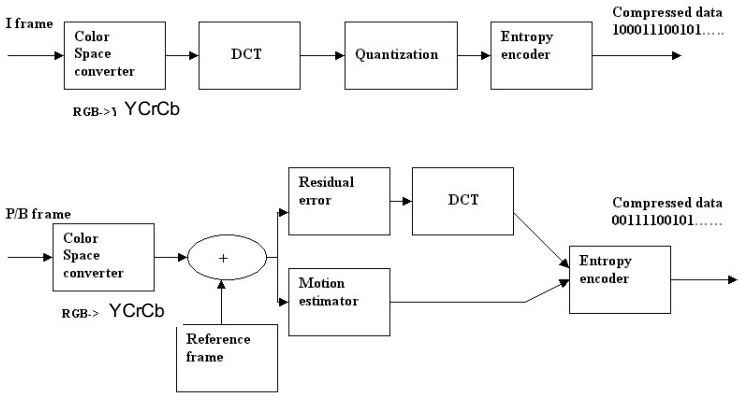
- The Y component of a macroblock is used for motion compensation. Cr and Cb are chrominance components.

Slices

- Macroblocks are organized into slices

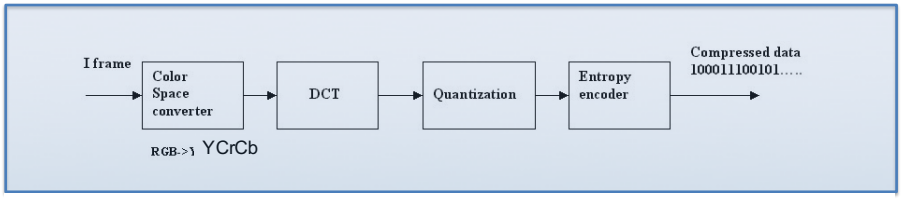


Encoding macroblocks

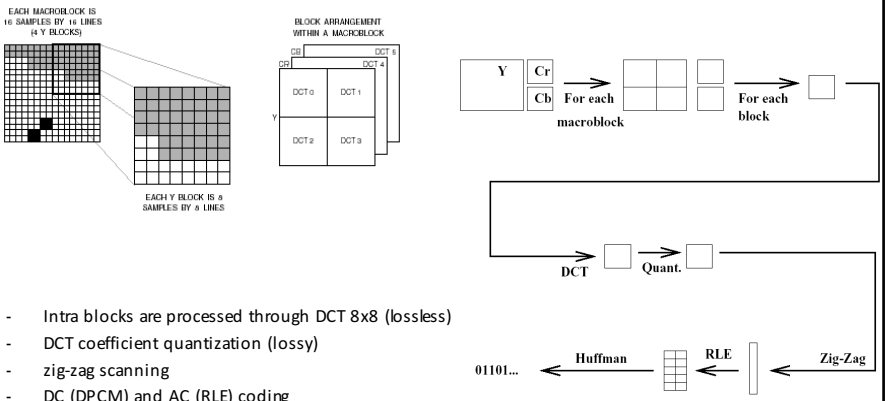


The block diagram of the MPEG encoder

I-macroblock coding



I-macroblock coding in detail



- Intra blocks are processed through DCT 8x8 (lossless)
- DCT coefficient quantization (lossy)
- zig-zag scanning
- DC (DPCM) and AC (RLE) coding
- Entropy coding (Huffman)

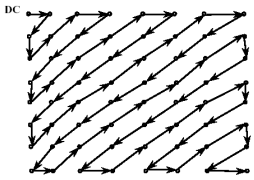
Spatially-Adaptive Quantization

- Spatially-adaptive quantization is made possible by the *factor quantizer_scale*.
  This parameter is allowed to vary from one macroblock to another within a picture to adaptively adjust the quantization on a macroblock basis.
- The default quantization matrix can be changed for each sequence.

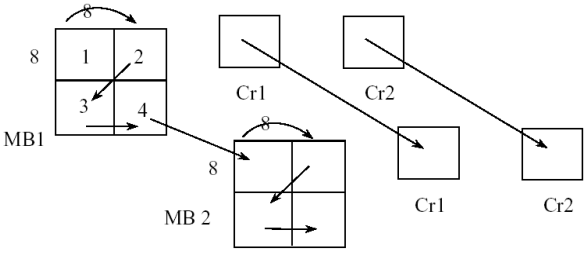
8	16	19	22	26	27	29	34
16	16	22	24	27	29	34	37
19	22	26	27	29	34	34	38
22	22	26	27	29	34	37	40
22	26	27	29	32	35	40	48
26	27	29	32	35	40	48	58
26	27	29	34	38	46	56	69
27	29	35	38	46	56	69	83

MPEG1 default quantization matrix

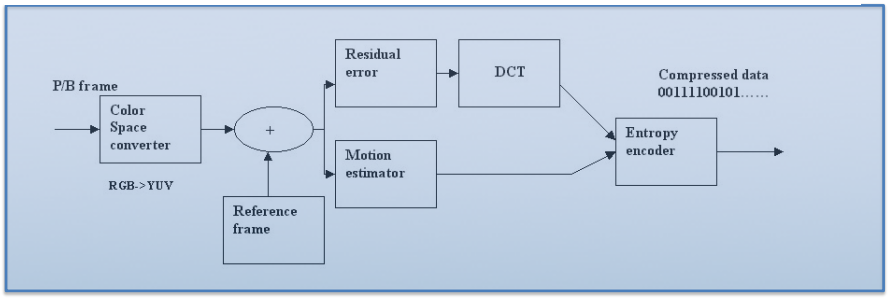
- zig-zag scanning is used to create a 1D stream



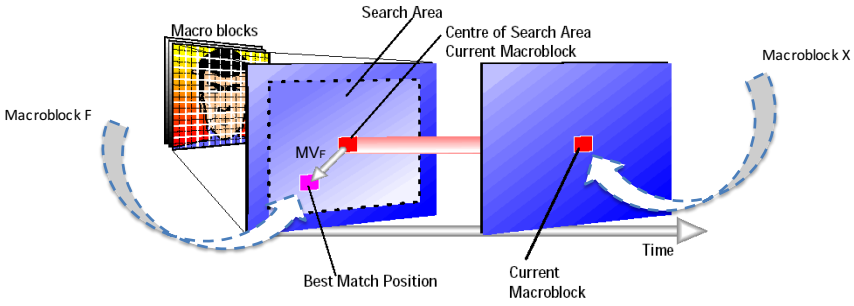
- AC coefficients are encoded losslessly according to *run length encoding* and Huffman coding. *Tables* are formed on a statistical *basis.* Different tables for Y and CbCr.
- DC coefficients encode differences between blocks of the macroblock:



P/B macroblock coding



Predictive encoding



- Predictive encoding aims to reduce the data transmitted by detecting the motion of objects. This will typically result in 50% - 80% savings in bits.
- Instead of sending quantized DCT coefficients of macroblock X:
  - Find the best-matching macroblock in the reference frame by searching an area and compare. Each macroblock can be assigned a match from either a *backward* (B) or *forward* (F) reference
  - Send quantized DCT coefficients of X-F (prediction error): if prediction is good, error will be near zero and will need few bits.
  - Encode and send the motion vector MV_F. This will be differentially coded with respect to its neighboring vector, and will code efficiently.

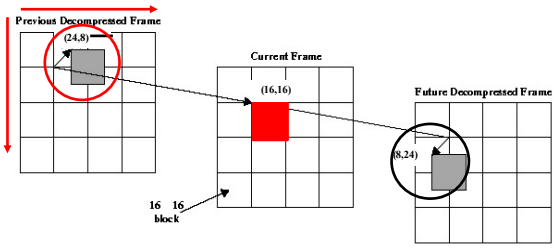
Block motion compensation

- The process of replacing macroblocks with a motion vector and the error block is referred to as *block motion compensation*. P and B macroblock coding is based on block motion compensation
  - A *motion vector* describes the transformation between the same (similar) macroblocks in adjacent frames in a video sequence. Motion vectors are assumed to be *constant over a macroblock*
  - The encoder must decide whether a macroblock is encoded as I o P. A possibile mechanism compares the variance of luminance of the original macroblock with the *error macroblock*. If variance is above a threshold a I macroblock is encoded.

Motion vectors

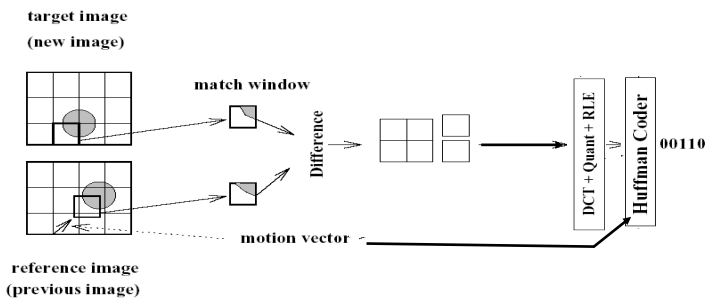
- A motion vector is specified with two components (horizontal and vertical offset). Offset is calculated starting from the top left pixel :
  - Positive values indicate moving right and bottom.
  - Negative values indicate moving left and top.
  - Set to 0,0 at the start of the frame or slice or I-type macroblock.
- P Macroblock have always a predictive base selected according to the motion vector. Absence of motion vector is indicated with (0,0): if motion vector is (0,0) the predictive base is the same macroblock in the reference frame
- Motion vectors are reset when a new I macroblock is found.

Example:
the match of the shaded macroblock of the current frame in the previous frame is in position (24,8). The forward predicted motion vector for the current frame is (8,- 8)
the match of the shaded macroblock of the current frame in the future frame is in position (8,24). The backward predicted motion vector for the current frame is (-8, 8)

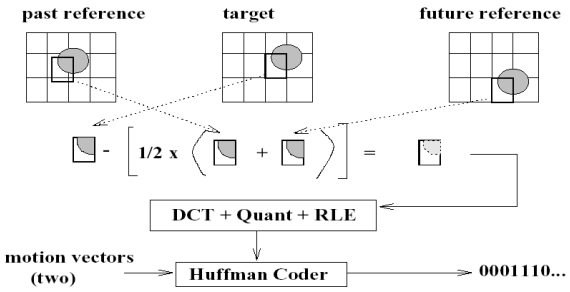


Error blocks

- P/B error blocks are obtained as the difference between two motion compensated blocks in adjacent frames. They are encoded as a normal block with a few differences wrt I blocks:
  - a different quantization matrix is used wrt I blocks: "16" value is set in all the matrix positions as error blocks have usually high frequency information
  - DC component and AC component are managed in the same way (there is no differential encoding as in I blocks)
- For a P macroblock:

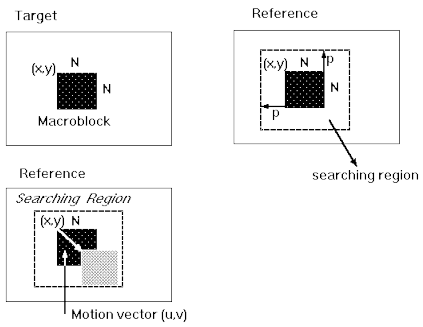


- For a B macroblock:



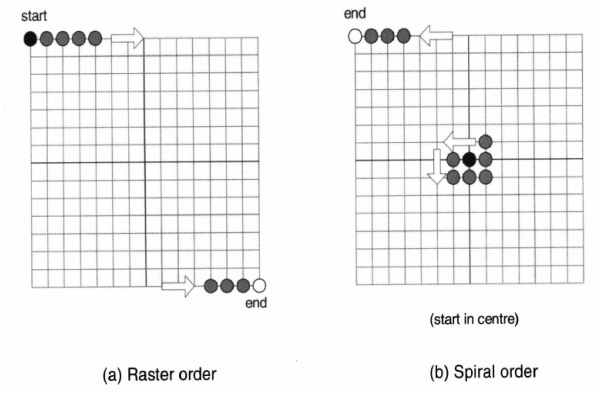
Motion estimation by block matching

- Motion estimation is performed by applying block matching algorithms. Different block matching techniques exist: often they limit the search area for matching.



Full search

- All the positions within the window are checked with a pre-defined criterion for block matching
  Computationally expensive, only suited for hardware implementation



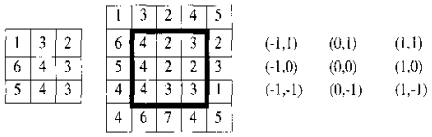
Mean Squared Error (MSE) criterion

- Mean Squared Error (MSE) (for N x N block):

$$\mathrm{MSE} = \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}(C_{ij} - R_{ij})^2$$

where C_{ij} is the sample in the current block and R_{ij} the sample in the reference block

Example:



$$\{(1-4)^2 + (3-2)^2 + (2-3)^2 + (6-4)^2 + (4-2)^2 + (3-2)^2 + (5-4)^2 + (4-3)^2 + (3-3)^2\}/9 = 2.44$$

block centered in MSE value:

Position (x, y)	$(-1, -1)$	$(0, -1)$	$(1, -1)$	$(-1, 0)$	$(0, 0)$	$(1, 0)$	$(-1, 1)$	$(0, 1)$	$(1, 1)$
MSE	4.67	2.89	2.78	3.22	2.44	3.33	0.22	2.56	5.33

minimum value

Mean Absolute Error/Difference (MAE/MAD) criterion

- Mean absolute error/difference (MAE/MAD): Easier wrt MSE

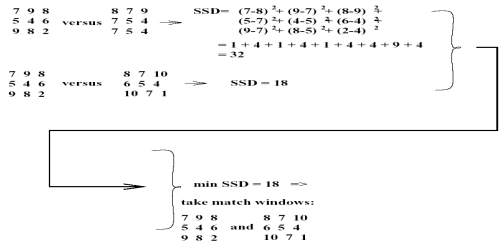
$$\mathrm{MAE} = \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1}|C_{ij} - R_{ij}|$$

- Matching pel count (MPC): similar pixels are counted in two blocks

Sum of Squared / Absolute Differences (SSD) (SAD)

- Sum of Squared Differences (SSD): Sensitive to outliers

$$SSD = \sum_i (x_i - y_i)^2$$



- Sum of absolute differences (SAD): Less sensitive wrt outliers wrt SSD

$$SAD = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}|C_{ij} - R_{ij}|$$

SSD vs. SAD

SSD:

```
7 9 8          8 7 10
5 4 6  versus  6 5 4      -> SSD = 18
9 8 2          10 7 1

7 9 8          8 7 10
5 4 6  versus  6 5 4      -> SSD = 40,017
9 8 2          10 7 202   Outlier
```

SAD:

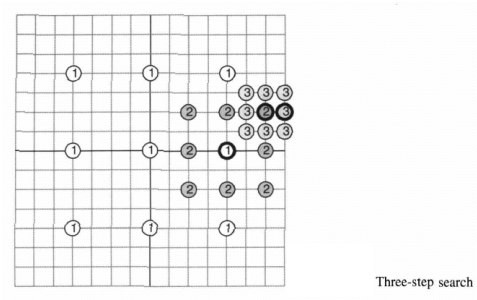
```
7 9 8          8 7 10
5 4 6  versus  6 5 4      -> SAD = 211
9 8 2          10 7 202
```

Fast search methods

- Full search always detects the global minimum of SAD
- As a less expensive alternative, fast search methods employ a reduced number of comparisons wrt full search but may fall into local minima:
  - Three step search
  - Logarithmic Search
  - One-at-a-Time Search
  - Nearest Neighbours Search

Three step search

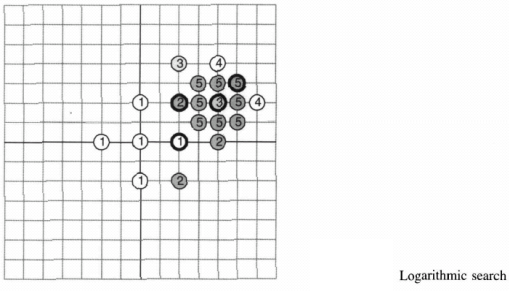
1. Start search from (0, 0).
2. Set S = 2^{N-1} (step size).
3. Look within 8 locations at +/-S pixel distance around (0, 0).
4. Select minimum SAD location between the 9 that have been analyzed
5. This location is the center for the new search
6. Set *S* = S/2.
7. Repeat from 3 to 5 until *S* = 1.



Three-step search

Logarithmic search

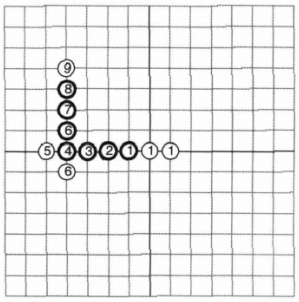
1. Start search from (0, 0).
2. Search in the 4 adjacent positions in the horizontal and vertical directions, at S pixel distance from (0,0) (S search step). The 5 positions model a ' + '.
3. Set the new origin at the best match. If best match is in the central position of '+' then *S* = S/2, otherwise *S* is not changed.
4. If *S* = 1 go to *5 , otherwise go to* 2.
5. Look for the 8 positions around the best match. Final result is the best match between the 8 positions and the central position



Logarithmic search

One-at-a-time search

1. Start from (0, 0).
2. Search at the origin and in the nearest positions horizontally
3. If origin has the lowest SAE then go to 5, otherwise. . . .
4. Set origin at the lowest SAE horizontally and search in the nearest position not yet checked and go to 3.
5. Repeat from 2 to 4 vertically.

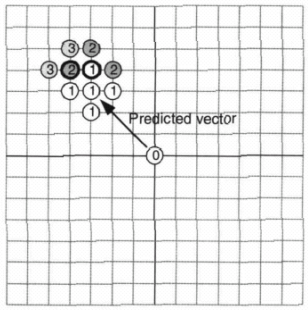


One-at-a-time search

Nearest neighbours search

- Used in H.263 e MPEG-4: motion vectors are predicted by the near vectors already coded. Assumes that near macroblocks have similar motion vectors

1. Start from (0, 0).
2. Set origin in the position of the predicted vector and start from there
3. Search in the nearest '+ '.
4. If the origin is the best then take this position as the correct one. Otherwise take the best match and proceed
5. Stop when the best match is at the center of ' + ' or at the border of the window.

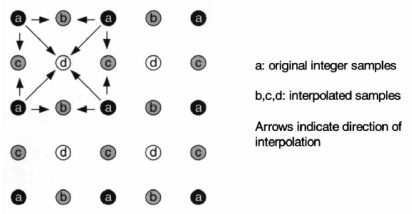


Block matching algorithms comparison

- Logarithmic search, Three step search e one-at-a-time have low computational complexity and low matching performance as well.
- Nearest-neighbours search, has good performance, similar to full search, and moderate computational complexity

Sub pixel motion estimation

- In some cases matching is improved if search is performed in a (artificially generated) region that is obtained by interpolating the pixels of the original region. In this case accuracy is sub-pixel.



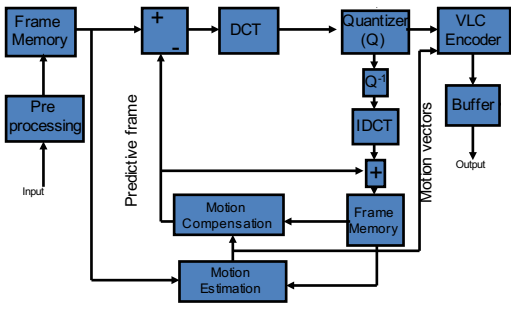
- Searching is performed as follows:
  1. Pixels are interpolated in the image search area so that a region is created with higher resolution than the original.
  2. Best match search is performed using both pixel and subpixel locations in the interpolated region
  3. Samples of the best matched region (full- o sub-pixel) are subtracted from the samples of the current block to obtain the error block.

MPEG1 encoding – decoding

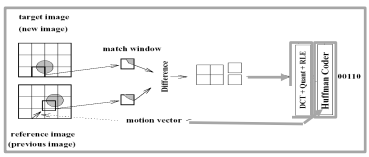
- In MPEG1 pictures are coded and decoded in a different order than they are displayed. This is due to bidirectional prediction for B pictures. The encoder needs to reorder pictures because B-frames always arrive late.

- Example: (a 12 picture long GOP)
  - Source order and encoder input order:
    I(1) B(2) B(3) P(4) B(5) B(6) P(7) B(8) B(9) P(10) B(11) B(12) I(13)
  - Encoding order and order in the coded bitstream:
    I(1) P(4) B(2) B(3) P(7) B(5) B(6) P(10) B(8) B(9) I(13) B(11) B(12)
  - Decoder output order and display order :
    I(1) B(2) B(3) P(4) B(5) B(6) P(7) B(8) B(9) P(10) B(11) B(12) I(13)

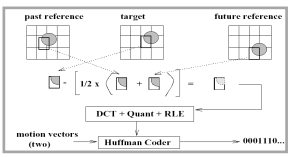
The MPEG1 encoder

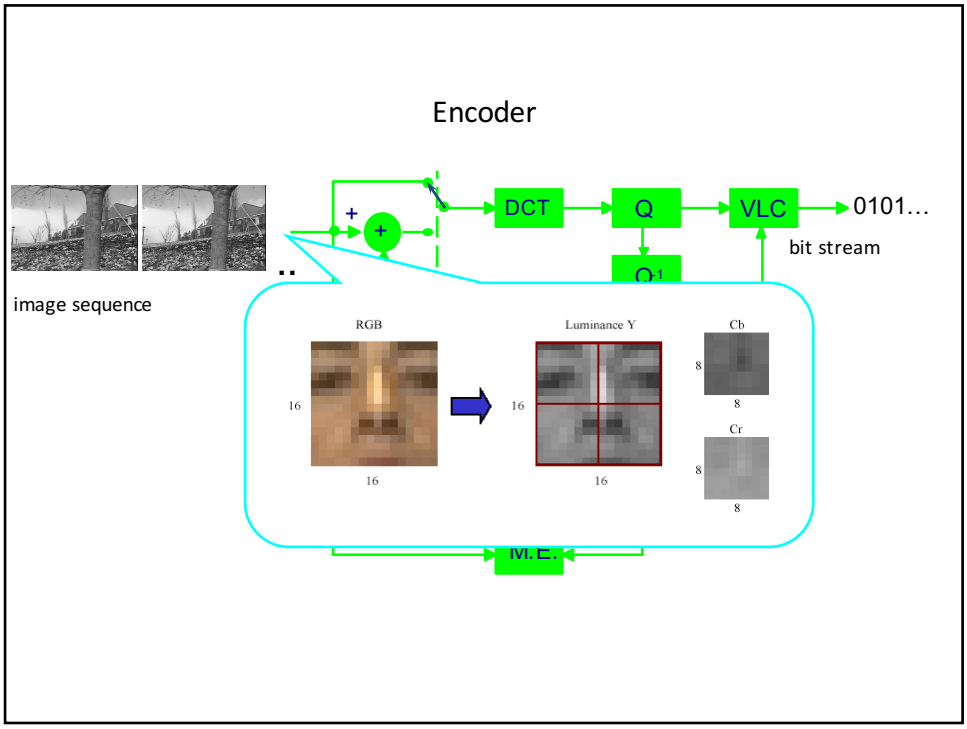


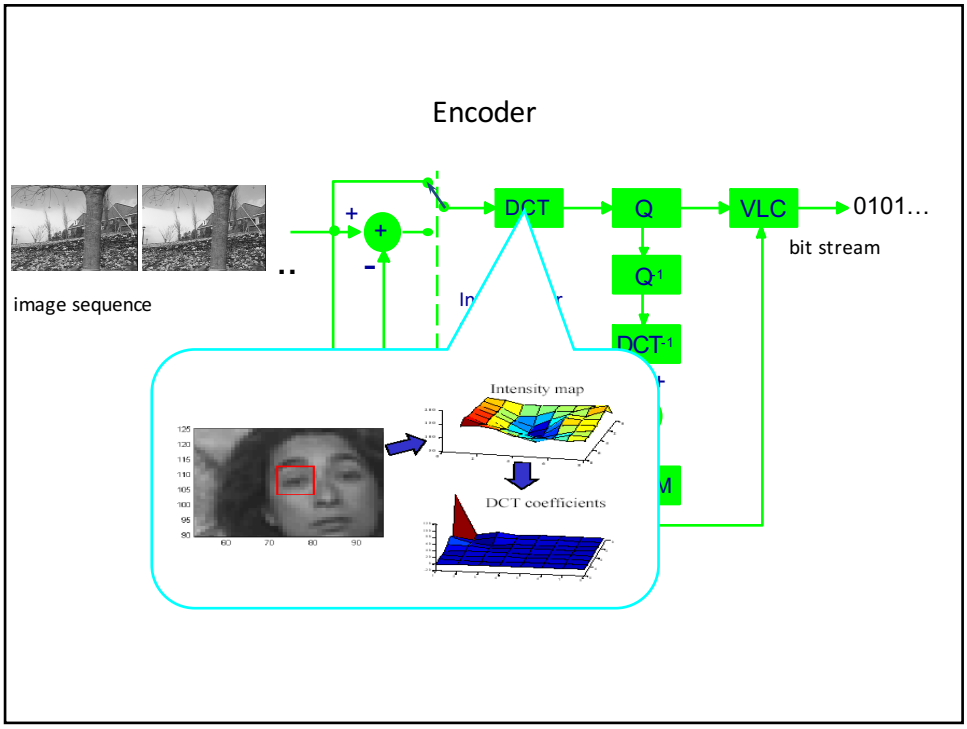
P macroblock

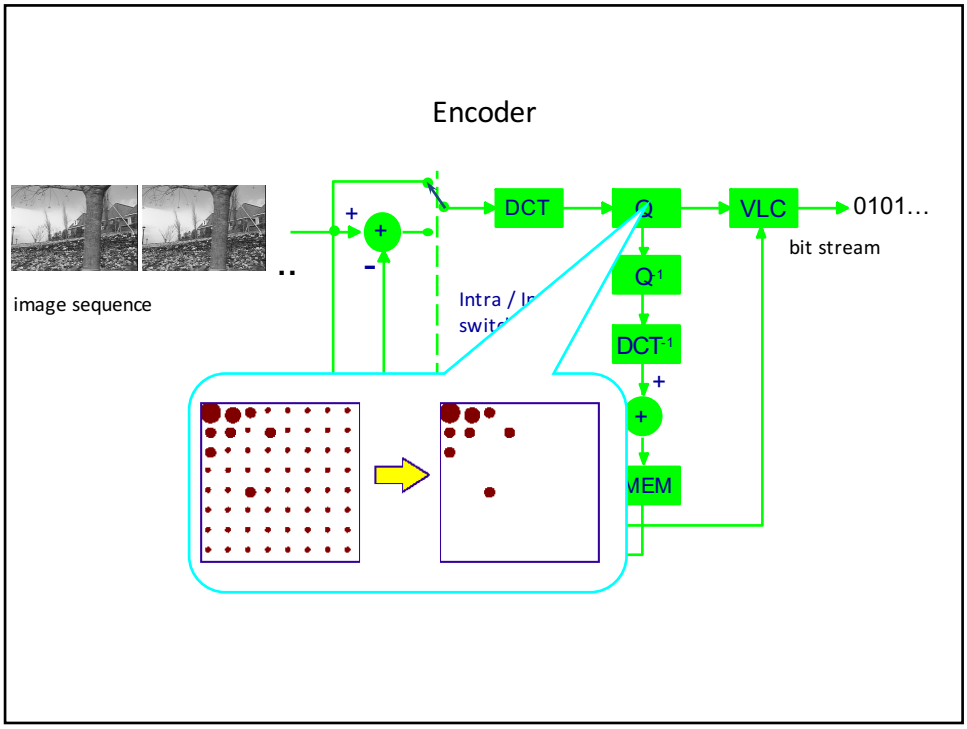


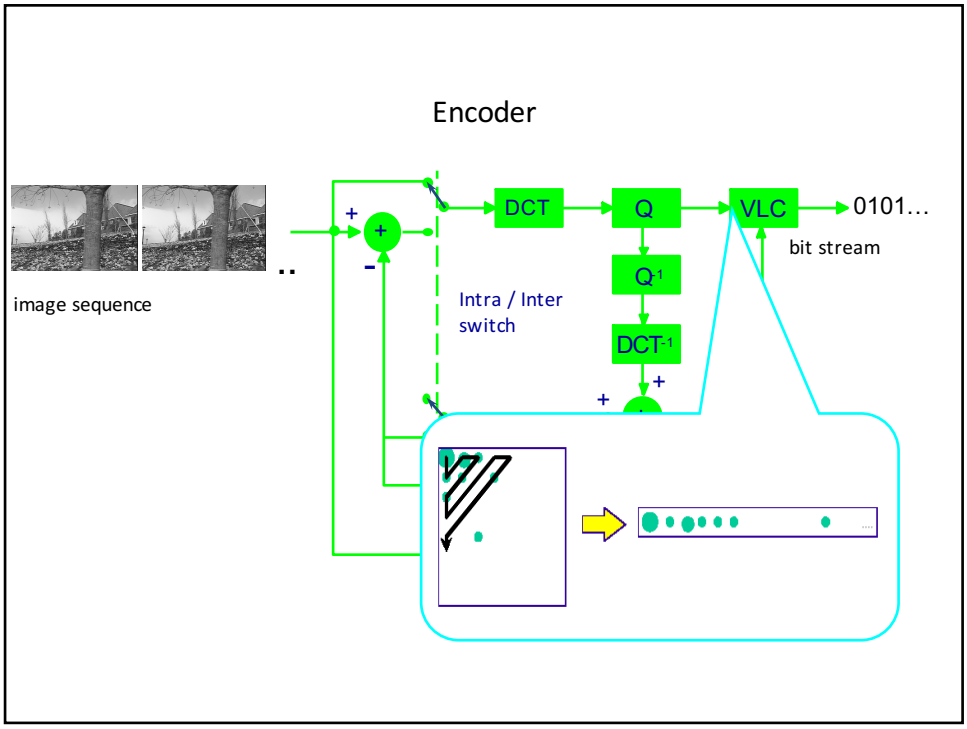
B macroblock



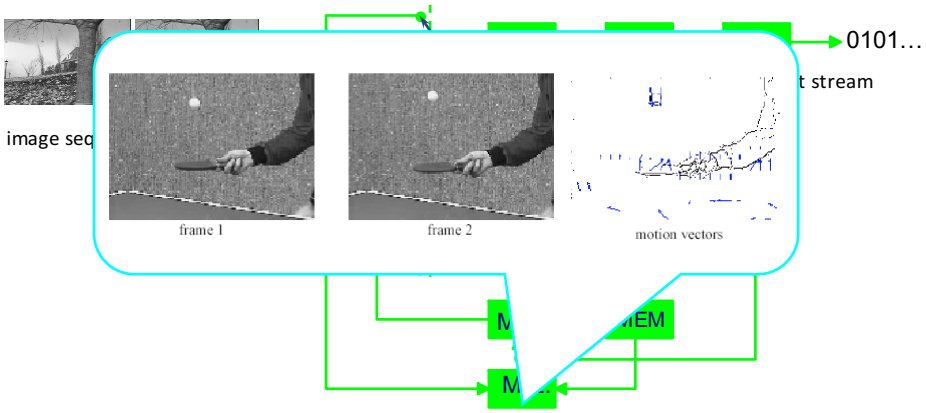

Encoder
DCT
Q
VLC
0101…
bit stream
image sequence
RGB
Luminance Y
Cb
Cr
16
16
16
16
8
8
8
8


Encoder
DCT
Q
VLC
0101…
bit stream
image sequence
Intensity map
DCT coefficients

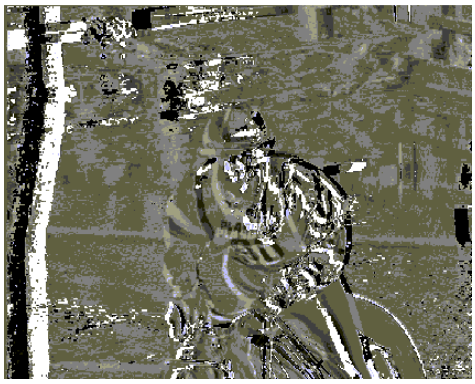

Encoder
image sequence
DCT
Q
VLC
0101…
bit stream
Q-1
DCT-1
MEM


Encoder
image sequence
DCT
Q
VLC
0101…
bit stream
Q-1
DCT-1
Intra / Inter
switch

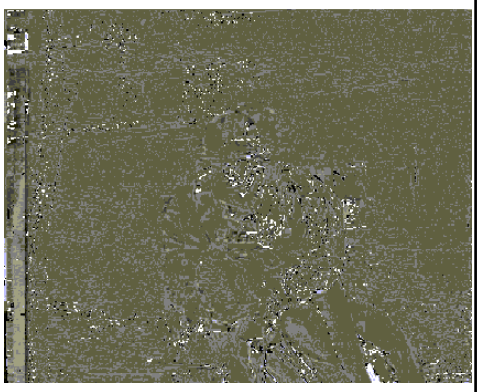
Encoder



- Frame N to be encoded
- Frame at t= N-1 used to predict content of frame N (with estimated motion vectors)



- Prediction error without motion compensation (the difference between frame N and frame N-1).



- Prediction error with motion compensation (the difference between frame N and frame N-1 shifted by the motion vector estimated)

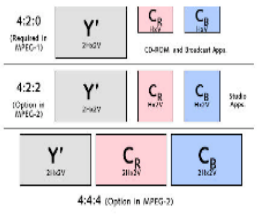
Part II - MPEG 2

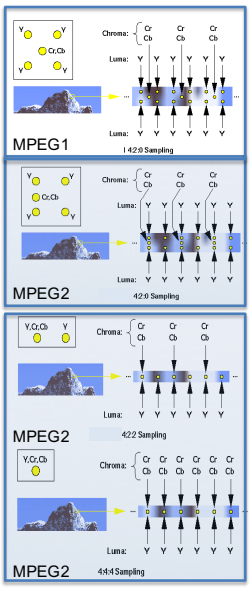
MPEG2 why another standard

- MPEG2 was designed as a superset of MPEG1 with support for broadcast video at 4-9 Mbps, HDTV up to 60 Mbps, CATV, S etc. Broadcast quality is obtained using fields instead of frames.
- MPEG-2 is widely used as the format of digital television signals that are broadcasted by terrestrial, cable, and satellite TV systems. It also specifies the format of movies and other programs that are distributed on DVD.
- MPEG-2 is similar to MPEG-1, but also provides support for interlaced video format. MPEG-2 video is not optimized for low bit-rates (less than 1 Mbit/s) but outperforms MPEG-1 at 4 Mbits and above
- MPEG2 features:
  - Interlaced and progressive video (PAL and NTSC)
  - Different color sampling modes: 4:2:0, 4:2:2, 4:4:4
  - Predictive and interpolative coding as in MPEG1
  - Flexible quantization schemes (can be changed at picture level)
  - Scalable bit-streams
  - Profiles and levels

Color subsampling

- MPEG2 supports different color subsamplings:
  - 4:2:0 (as MPEG1)
    In MPEG1 chrominance samples are horizontally and vertically positioned in the center of a group of 4 luminance samples.
    In MPEG-2 chrominance samples co-located on luminance samples
  - 4:2:2, 4:4:4
    Allow professional quality
    Use different macroblocks
    Different quantization matrices for Y and CrCb can be used with 4:2:2 and 4:4:4 sampling

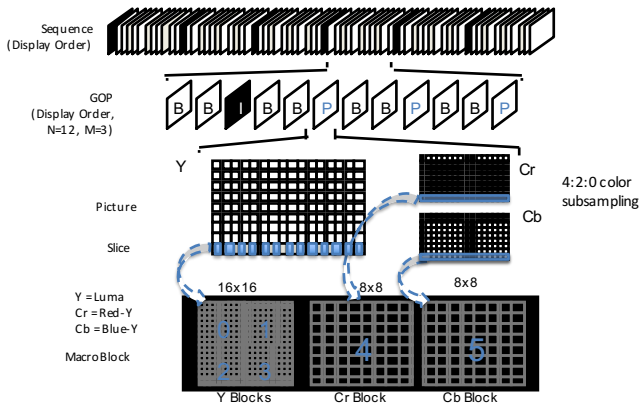




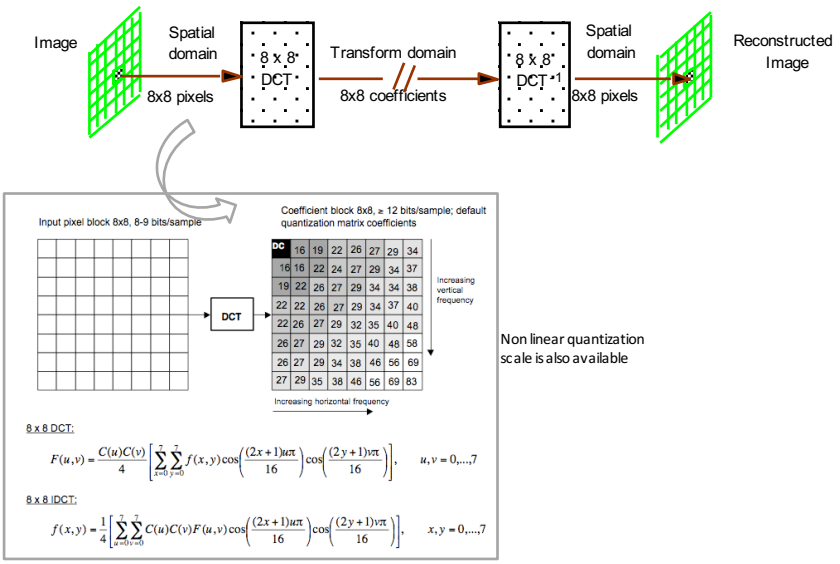
I, P, B frame encoding

- Same as MPEG1: I, P and B frames (pictures) are encoded on a macroblock basis using DCT:
  - P-pictures have interframe predictive coding:
    - Macroblocks may be:
      - coded with forward prediction from previous I and P pictures
      - intra coded
    - For each macroblock the motion estimator produces the best matching macroblock
    - The prediction error is encoded using a block-based DCT
  - B-pictures have interframe interpolative coding:
    - The motion vector estimation is performed twice (forward and backward).
    - Macroblocks may be coded with:
      - forward (backward) prediction from past (future) I or P references;
      - interpolated prediction from past and future I or P references;
      - intra coded
    - The encoder forms a prediction error macroblock from either or their average
    - The prediction error is encoded using a block-based DCT
- Differently from MPEG1 it has no D pictures

The MPEG2 stream

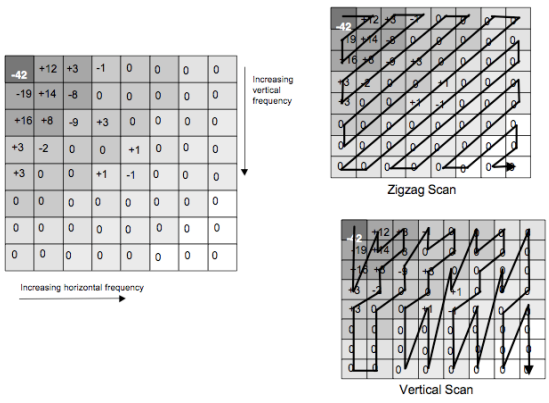


Discrete Cosine Transform and quantization scale



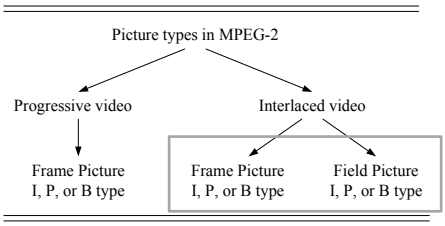
Multiple scanning options

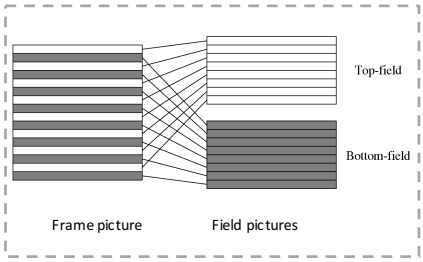
- Zig-zag scanning is accompanied with a different scanning that is better suited for interlaced frames



Support of progressive and interlaced video

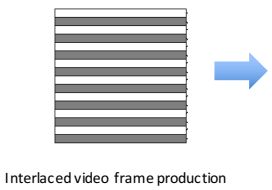
- With MPEG2:
  - progressive frames are encoded as *frame pictures* with frame-based DCT coded macroblocks only . The 8x8 blocks that compose the macroblock come from the same frame of video. Same as MPEG1.
  - interlaced frames may be coded as either a *frame picture* (*frame-based production*) or as *two separately coded field pictures* (*field-based production*). The encoder may decide on a frame by frame basis to produce a frame picture or two field pictures.

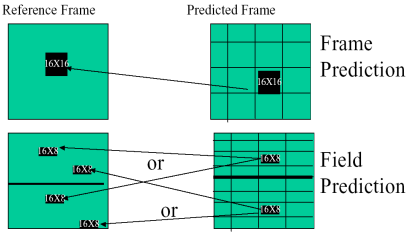




Frame and field prediction for produced frames

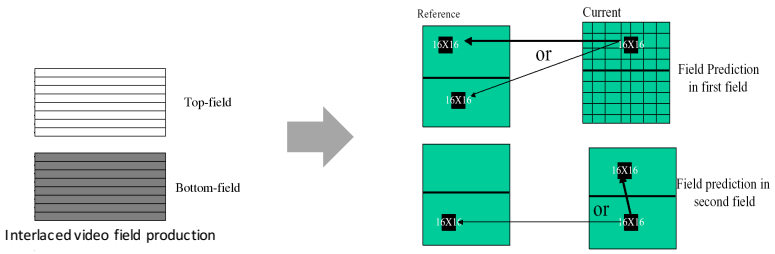
- With interlaced video it is possible to choose whether the luminance in the two fields must be encoded jointly or separately (frame prediction or field prediction). Frame or field-based DCT macroblock prediction are applied on a macroblock-by-macroblock basis.
- **Frame-based prediction**: identical to MPEG-1 prediction methods. Uses a single motion vector for each 16x16 macroblock.
- **Field-based prediction:** the top-field and bottom-field of a frame-picture are treated separately.
  - Each 16 × 16 macroblock from the target frame-picture is split into two 16×8 parts, each coming from one field.
  - Two motion vectors are used for each macroblock taken from either of the two most recently decoded anchors. The first motion vector is used for the upper 16x8 region the second for the lower 16x8 region.
  - Each field is predicted separately with its motion vectors.





Field prediction for produced fields

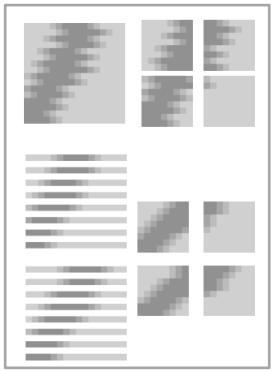
- For interlaced sequences and field-production at the encoder, field-based prediction must be used based on a macroblock of size 16 × 16 from field-pictures.



- Frame prediction is suited for macroblocks with little motion and high spatial activity.
- Field-based prediction is suited in the presence of fast motion. With field-based prediction motion vectors are evaluated on a half-pixel basis, so they are more precise and a better compression is obtained

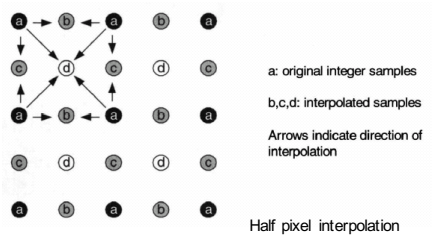
If there is fast motion it is possible that blocks obtained from separate encoding of the 8 lines of the top field and the 8 lines of the bottom field have higher correlation than the 4 blocks obtained from the 2 fields combined in a single frame.

Note that the size of 16 × 16 in the field picture covers a size of 16 × 32 in the frame picture. It is too big size to assume that behavior inside the block is homogeneous. Therefore, 16 × 8 size prediction was introduced in field picture.



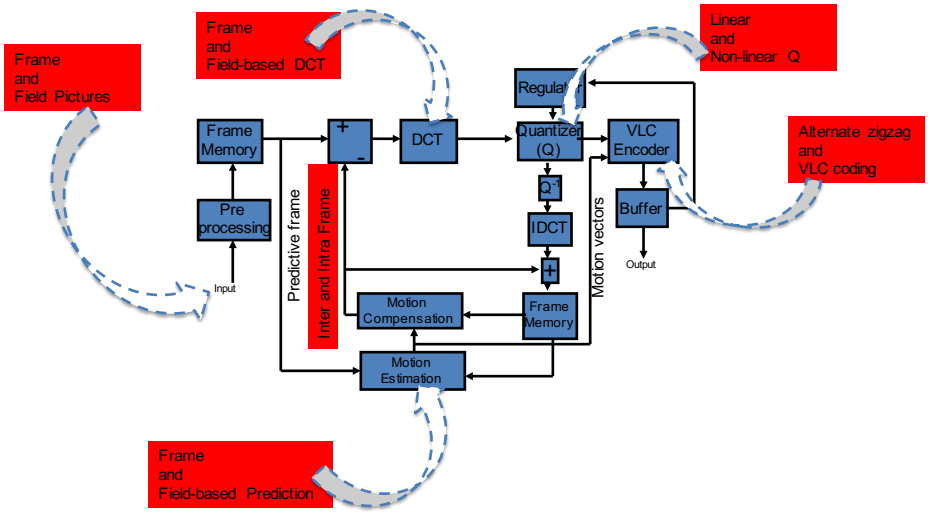
Half pixel interpolation for motion estimation

- MPEG2 uses half-pixel interpolation for motion vector estimation.
- Searching is performed as follows:
  - pixels are interpolated in the image search area so that a region is created with higher resolution than the original
  - best match search is performed using both pixel and subpixel locations in the interpolated region
  - samples of the best matched region are subtracted from the samples of the current block to obtain the error block



Half pixel interpolation

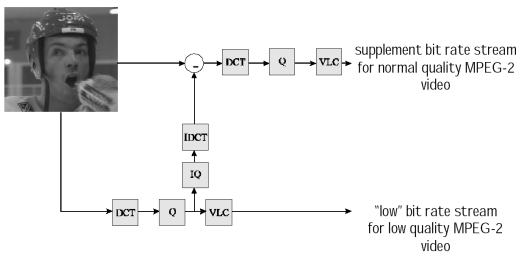
MPEG2 Enhancements



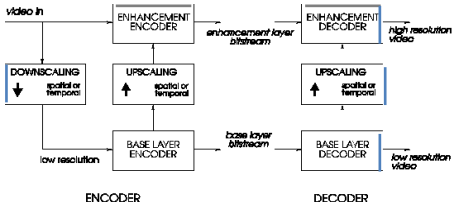
Scalability

- Scalability is the ability of decoding only part of the stream to obtain a video of the resolution desired. It is possible to have:
  - SNR scalability,
  - Spatial scalability
  - Temporal scalability
- Scalability mode permits interoperability between different systems (f.e. a HDTV stream is also visible with SDTV). A system that does not reconstruct video at higher resolution (spatial or temporal) can simply ignore data refinement and take the base version.

- SNR scalability (2 layers)
  - Suited for applications that require different degrees of quality
  - All layers have the same spatial resolution. The base layer provides the base quality, the enhancement layer provides quality improvements (with more precise data for DCT)
  - Permits "graceful degradation"



- Spatial scalability (2 layer)
  - Base layer at lower spatial resolution (MPEG1 can be used to encode the base layer)
  - Enhancement layer at higher resolution (obtained by spatial interpolation)
  - Upscaling is used to predict coding of the high resolution version. Prediction error is encoded in the enhancement layer bitstream



- Temporal scalability
  - Similar to spatial scalability, but referred to time
  - Base Layer : 15 fps
  - Enhancement layer : Supplements the remaining frames to achieve higher fps

Profiles and Levels

- In MPEG2 profiles and levels (profile@level) define the minimum capability required for the decoder:
  - Profiles: specify syntax and algorithms (define the compression rate and decoding complexity)
  - Levels: define parameters such as resolution, bitrate, etc.

Profiles

- Simple Profile (4:2:0)
  - For videoconferencing
  - Corrisponds to MPEG1 Main profile without B frame
- Main profile (4:2:0)
  - For videoprofessional SDTV (bitrate at 50 Mbps)
  - The most important; of general applicability
- Multiview profile
  - For multiple cameras filming the same scene.
- 4:2:2 profile
  - For video professional SDTV and HDTV (bitrate at 50 Mbps)
- SNR and Spatial Scalable profile (4:2:0)
  - Add SNR / spatial scalability SNR with different quality levels
- High 4:2:0 profile
  - Suitable for HDTV

Levels

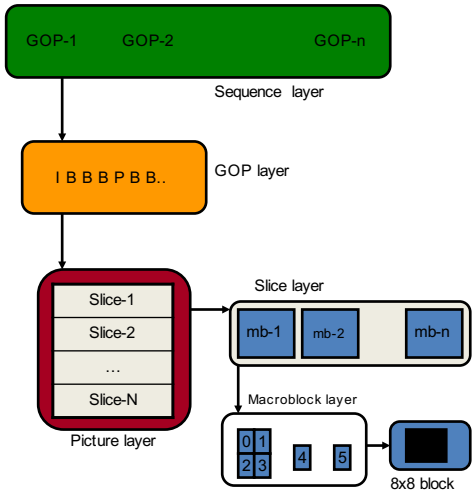
- Low Level
  - MPEG1 CPB (Constrained Parameters Bitstream): max. 352x288 @ 30 fps
- Main Level
  - MPEG2 CPB (720x576 @ 30 fps)
- High-1440 and High Levels
  - Typical of HDTV

Profiles@levels

Level	Profile				
	Simple 4:2:0	Main 4:2:0	SNR Scalable 4:2:0	Spatially Scalable 4:2:0	High 4:2:0 or 4:2:2
High 1920x1152 (60 frames/s)		62.7 Ms/s 80 Mbit/s			100 Mbit/s for 3 layers
High-1440 1440x1152 (60 frames/s)		47 Ms/s 60 Mbit/s		47 Ms/s 60 Mbit/s for 3 layers	80 Mbit/s for 3 layers
Main 720x576 (30 frames/s)	10.4 Ms/s 15 Mbit/s	10.4 Ms/s 15 Mbit/s	10.4 Ms/s 15 Mbit/s for 2 layers		20 Mbit/s for 3 layers
Low 352x288 (30 frames/s)		3.04 Ms/s 4 Mbit/s	3.04 Ms/s 4 Mbit/s for 2 layers		

MPEG2: Structure of the bit-stream

- *Sequence layer*: picture dimensions, pixel aspect ratio, picture rate, minimum buffer size, DCT quantization matrices
- *GOP layer*: will have one I picture, start with I or B picture, end with I or P picture, has closed GOP flag, timing info, user data
- *Picture layer*: temporal ref number, picture type, synchronization info, resolution, range of motion vectors
- *Slices*: position of slice in picture, quantization scale factor
- *Macroblock*: position, H and V motion vectors, which blocks are coded and transmitted



MPEG2 criticals

- There are several conditions that are critical for MPEG2 compression:
  - Zooming
  - Rotations — determine *mosquito noise*
  - Non-rigid motion
  - Dissolves and fades — determines *blockiness*
  - Shadows
  - Smokes
  - Scene cuts
  - Panning across crows — determine *wavy noise*
  - Abrupt brightness changes
  - …….

Part III – H.264

H264

- The motivations for H.264 standard are that digital representation of the Television signals created many different services for the content delivery, namely Satellite, Cable TV, Terrestrial Broadcasting, ADSL and Fiber on IP with contrasting requirements:

- To optimize this services, there is the need of:
  - High Quality of Service (QoS)
  - Low Bit-Rate
  - Low Power Consumption

- The H.264 standard is designed for technical solutions including at least the following application areas
  - Broadcast over cable, satellite, Cable Modem, DSL, terrestrial, etc.
  - Interactive or serial storage on optical and magnetic devices, etc.
  - Conversational services over Ethernet, LAN, wireless and mobile networks, modems, etc. or mixtures of these.
  - Video-on-demand or multimedia streaming services, wireless networks, etc.
  - Multimedia Messaging Services (MMS) over Ethernet, wireless and mobile networks, etc.
  - …

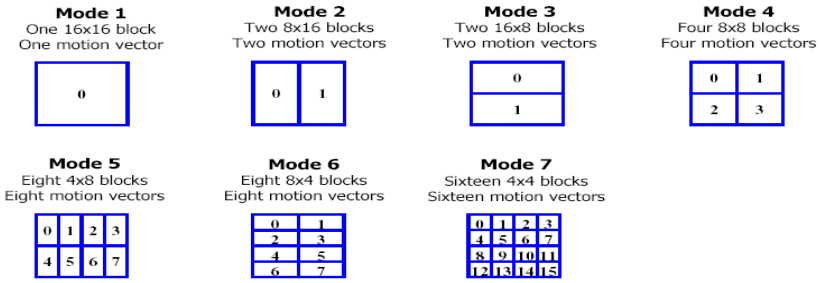
- The H.264 standard provides good video quality at a lower bitrate wrt MPEG2 with no additional cost of implementation or complexity. Its design represents a delicate balance between:
  - Coding Gain (improved efficiency by a factor of two over MPEG-2)
  - Implementation complexity
  - Costs based on state of VLSI design technology

- H.264 is licenced by MPEG LA company. MPEG LA permits free use of H264 for streaming video over the Internet to final users.

- Apple has officially adopted H.264 as the format for QuickTime

Motion Estimation & Compensation

- Motion Estimation is where H.264 makes most of its gains in coding efficiency.
  - **Quarter pixel accurate motion compensation.**
  - Translation only.
  - The standard does not determine which algorithm should be used.
  - **A number of different block sizes are used for motion prediction.**
  - Seven optional modes

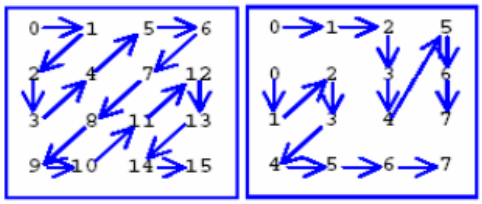


Different modes of dividing a macroblock for motion estimation in H.264

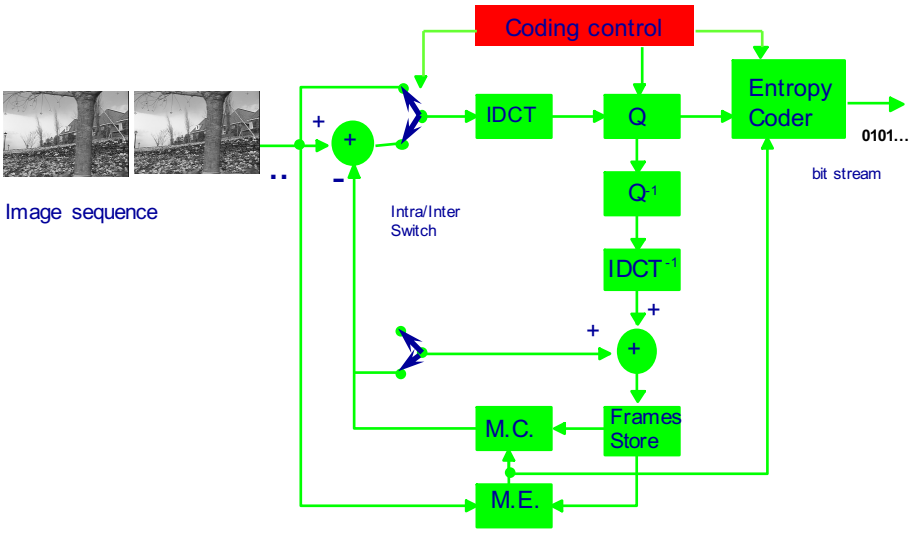
Transform and quantization

- DCT like.
- Integer arithmetic only without multiplications.
- Optional use of a 4x4 transform block size.

- Compounding quantization step.
- Thirty-two different quantization step sizes.
  - The step sizes are increased at a compounding rate of approximately 12.5%.
- Different quantizer for luminance, chrominance.
- Two different coefficient-scanning patterns.
  - The simple zigzag scan.
  - The double scan.



H.264 Encoder



Part IV - MPEG 4

MPEG4

- MPEG4 has been designed for.
  - Real-time communication (videoconferencing)
  - Digital television
  - Interactive graphic applications (DVD, ITV);
  - digital video composition, manipulation, indexing, and retrieval
  - World Wide Web applications

  Uses of MPEG-4 include compression of AV data for web (streaming media).

- Provides effective solutions for: authors, service providers, final users. To this end it:
  - **adopts a object-based coding**
  - allows higher compression ratio, but also supports covers a wide range of bitrates between 5 kbps to 10 Mbps
  - Supports Very Low Bit-rate Video: algorithms and tools for applications at 5 e 64 kbits/s: sequences at low spatial resolution and low frame rate (up to 15 fps).

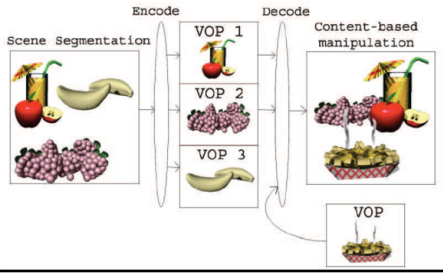
- MPEG-4 is an evolving standard and is divided into a number of parts. The key parts are **MPEG-4 Part 2** (including Advanced Simple Profile) and **MPEG-4 part 10** (also referred to as H.264 video coding).
- Most of the features included in MPEG-4 are left to individual developers so that there are probably no complete implementations of the full MPEG-4.

MPEG4 distinguishing elements

- MPEG4 distinguishes:
  - **Video-object Sequence (VS):** delivers the complete MPEG-4 visual scene, which may contain 2D natural or 3D synthetic objects
  - **Video Object (VO):** an object in the scene, which can be of arbitrary shape corresponding to an object or background of the scene (must be tracked)
  - Video Object Layer (VOL): facilitates a way to support (multi-layered) scalable coding. A Video Object can have multiple VOLs under scalable coding or have a single VOL under non-scalable coding
  - **Video Object Plane (VOP):** a snapshot of a Video Object at a particular moment
  - Group of Video Object Planes (GOV): groups Video Object Planes together (optional level)

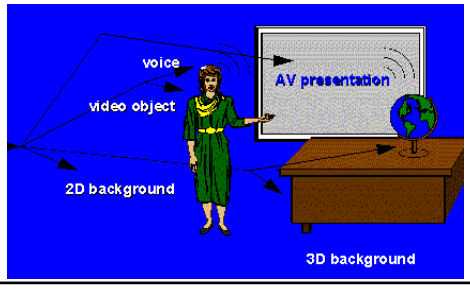
Main features on client and server sides

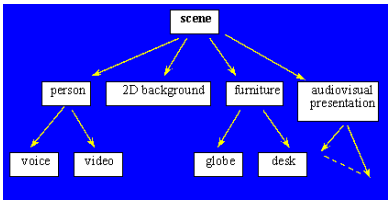
- MPEG4 includes technologies to support:
  - **server side**
    - Encoding based on and audio-visual objects. When a VOP is the rectangular frame it corresponds to MPEG2
    - Audio-visual objects manipulation
    - Hierarchical scene composition (audio-visual objects local coordinates, temporal syncronization….. described as an acyclic graph)
    - Multiplexing and syncronization of audio-visual objects and audio-visual objects transfer with appropriate QoS
  - **client side**
    - Audio-visual objects manipulation: display primitives to represent objects (2D and 3D, color, contrast change, talking 3D heads, head moving, 3D body animation..), syntethize speech from text, add objects, drop objects…..
    - User interactivity (viewpoint change, object clicking…)



Scene composition (server side)

- Scene Composition permits to:
  - Drop, change the position of audio-visual objects in a scene
  - Cluster audio-visual objects and form composite audio-visual objects that can be manipulated as a single audio-visual object
  - Associate parameters (motion, appearance) to audio-visual object and modify their attributes in a personalized way
  - Change the viewpoint of a scene



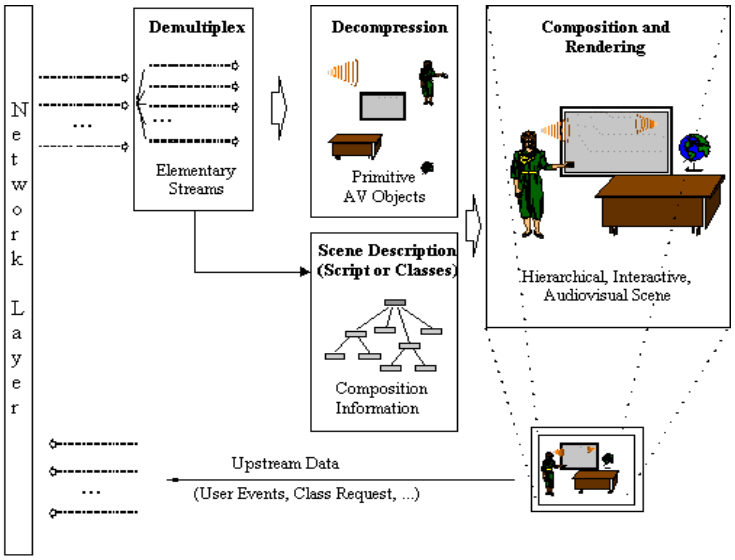


BInary Format for Scene description

Binary language derived from VRML

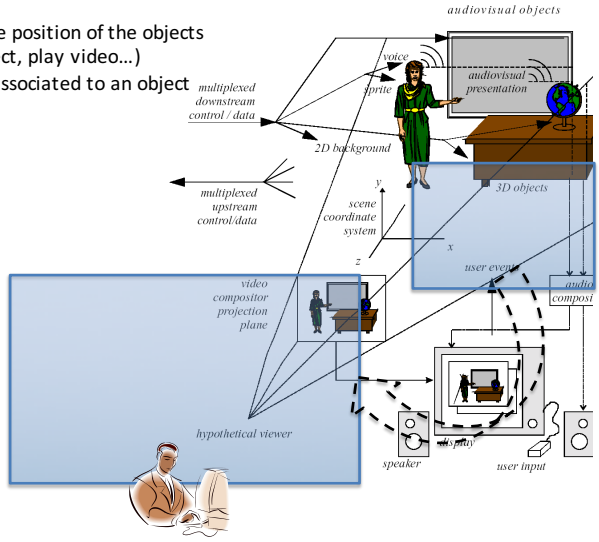
Scene description is encoded separately from the rest of the stream. It does not include parameters that are referred to audio-visual objects (like motion…)

Decoding (client side)



Interactive display (client side)

- Users can interact with the scene displayed through:
  - Navigation of the scene
  - Dropping or changing the position of the objects
  - Start actions (select object, play video…)
  - Selecting the language associated to an object



MPEG4 video compression

- MPEG-4 Part 10 specifies a compression format for video signals which is **technically identical to the ITU-T H. 264 standard.**

Profiles and levels

- MPEG4 profiles define resolution, bitrate and number of the objects that can be coded separately
  - *Simple profile*: for visual rectangular objects (suited for mobile terminals)
  - *Simple scalable profile*: like simple profile, but with temporal and spatial scalability (suited for internet services)
  - *Core profile*: with support of objects of any form with temporal scalability
  - Other profiles support: Facial animations; Audio; Meshes; Graphics…
- Levels define different degrees of computational complexity and quality

Profile	Level	Typical picture size	Bit-rate (bits/sec)	Max number of objects
Simple	1	176 × 144 (QCIF)	64 k	4
	2	352 × 288 (CIF)	128 k	4
	3	352 × 288 (CIF)	384 k	4
Core	1	176 × 144 (QCIF)	384 k	4
	2	352 × 288 (CIF)	2 M	16
Main	1	352 × 288 (CIF)	2 M	16
	2	720 × 576 (CCIR601)	15 M	32
	3	1920 × 1080 (HDTV)	38.4 M	32

Useful for

- MPEG4 is useful for:
  - Multimedia authors: permits to produce content with object-based flexibility wrt to single tecnologies such as digital television, graphic animation, web pages …..
  - Network providers: provides object and media -based infomation that can be appropriately processed and exploited
  - Final users: provides interactive object-based facilities, suited for real-time, surveillance, mobile applications
- Most of MPEG4 features are optional and their implementation is left to the developer. Most of the software for MPEG4-coded multimedia files do not support all the features. Profiles help to understand what features are supported.

OTHER COMPRESSION STANDARDS

High Efficiency Video Coding

- High Efficiency Video Coding (HEVC) is one of several potential successors to H.264 or MPEG4. It offers about double the data compression ratio at the same level of video quality and supports resolutions up to 8192×4320
- Similarly to H.264/MPEG-4 AVC it looks for areas that are redundant, both within a single frame as well as subsequent frames and replace them with a short description instead of the original pixels. The primary changes for HEVC include:
  - the expansion of the pattern comparison from 16×16 pixel to sizes up to 64×64
  - improved variable-block-size segmentation
  - improved "intra" prediction within the same picture
  - improved motion vector prediction and motion region merging
  - improved motion compensation filtering
- The first version of HEVC was published in June 2013. The second version was published in early 2015. Additional 3D-HEVC extensions for 3D video were completed in early 2015
- Further extensions are expected in 2016, covering video containing rendered graphics, text, or animation

VP8

- WebM Project is a Google sponsored project aimed at creating a free video format distributed as open source with high quality video compression to be used with HTML5. It includes the VP8 video codec by On2 Technologies, and the audio codec Vorbis.
- Although WebM is a great codec that has no chance to supplant H.264 because of absent device support/application integration

Video files formats

- A video file format is like an envelop that contains video data. It might support several algorithms for compression. A file in some format can be transcoded into another format: in this case the header is changed and the other data (if possible) are simply copied.

- Most common video formats:
  - Apple Quicktime (multiplatform) .mov
  - Microsoft AVI .avi
  - Windows Media Video .wmv
  - MPEG (multiplatform) .mpg o .mpeg

- Streaming video formats (for live video):
  - RealMedia (RealAudio e RealVideo) .rm
  - Microsoft Advanced System Format .asf
  - [Flash Video .swf]