Part II – Video

- General Concepts
- MPEG1 encoding
- MPEG2 encoding
- MPEG4 encoding

Video General Concepts

Video generalities

- Video is a sequence of frames 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.

- Video follows different modes of being formed and delivered, namely *analog* and *digital*, and consequently different standards.

- Distinguishing aspects of video are:
  - Color spaces
  - Color encoding
  - Color sampling rate
  - Video bandwidth

Analog and digital video

- *Analog video* is a video signal transferred by analog signal. It contains the luminance (brightness) and chrominance (color) of the image. No more in use in Italy from 2012.

- *Digital video* was initially obtained in the late 1970s by digitizing a standard analog video input to enhance the video signal and add effects to the video.
  - Digital video was introduced commercially in 1986 with the Sony D1 format, which recorded an uncompressed video signal in digital form, hence followed by cheaper systems using compressed data, most notably Sony's Digital Betacam.
  - With computers, digital video content creation tools initially required an analog video source to be digitized to a computer-readable format.
  - Digital video increased rapidly in quality with the introduction of MPEG-1 and MPEG-2 standards (adopted for use in television transmission and DVD media), and then of the DV tape format allowing recording direct to digital data and simplifying the editing process.

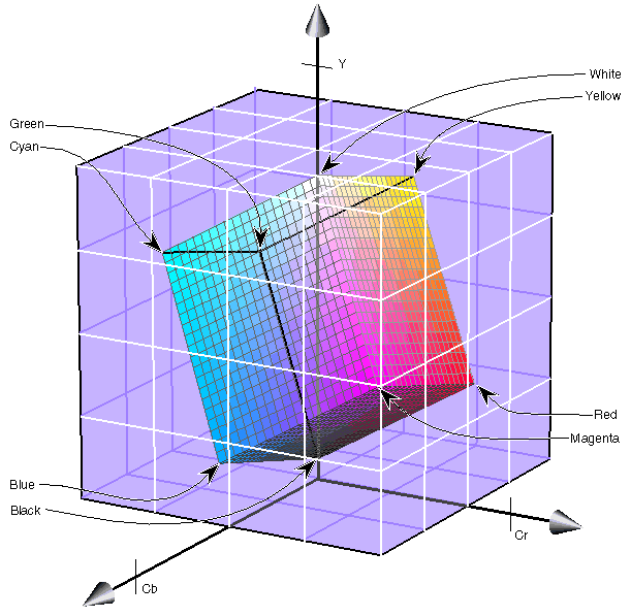
Video color spaces

- *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. Video is therefore transmitted and stored using color spaces that distinguish instead *brightness* and *chrominance* information.

- Color spaces for analog video are YUV or YIQ.
- Digital video is coded in YCrCb. Colors are distorted passing from RGB to YCrCb color space:
  - Brightness Y is obtained as a combination of R G and B signals.
  - Chrominance information is obtained instead subtracting Y from R and B signals.

YUV e YCrCb are similar but differ in the range of Y component values:

- YUV: from 0 to 255
- YCrCb: from 16 to 235/240



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*.

- Analog video signal is either transferred with composite or component encoding. Quality of component is usually better than composite.
- Digital video uses component color encoding.

Video sampling

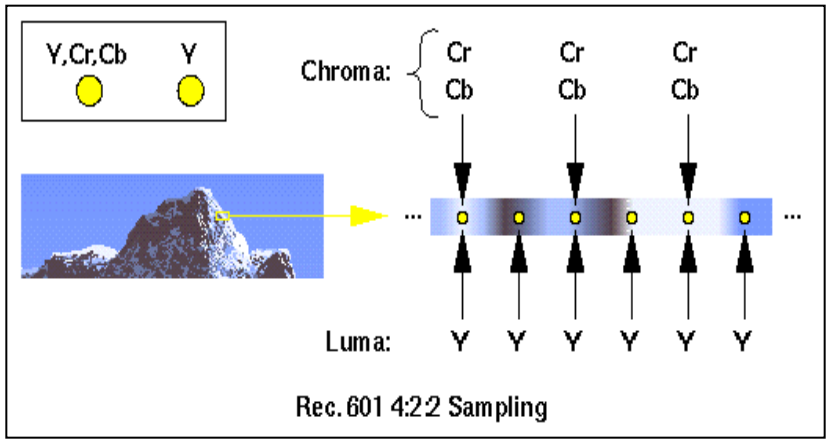
- Sampling is a mechanism for data compression in video. It applies to luminance and chroma information in each video frame. Because the human visual system is less sensitive to the position and motion of color than luminance, bandwidth can be optimized by storing more luminance detail than color detail.

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

  Es. 4:2:2 means that every 4 samples of luma, 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

- Video compression algorithms are also available like MPEG1, MPEG2…

For each line



Video bandwidth and bitrate

- *Bandwidth* is the frequency range of the video signal measured in MHz. The higher the bandwidth is the more information is carried on. Standard TV signal has about 5.5 MHz bandwidth.

- Bandwidth is directly related to video resolution. For digital video we use the term *bitrate* (the number of bits that are conveyed or processed per unit of time, measured in bits per second) as the equivalent of bandwidth:
  - 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 ≤ Nyquist rate = 2 · bandwidth
Band-pass:	Bitrate ≤ Bandwidth

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 example video has the following properties:
  - 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

- When compressing video we aim 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

Video formats

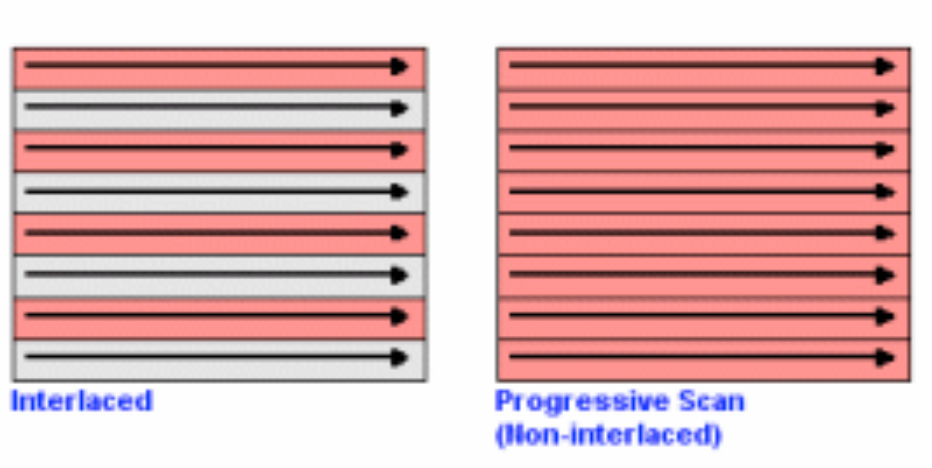
Analog video formats: PAL, NTSC, SECAM, S-VIDEO….

- There are three main systems of anolog color video broadcast transmission (television):
  - NTSC (North America, Japan)
  - PAL (most Europe, Australia, South Africa)
  - SECAM (France, Eastern Europe and Middle East)

- Standard for analog video cable transmission are:
  - S-Video….
- Standard for analog video registration are:
  - VHS, Betacam…

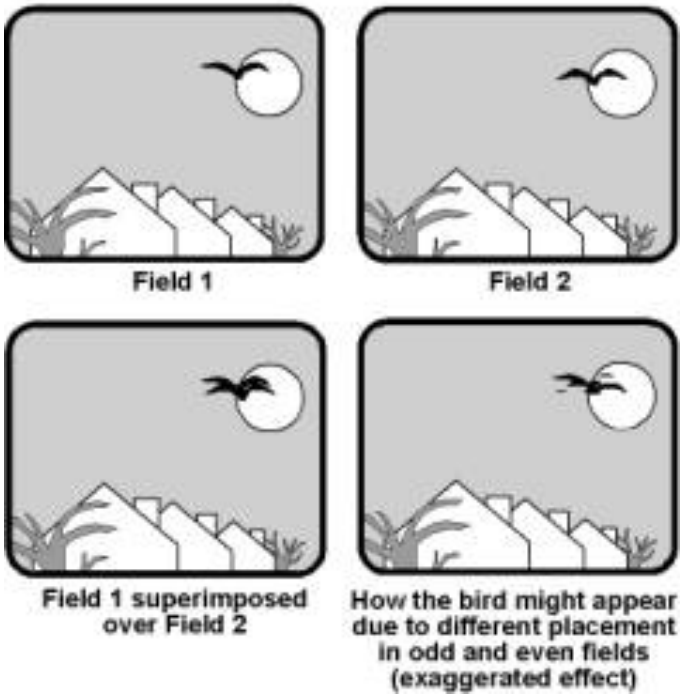
Interlace and progressive scan

- A television or recorded video image is basically made up of *scan lines* or *pixel rows* displayed across a screen starting at the top of the screen and moving to bottom. These lines or pixel rows can be displayed in two ways:
  - *By interlaced scan*: is to split the lines into two *fields* in which all of the odd numbered lines or pixel rows are displayed first and then all of the even numbered lines or pixel rows are displayed next, in essence, producing a complete frame.
  - *By progressive scan*: allows the lines to displayed sequentially. This means that both the odd and even numbered lines are displayed in numerical sequence (720 or 1080 pixels).



Video fields

- Fields have been used historically due to the limited bandwidth of the TV signal (5,5 MHz). Fields are displayed interlaced i.e. first the odd, then the even lines… Frequency is such that two fields are perceived as a single image.
- Data in a video field are distinguished both spatially and temporally. At each time instant one half of the information is lost.



- By applying “progressive” scanning rather than "interlacing" alternate lines a smoother, more detailed image can be produced on the screen

PAL, NTSC, SECAM

- PAL (Phase Alternate Line) uses 625 horizontal lines at a field rate of 50 fields per second (or 25 frames per second). For Au, NZ, UK, Europe:

  312 lines (290 active) per field, 576 pixels per line (625 lines in total)

- NTSC (National Television Standards Committee) is a black-and-white and color compatible 525-line system that scans interlaced television picture frames at ~60 field/sec (nominal 29.97 frames per second) . For USA, Canada, Japan:

  262 lines (242 active) per field, 483 pixels per line (525 lines in total)

- SECAM, (Sequential Couleur avec Memoire or sequential color with memory) uses the same bandwidth as PAL but transmits the colour information sequentially. (France, East Europe…)

- NTSC, PAL, SECAM are known as *composite video* because the brightness and color information are mixed together into a single signal. Color information of composite analog signals is coded in YUV (PAL) and YIQ (NTSC). Chrominance information is given in UV (IQ) and combined in a chroma signal, that is in its turn combined with luma Y.

- Having a composite signal is troublesome when the analog video is digitized in that it is difficult to separate the two signals.





- S-Video, Super-video and S-VHS transmit separate luminance Y and chroma C (Y/C *component color*). Y/C is commonly used to transmit video via cable between devices. It was developed by the VTR industry to support higher quality for video professionals. It is recommended that S-video is used instead of composite video.



Digital video formats: HDTV

- HDTV (High Definition TeleVision) was finalized in the 90's with Recomm.709:
  - High resolution: digital video format 1125 x 660 pixels per frame
  - Aspect ratio: 16:9 instead of 4:3 of NTSC and PAL

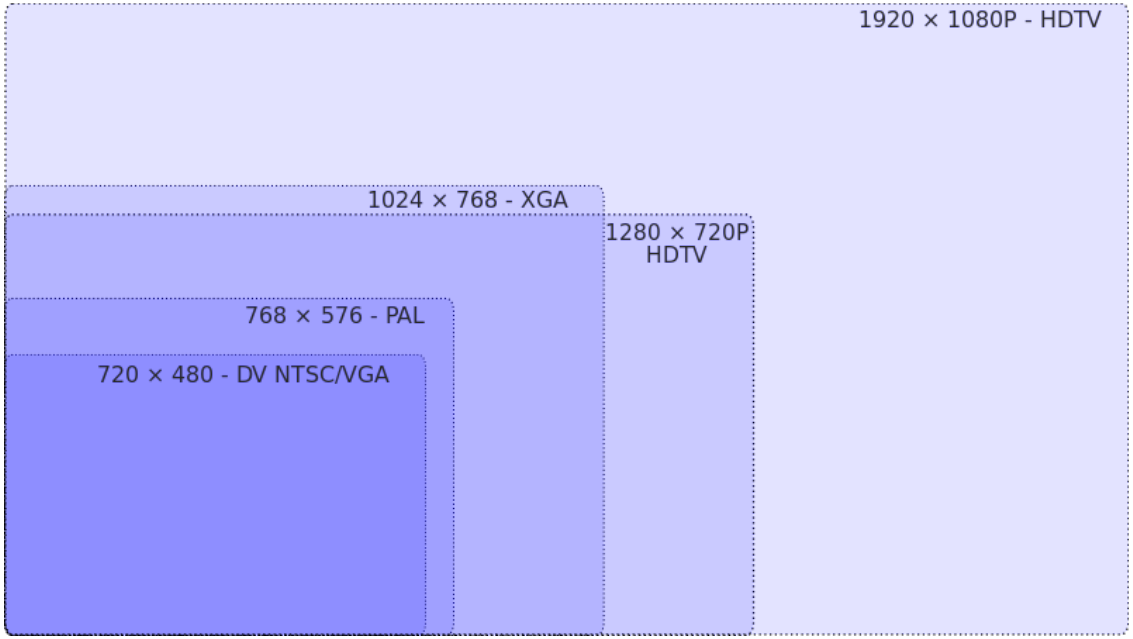
- With HDTV, the foundation of how frames are displayed still have their roots in the original NTSC and PAL analog video formats:
  - Using NTSC as a foundation for HDTV, a unique high definition frame is displayed every 30th of a second.
  - Using PAL as a foundation for HDTV, a unique high definition frame is displayed every 25th of a second.



- HDTV broadcast systems are identified with three major parameters:
  - Frame size: defined as *number of horizontal pixels × number of vertical pixels*.
  - Scanning system: both progressive and interlaced pictures are supported. It is identified with the letter *p* for progressive scanning or *i* for interlaced
  - Frame rate: identified as number of video frames per second or number of fields per second (for interlaced systems)

- Today HDTV includes different frame sizes:
  - 720p (HD ready) 921.600 pixel (1280×720) with progressive scan, (720 lines per scan)
  - 1080i 2.073.600 pixel (1920x1080) with interlaced scan (540 lines per scan)
  - 1080*p* 2.073.600 pixel (1920x1080) with progressive scan (1080 lines per scan)

Comparison of formats





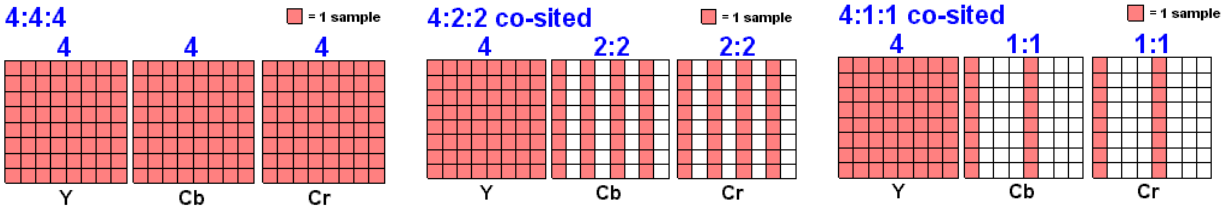
Video sampling

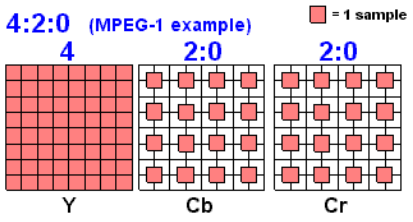
- 4:4:4 (Cb/Cr Same as Luma) Cb and Cr are sampled at the same full rate as the luma. MPEG-2 supports 4:4:4 coding. When video is converted from one color space to another, it is often resampled to 4:4:4 first.

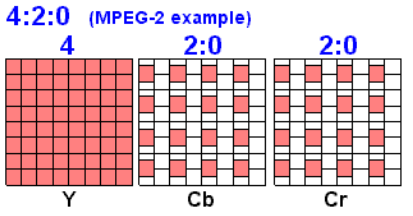
- 4:2:2 (1/2 the Luma Samples) Cb and Cr are sampled at half the horizontal resolution of Y. Co-sited means that Cb/Cr samples are taken at the same time as Y. It is considered very high quality and used for professional digital video recording, including DV, Digital Betacam and DVCPRO 50 . It is an option in MPEG-2.

- 4:1:1 (1/4 the Luma Samples) Cb and Cr are sampled at one quarter the horizontal resolution. Co-sited means that Cb/Cr samples are taken at the same time as the Y. It is used in DV, DVCAM and DVCPRO formats.

- 4:2:0 (1/4 the Luma Samples) The zero in 4:2:0 means that Cb and Cr are sampled at half the 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.





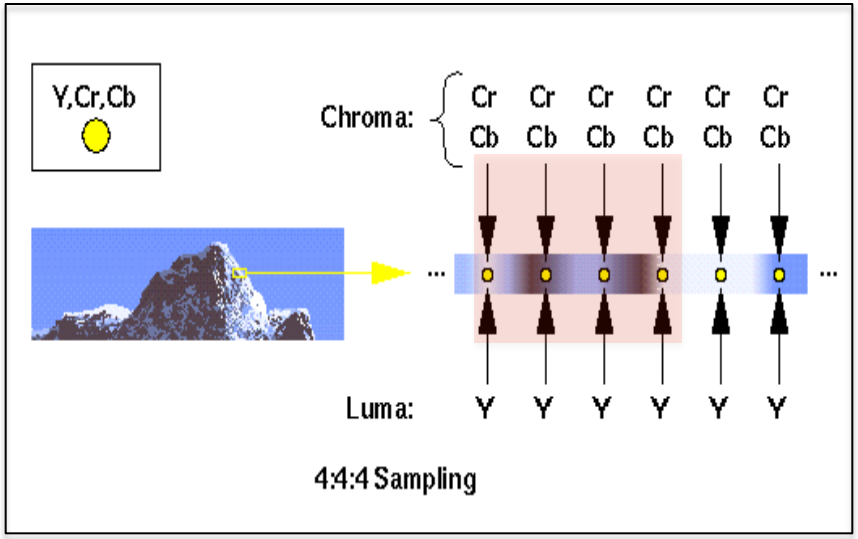


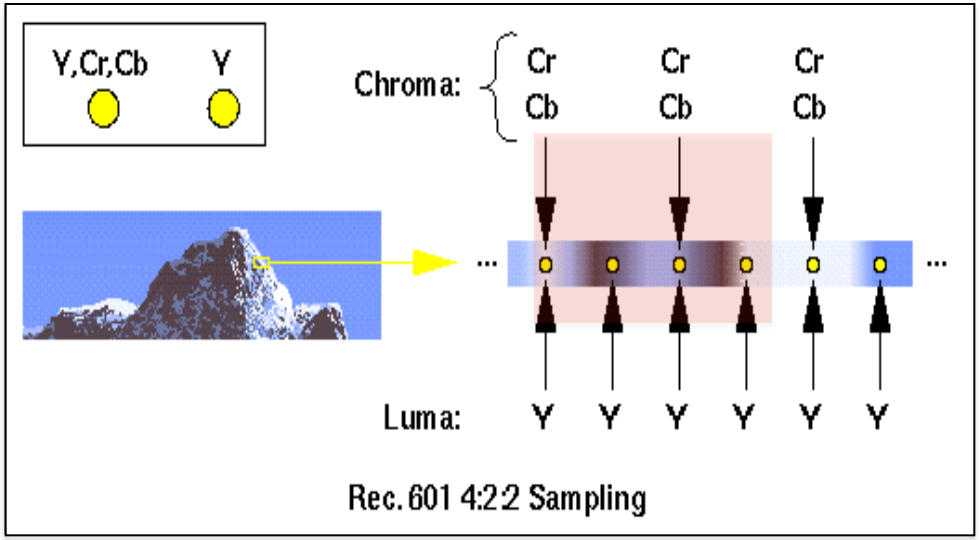
Digital video formats: ITU-R BT.601

- 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

4:2:2 is used in: D1, Digital Betacam, DVCPRO 50

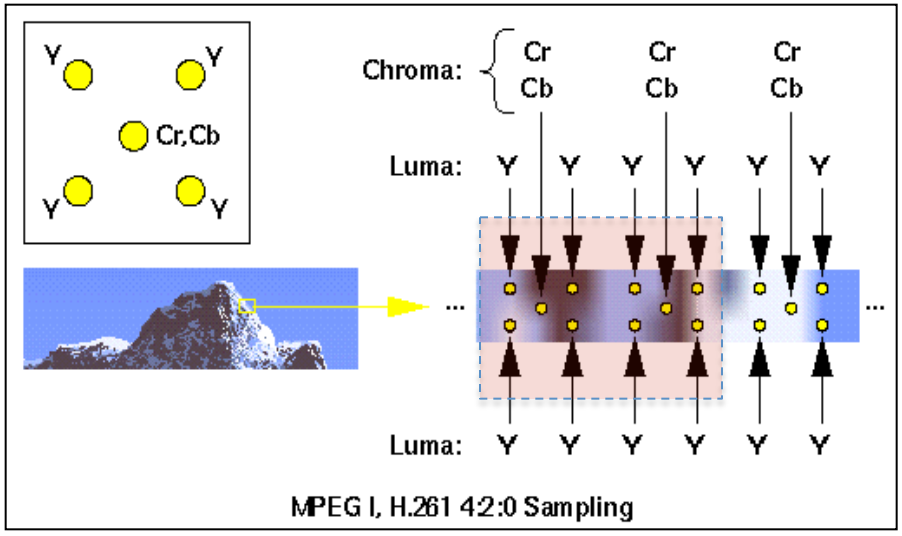


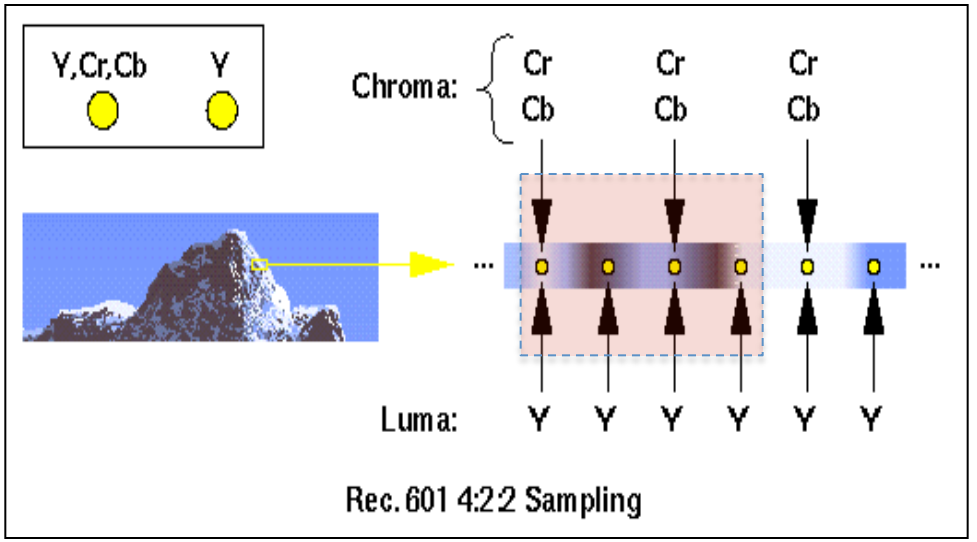


Digital video formats: MPEG 1

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

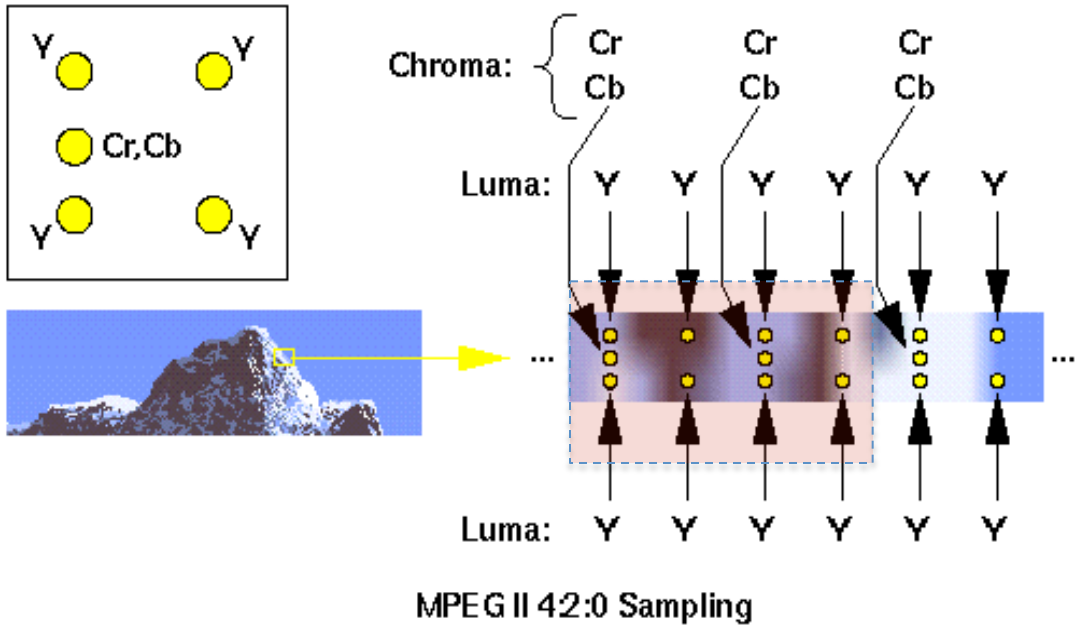
- In MPEG1 lines are dropped so to make data divided by 8 and 16.
  In comparison with CCIR 601 NTSC 4:2:2 sampling: 2:1 in horizontal luminance; 2:1 in time; 2:1 in vertical chrominance.





Digital video formats: MPEG 2

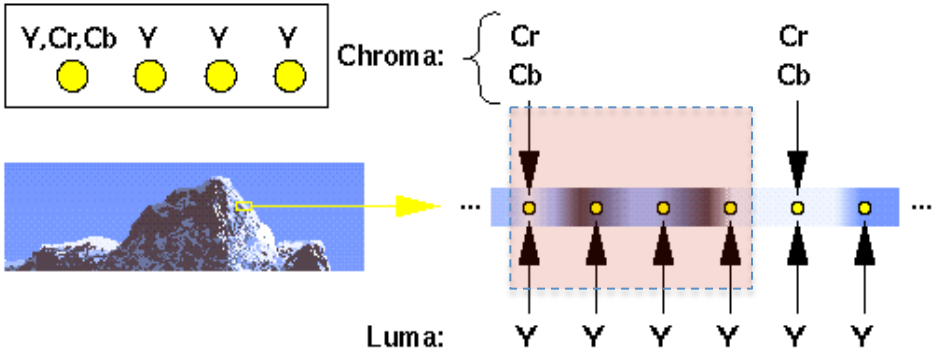
- MPEG2 bitrate 4 Mbit/s. MPEG2 was defined to provide a better resolution than MPEG1 and manage interlaced data. Based on fields instead of frames. Used for DVD and HDTV:
  - Frame sixe: 720x480
  - 4:2:0 sampling



MPEG II 42:0 Sampling

Digital video formats: 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)
  - Digital connectivity follows IEEE 1394 (“Firewire” or “i.Link” Sony).



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

- Horizontal resolution for luminance is 550 for DV.
- Horizontal resolution for chroma is about 150 lines (about ¼)

- DV25 has 25 Mb/sec data rate. Audio is not compressed with data rate equal to 3.5 Mb/sec. 1 Hour of DV25 requires approx 13 GB
- DV50 has 50 Mb/sec data rate
- DV100 is used for HDTV.

- The audio, video, and metadata are packaged into 80-byte Digital Interface Format (DIF) blocks. DIF blocks are the basic units of DV streams and can be stored as files in raw form or wrapped in file formats as AVI and QuickTime.







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)

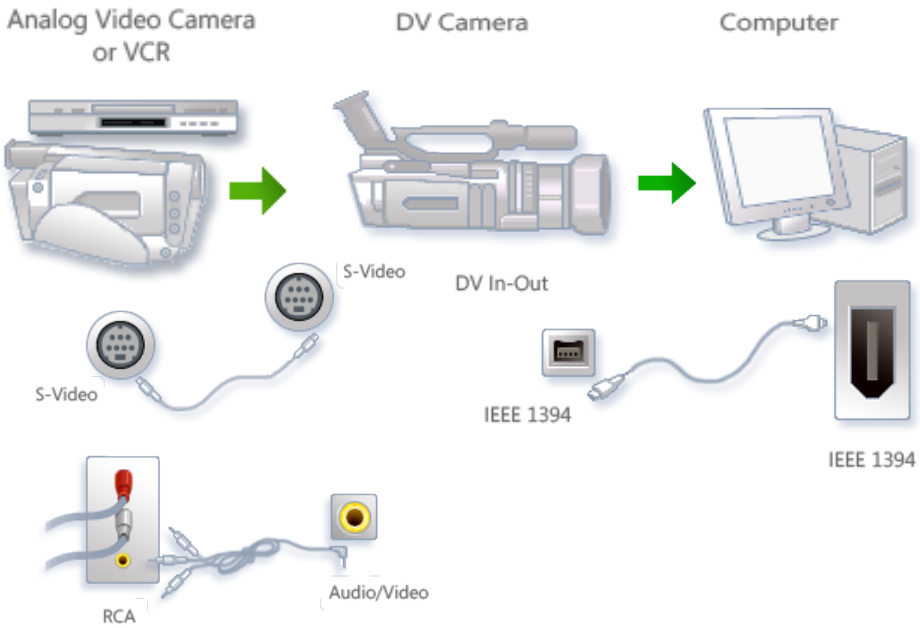
From analog to digital: fields

- Computers use *frames instead of fields* (all the lines are sent together) and video formats for computer are not interlaced (*noninterlaced* or *progressive scan*). This can create problems when transferring analog video to computers as in figure.
- Software tools are needed to reconstruct the full frame.





From analog camera to computer



- Many cameras both have analog (S-VHS or RCA) and digital (DV) connection. To connect a analog camera film to a computer you need:
  - A DV camera that supports DV pass-through
  - An IEEE 1394 cable (FireWire cable)
  - An IEEE 1394 port on your computer
  - An Audio/Video (A/V) cable
  - An S-Video cable

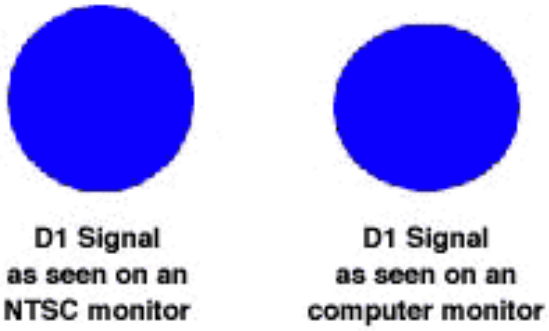
- With Windows Vista import video using Windows Import Video
- With Mac, Mac should automatically launch iMovie.

Frame aspect ratio

- Aspect ratio: is the ratio between image width and image height
  - PAL and NTSC aspect ratio: 4:3 (1.33)
  - HDTV Panorama format: 16:9 (1.77)
  - Film USA: 1.85
  - Film Europe: 1.66

Pixel aspect ratio

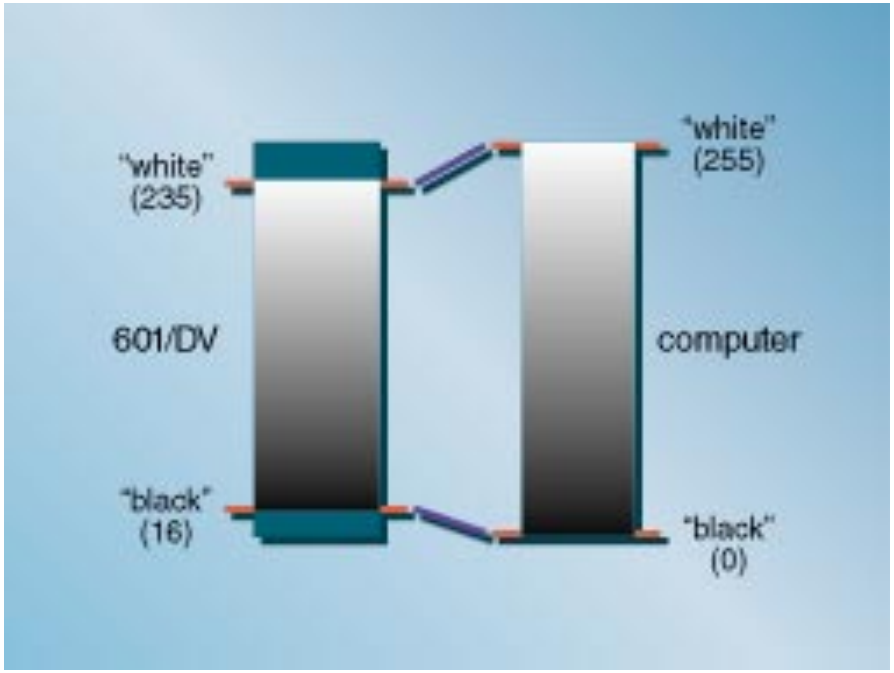
- Pixel aspect ratio = pixel height/width
- Some video formats have non square pixels (e.g. 10:11 for digital video D1 - CCIR 601); computer monitors use square pixels instead.
- According to this, a circle in a computer screen will appear as an ellipse on a TV screen. Therefore a video in D1 - CCIR 601 format must be converted in order to be displayed correctly on a TV screen. Similarly a computer graphics video must be converted to be displayed in a video format. Software like After Effect provide such support.



Black level

- for digital video D1 - CCIR 601 defines *black level* (the ultimate black) at value 16.
  In computer graphics *black level* has value 0.

- When images are transferred from computer graphics to video or viceversa, the resulting image will appear darker or lighter if this difference has not been accounted correctly.



Video compression methods

Video compression

- Due to the large amount of data included in a video stream, compression algorithms have main relevance for video. Video compression algorithms can be lossy and lossles but typically are lossy, starting with color subsampling.

- Compression can be spatial or/and temporal:
  - remove spatially redundant data (as in JPEG)
  - remove temporally redundant data (the basis for good video compression)

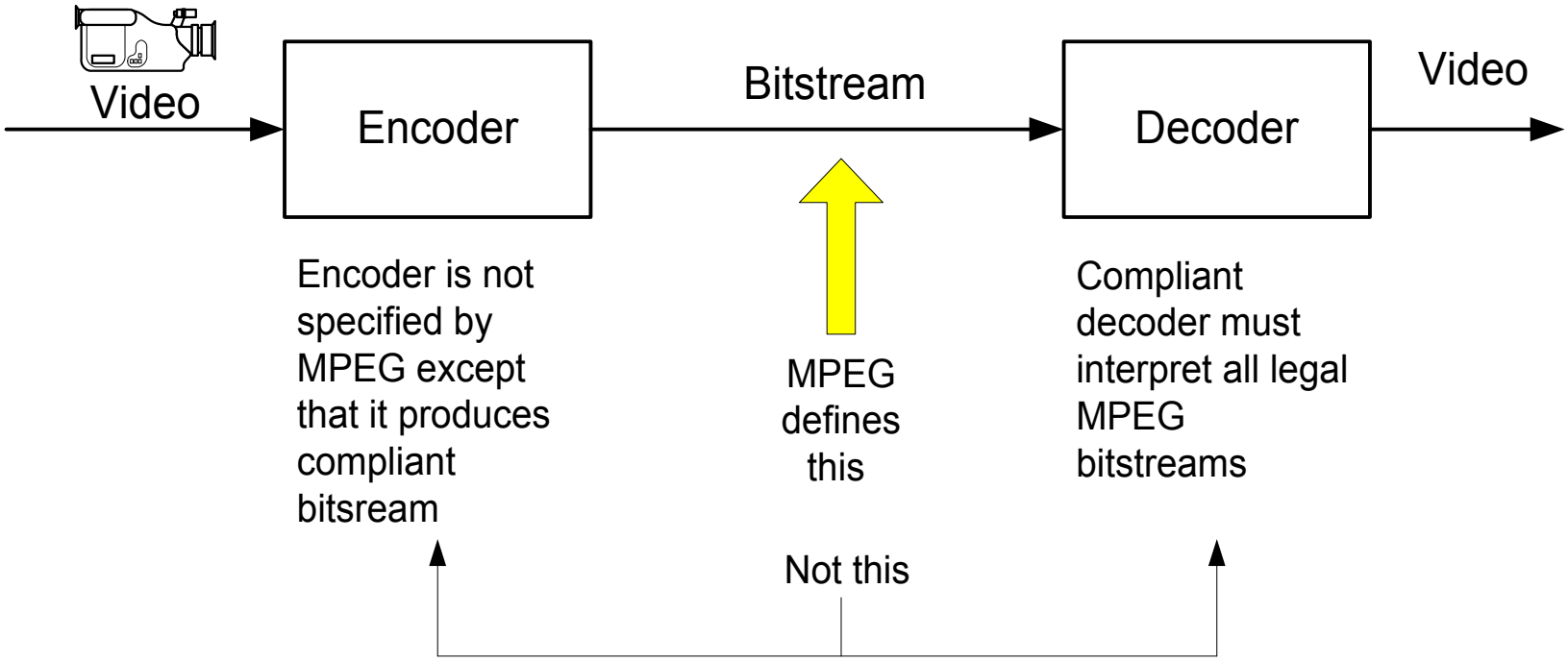
MPEG

- MPEG is a lossy compression method for video developed by the Moving Picture Expert Group defined according to ISO standard. It 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 MPEG are:
  - Transform-domain-based compression i.e *intra-frame coding* (similar to JPEG with 2D DCT, quantization and run-length encoding)
  - Block-based motion compensation (similar blocks of pixels common to two or more successive frames are replaced by a pointer i.e. a *motion vector* that references one of the blocks). Predictive Encoding is done with reference to an anchor frame according to interpolative techniques, i.e. *Inter-frame coding*.

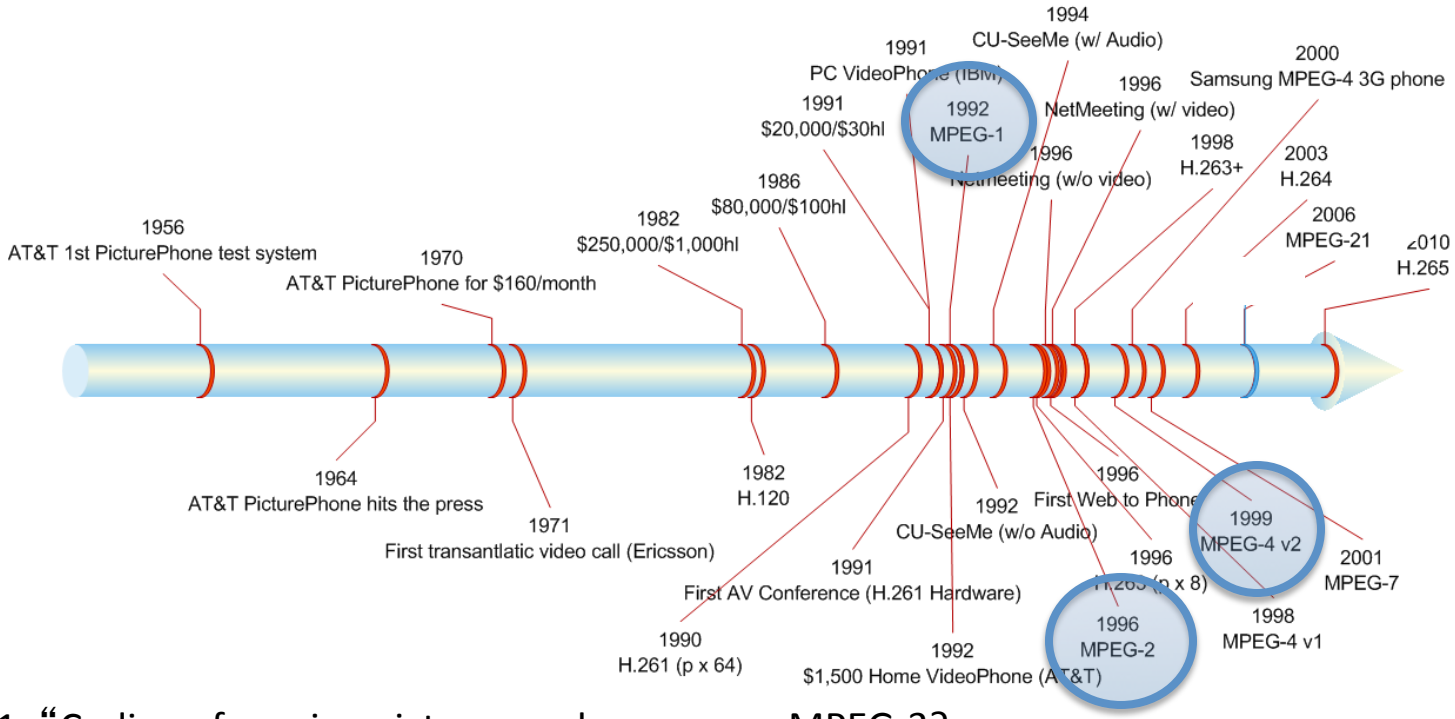
- MPEG distinguishes: MPEG1, MPEG2, MPEG4.

What MPEG defines



- 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 standards (1990-2010)



- 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

Video file 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.

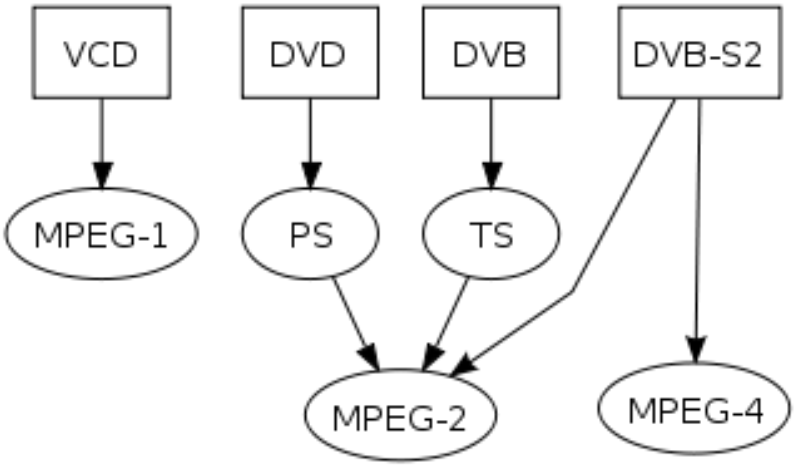
- Most common video formats:
  - Apple Quicktime (multiplatform) .mov
  - Microsoft AVI .avi
  - Windows Media Video .wmv
  - MPEG (multiplatform) .mpg o .mpeg

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

MPEG1, MPEG2 file formats

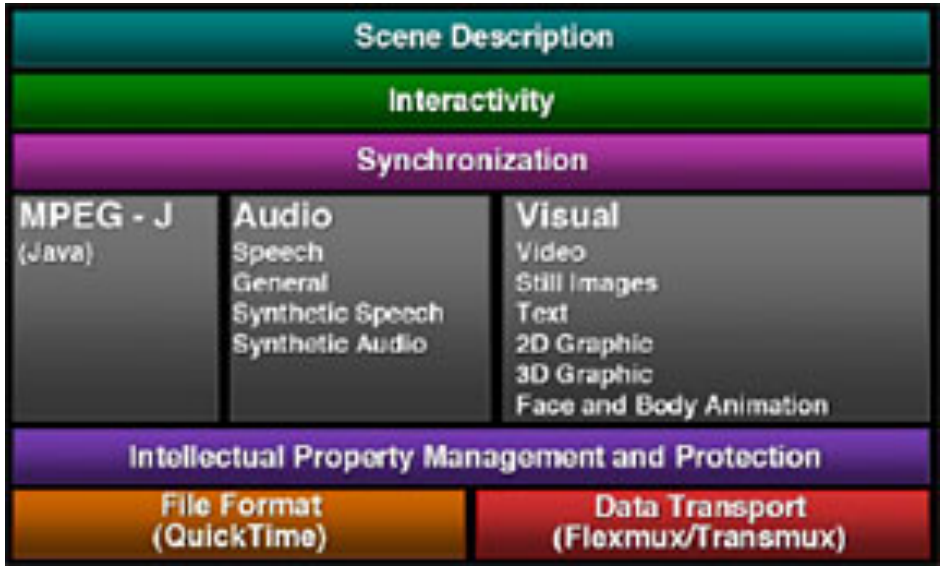
- MPEG1 and MPEG2 compression standards have defined the Program Stream (PS).
  MPEG-PS is a container format for multiplexing digital audio, video. It was designed for reliable media, such as disks (like DVDs).

- MPEG2 has defined the Transport Stream (TS).
  MPEG-TS is a standard format for transmission and storage of audio, video, and data, and is used in broadcast systems such as DVB and ATSC. MPEG-TS specifies a container format encapsulating packetized elementary streams, with error correction and stream synchronization features for maintaining transmission integrity when the signal is degraded.



MPEG 4 file format

- MPEG4 file format was inspired by the QuickTime format, and may contain different streams and media. Can contain metadata.
  - Audio-only MPEG-4 files generally have extension .m4a.
  - MPEG4 files can be streamed or used for progressive download
  - Supports very low bit rates: ~ 64 Kb/sec
  - Mobile phones use 3GP, an implementation of MPEG-4 Part 12 (a.k.a MPEG-4/ JPEG2000 ISO Base Media file format), similar to MP4.



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; they are separated when decoding. 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.

CPB Constrained Parameters Bitstream

- MPEG1 can 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. MPEG1 specifications:

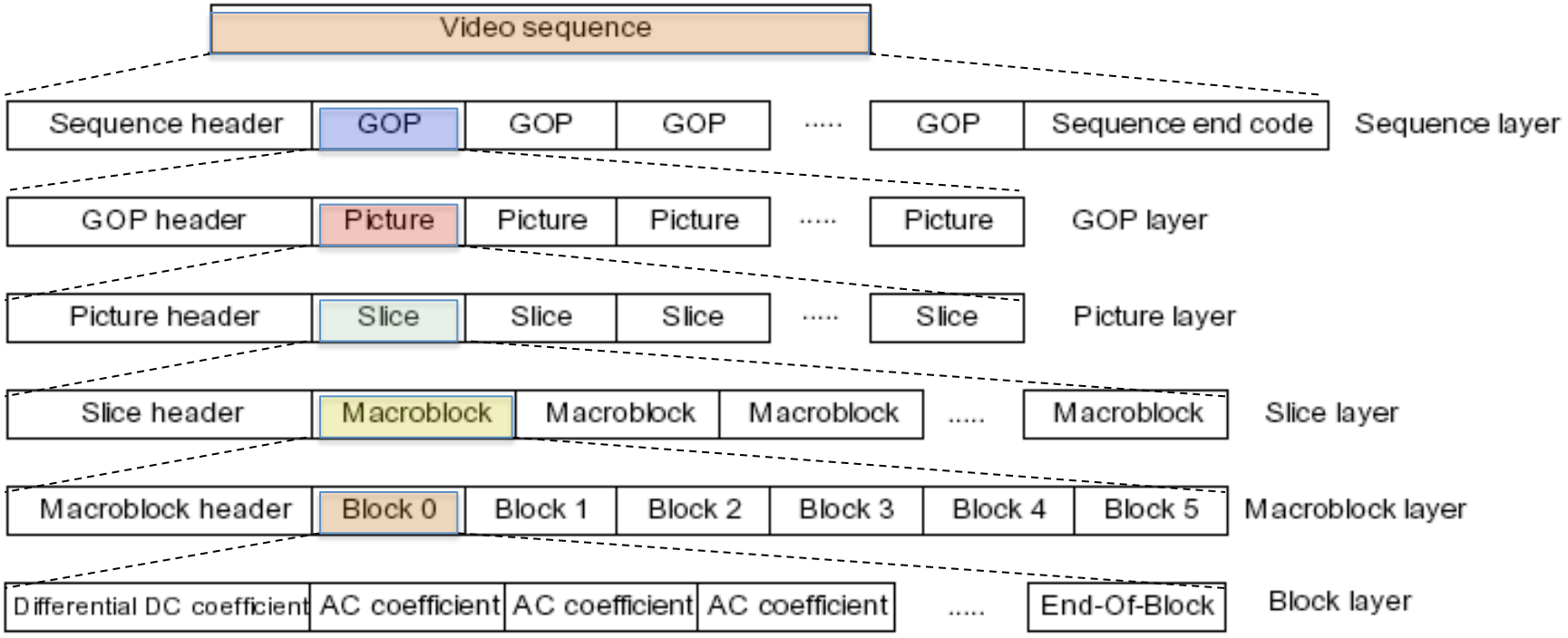
horizontal resolution	≤ 768 samples
vertical resolution	≤ 576 scan lines
picture area	≤ 396 macroblocks
pel rate	≤ 396 × 25 macroblocks per second
picture rate	≤ 30 frames per second
bit rate	≤ 1.856 Mbps

One macroblock is composed by 16x16 pixel (396 macroblocks = 101.376 pixel)

- However the usual MPEG1 video resolution is 352x240 (30 fps) or 320x240 (24 fps) at a bitrate of ~1.5 Mbps. This modality is also 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.

6 layers

- The MPEG1 video stream comprises 6 distinct 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


Video sequence
Sequence header
GOP
GOP
GOP
.....
GOP
Sequence end code
Sequence layer
GOP header
Picture
Picture
Picture
.....
Picture
GOP layer
Picture header
Slice
Slice
Slice
.....
Slice
Picture layer
Slice header
Macroblock
Macroblock
Macroblock
.....
Macroblock
Slice layer
Macroblock header
Block 0
Block 1
Block 2
Block 3
Block 4
Block 5
Macroblock layer
Differential DC coefficient
AC coefficient
AC coefficient
AC coefficient
.....
End-Of-Block
Block layer

GOP

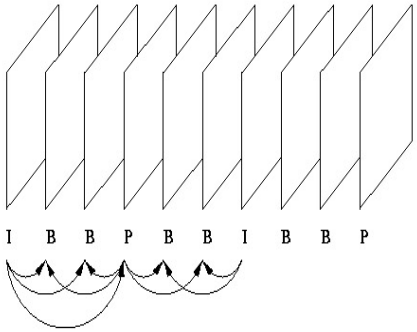
- A video sequence is decomposed in Groups of Pictures (GOPs).
  Within GOPs, frames have different typology:
  - I (intra-coded),
  - P (Predictive),
  - B (Bi-directional),
  - D (DC) frame.

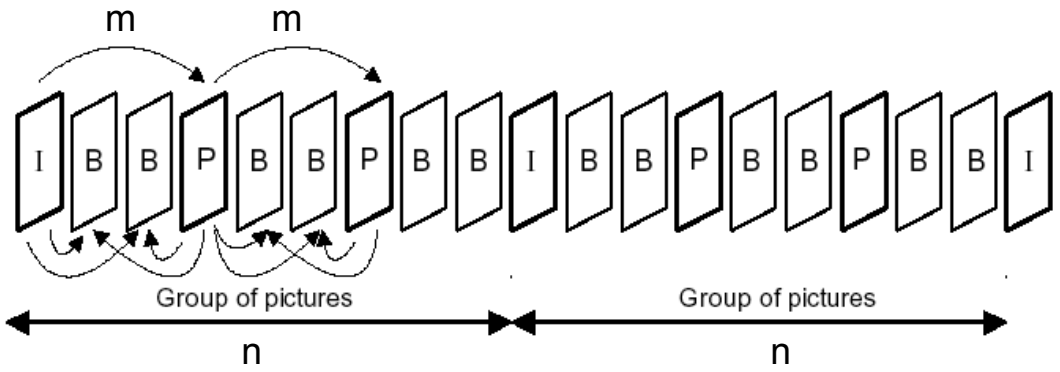
  Distance between I, P e B frames can be defined when coding.

  The smaller GOP is the better is fidelity to motion and the smaller compression.

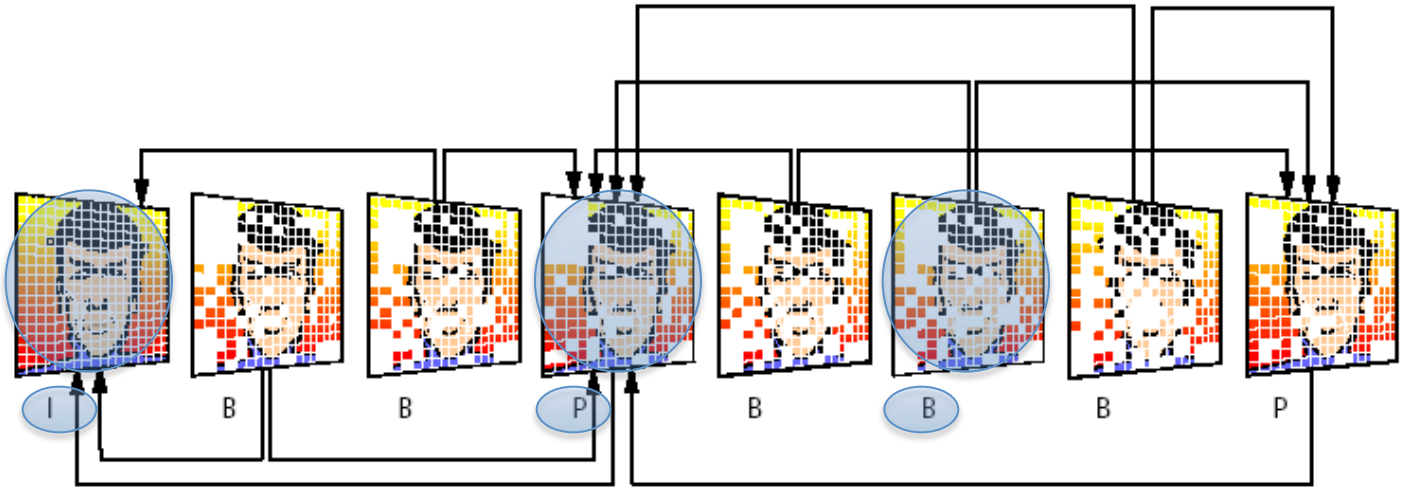
- A GOP is *closed* if can be decoded without information from frames of the preceding GOP (ends with I,P or B with past prediction). Max GOP lenght are 14-17

Typically m=3, n=9:





Frames



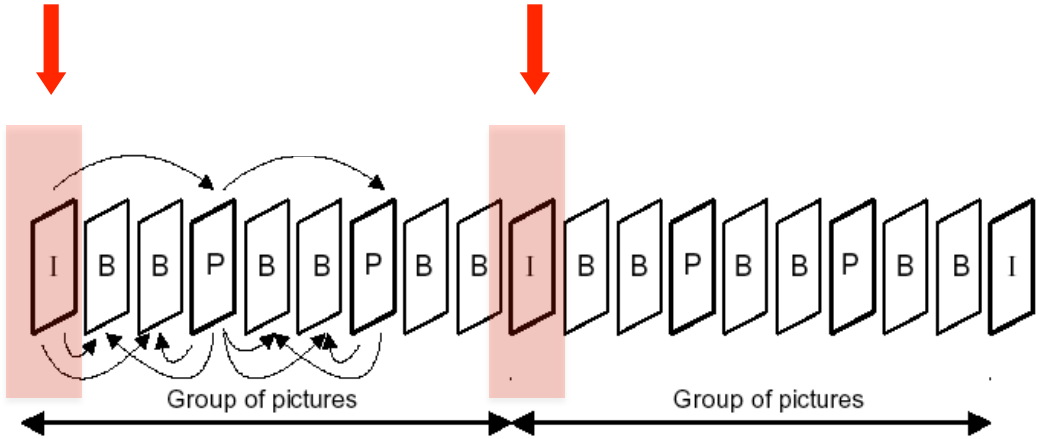
- I-frames contain the full image and are JPEG compressed
- P-frames are based on preceding I o P-frame according to Motion prediction
- B-frames use past or future I o P frames according to Motion prediction
- D frames are similar to I frames but are only DC encoded (no AC coefficients). They are low quality representations used as thumbnails in video summaries

- Frame types: I, P, B occur in repetitive patterns within a GOP; there are predictive relationships between I, P and B frames.

I-frames

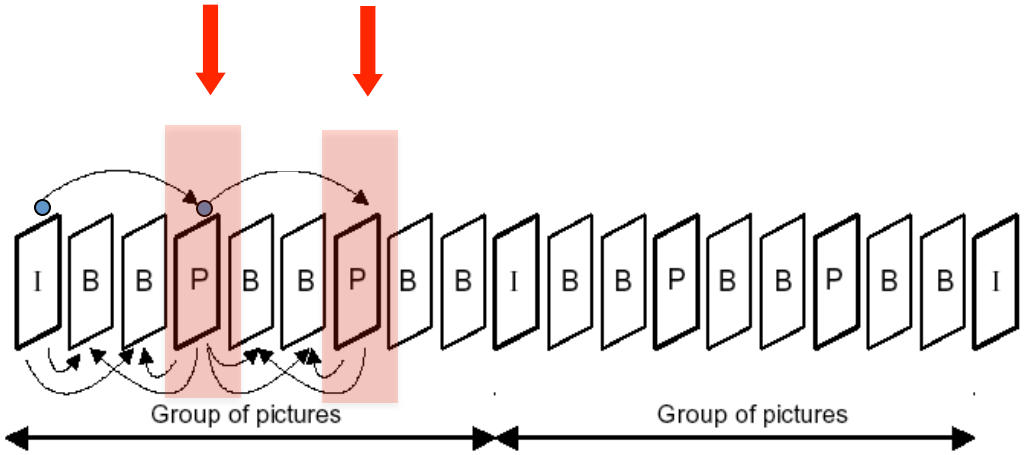
- *Intra – coded frames* are so called because they are decoded independently from any other frames. They are identical to JPEG frames.

- Intra-Coded frame are coded with no reference to other frames (*anchor*). Minimize propagation of errors and permit random access. I-frame compression is very fast but produces large files (three times larger than normally encoded MPEG video)



P-frames

- *Predictive-Coded frame* 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 immeditely preceding it. The difference is calculated using *motion vectors*.



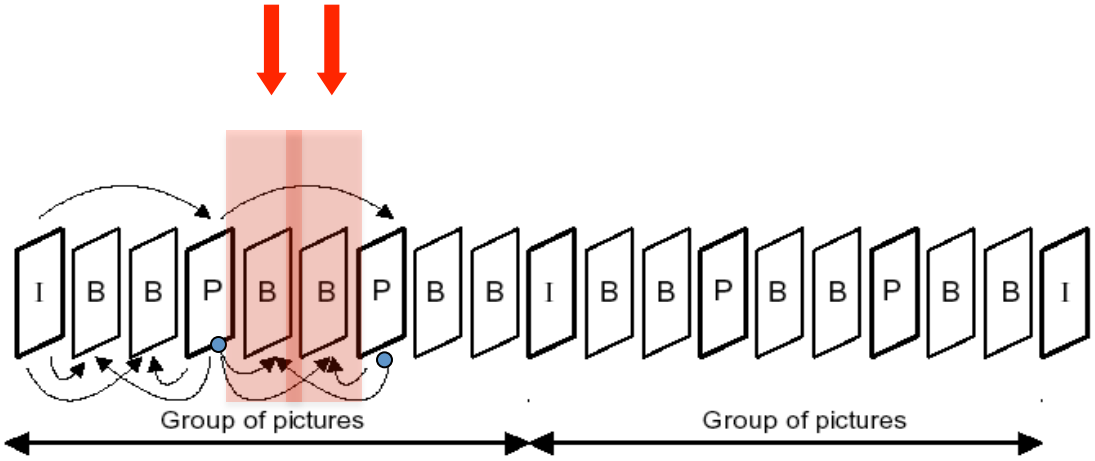
- In the case in which an object moves in front of a fixed background, P-frames can code the object but can't code the revealed background as well. Answer is use of B-frames



B-frames

- *Bi-directional-Coded frame* 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. 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.

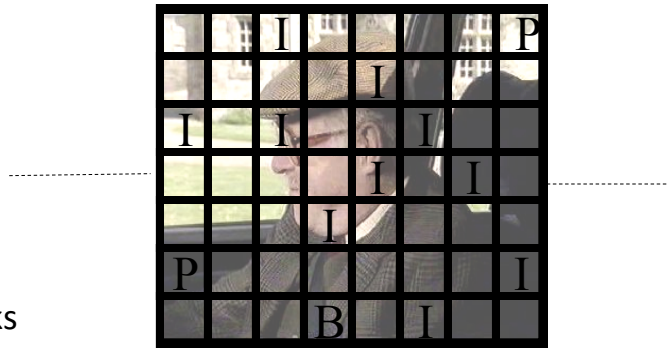


- Relative number of (I), (P), and (B) pictures can be arbitrary. It depends on the nature of the application. It may depend on fast access and compression ratio requirements:
  - relatively smaller amount of compression is expected to be achieved at (I) pictures compared to (P) and (B) pictures.
  - the (B) pictures are expected to provide relatively the largest amount of compression under favorable predict

Frames and macroblocks

- Each video frame contains *macroblocks* that is the smallest independent unit of video considered by MPEG. Macroblocks are set of 16x16 pixels and are necessary for purposes of the calculation of motion vectors and error blocks for motion compensation.

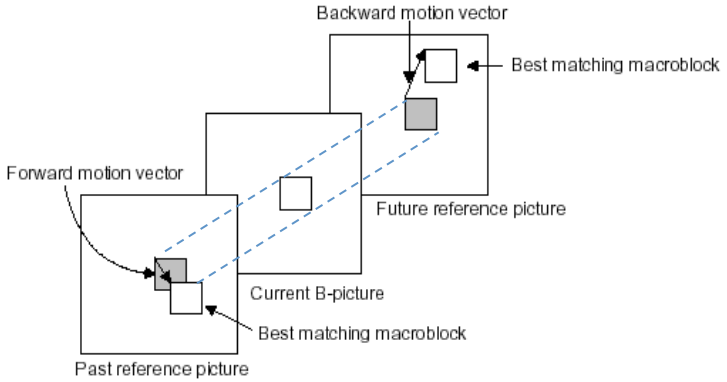
- I and D frames contain Intra-coded (I) macroblocks with direct encoding from the image samples
- P and B frames contain encoding of residual error after prediction:
  - P frames contain Intra-coded (I) macroblocks or forward-predicted (P) macroblocks
  - B frames contain Intra-coded (I), forward or/and backward-predicted (P or B) macroblocks

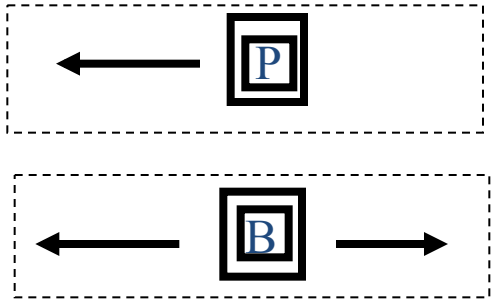


B Frame with macroblocks

Macroblocks

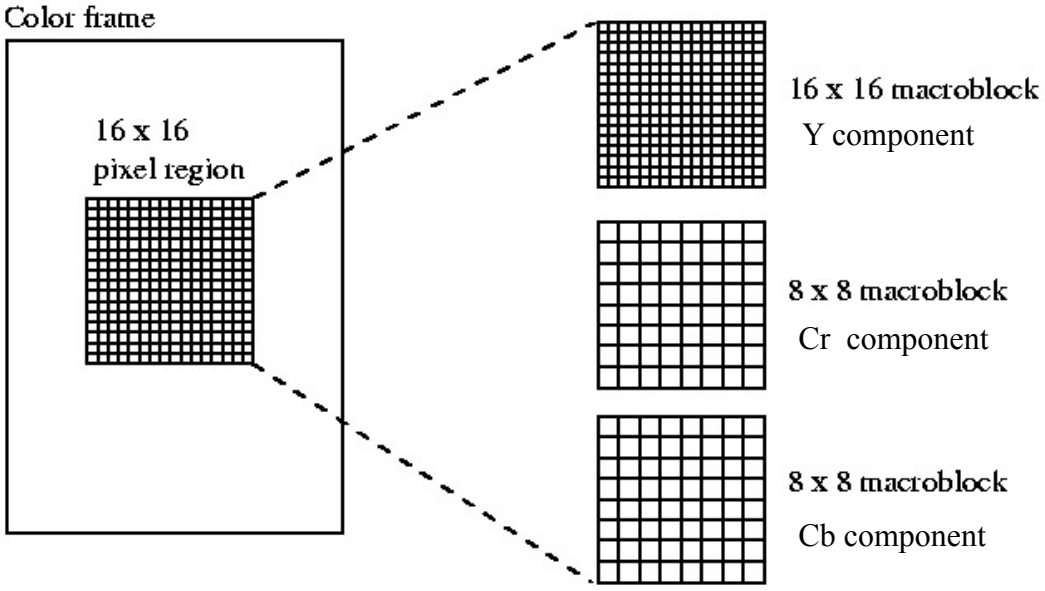
- Main types of macroblocks:
  - I encoded independently of other macroblocks (by 2D Discrete Cosine Transform as in JPEG blocks)
  - P encode not the region but the motion vector and error block of the previous frame (forward predicted macroblock)
  - B same as above except that the motion vector and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





Macroblock components

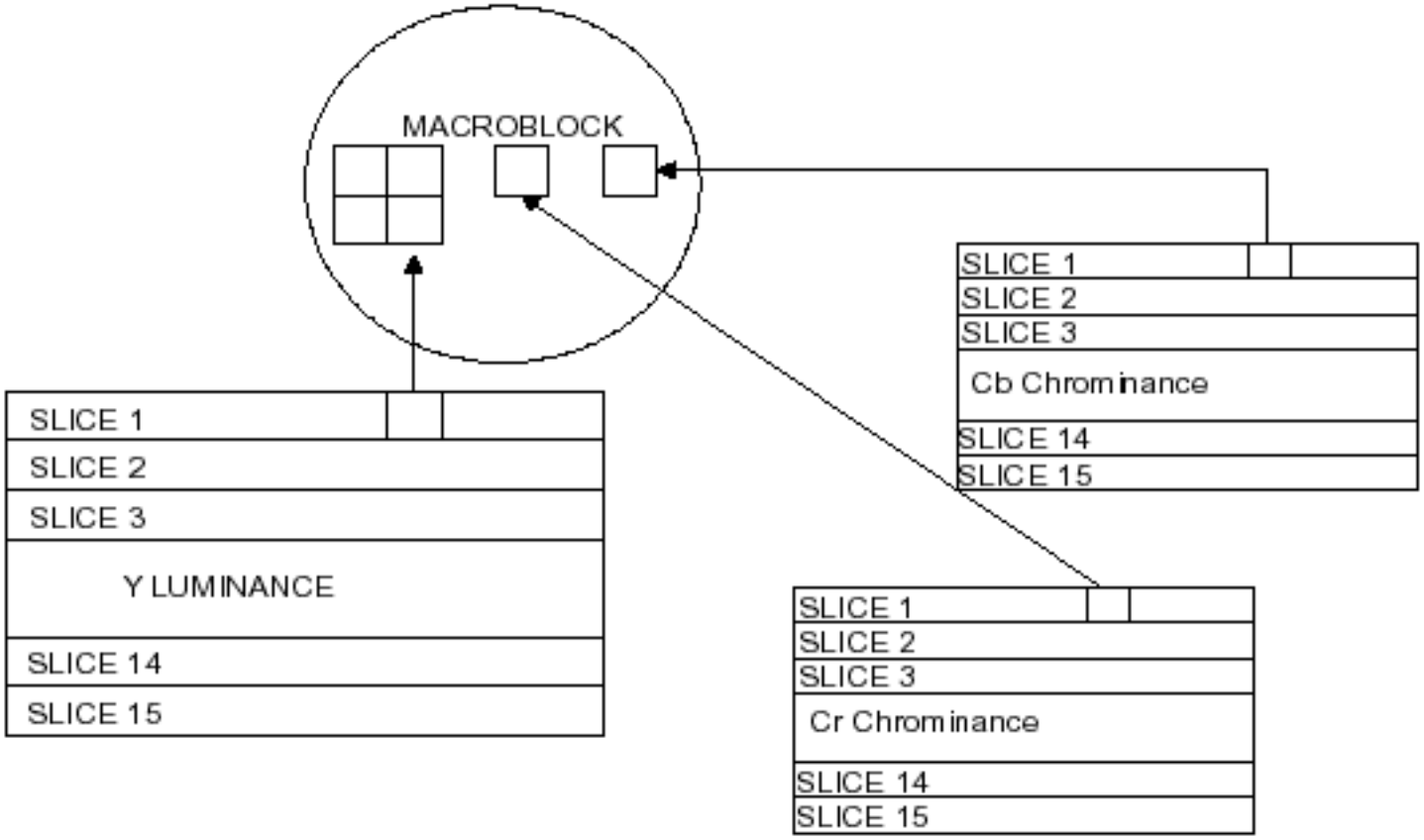
- Each macroblock is encoded separately.
- The component of a macroblock for motion compensation is luminance Y component. Cr and Cb are chrominance components.



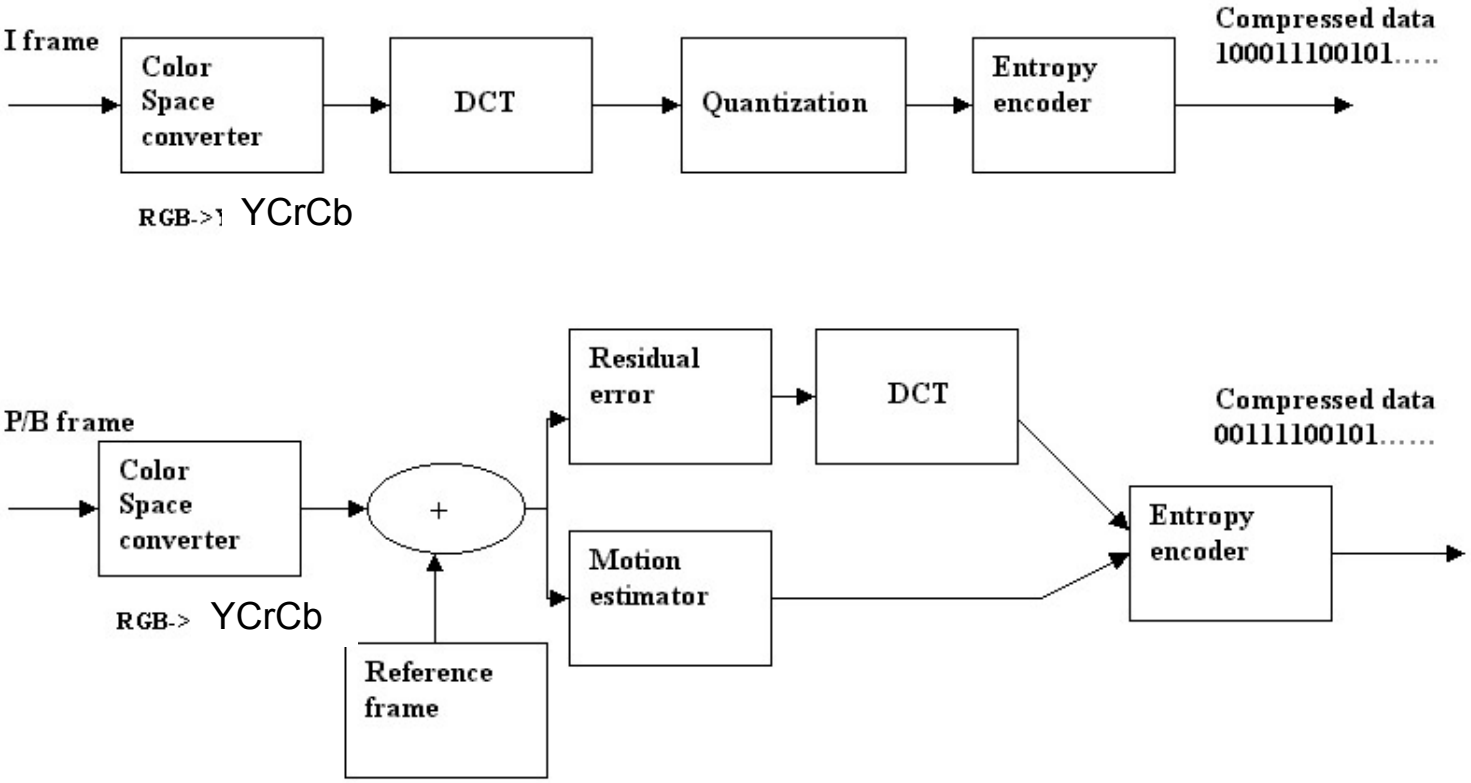
4:2:0 sampling

Slices

- Macroblocks are organized into slices

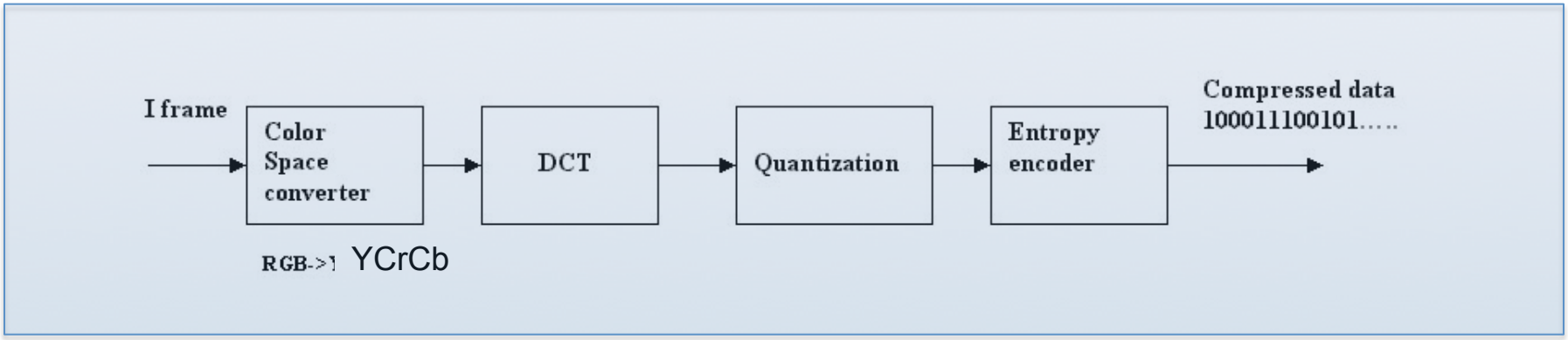


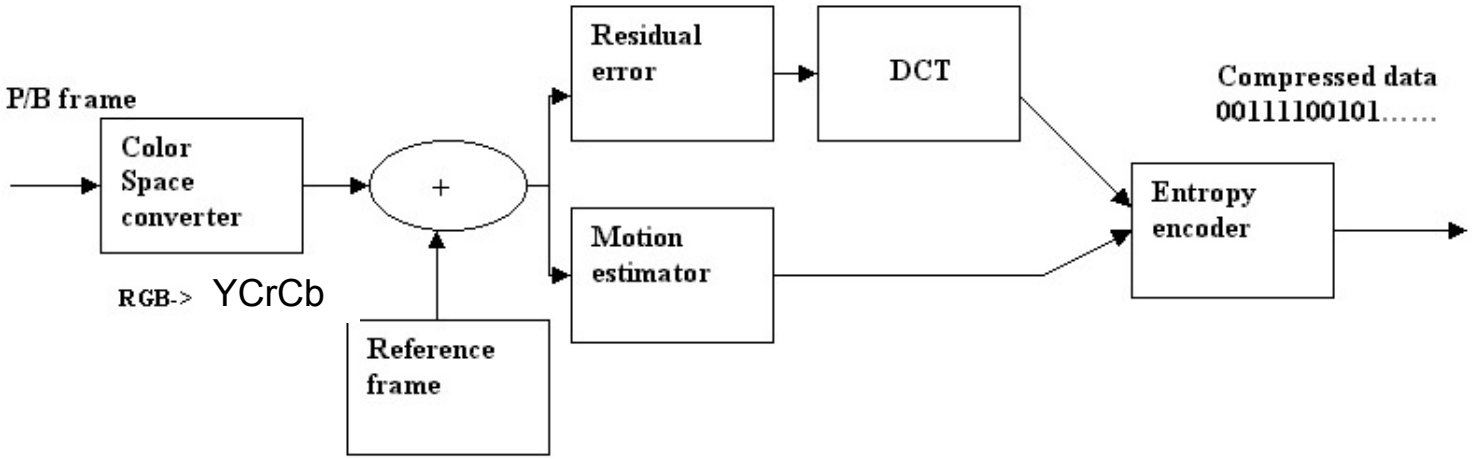
Encoding macroblocks



The block diagram of the MPEG encoder

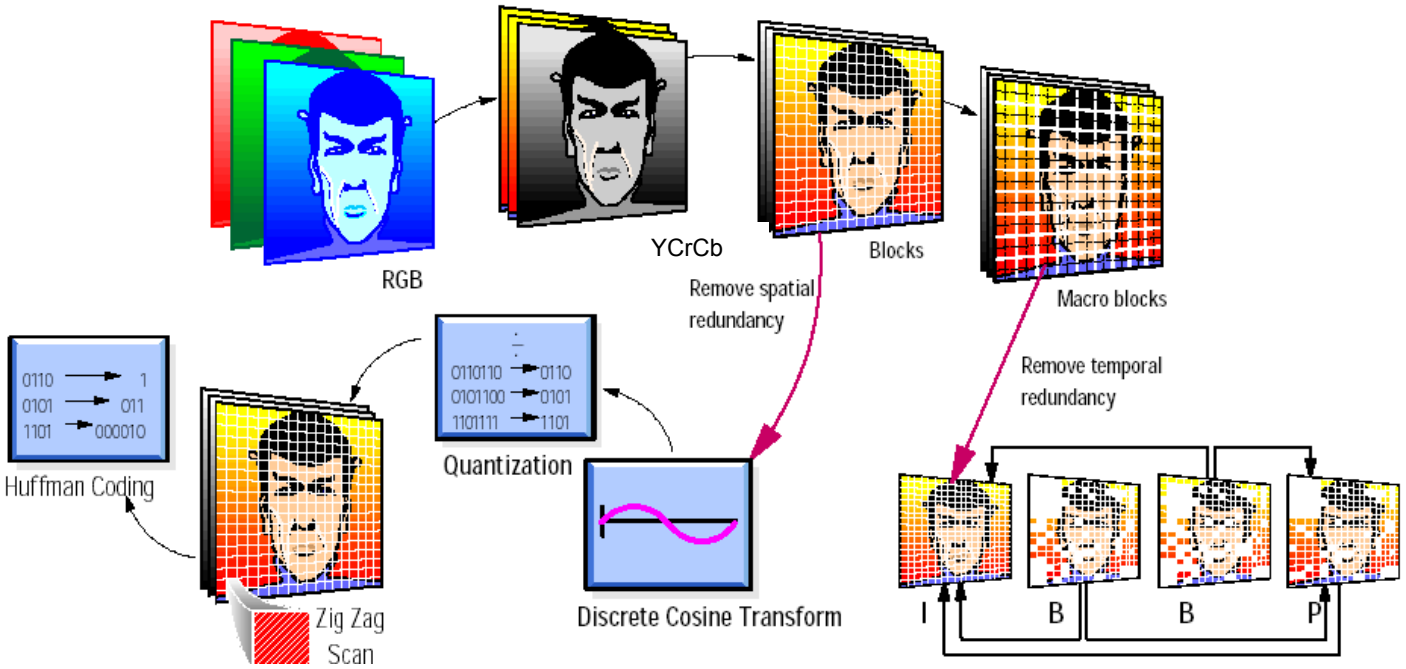
I-macroblock coding



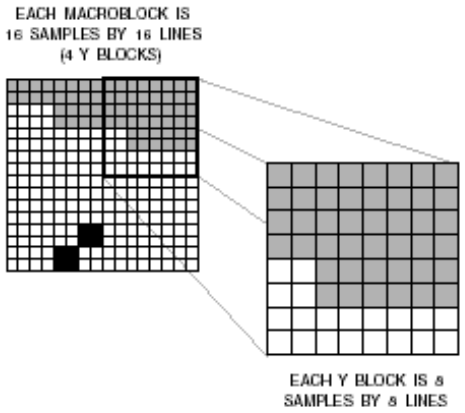


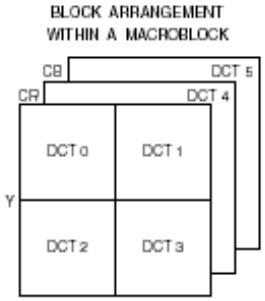
I-macroblock coding

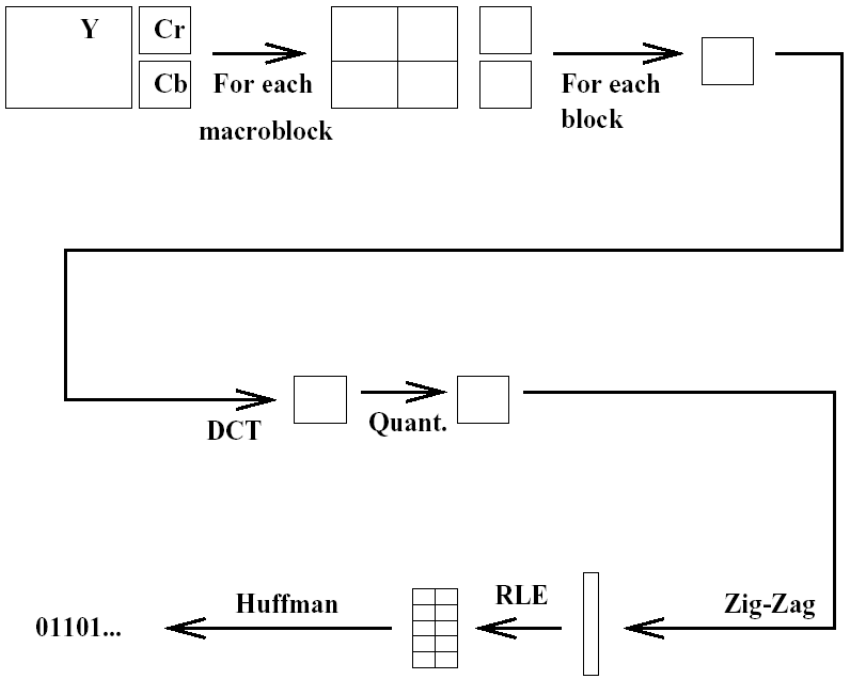
- I-macroblock coding is performed according to JPEG encoding



I-macroblock coding in more 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)

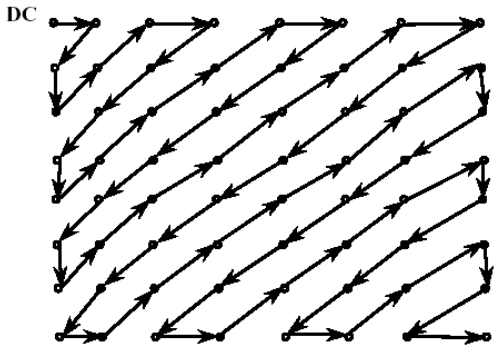
Spatially-Adaptive Quantization

- Spatially-adaptive quantization is made possible by the scale 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

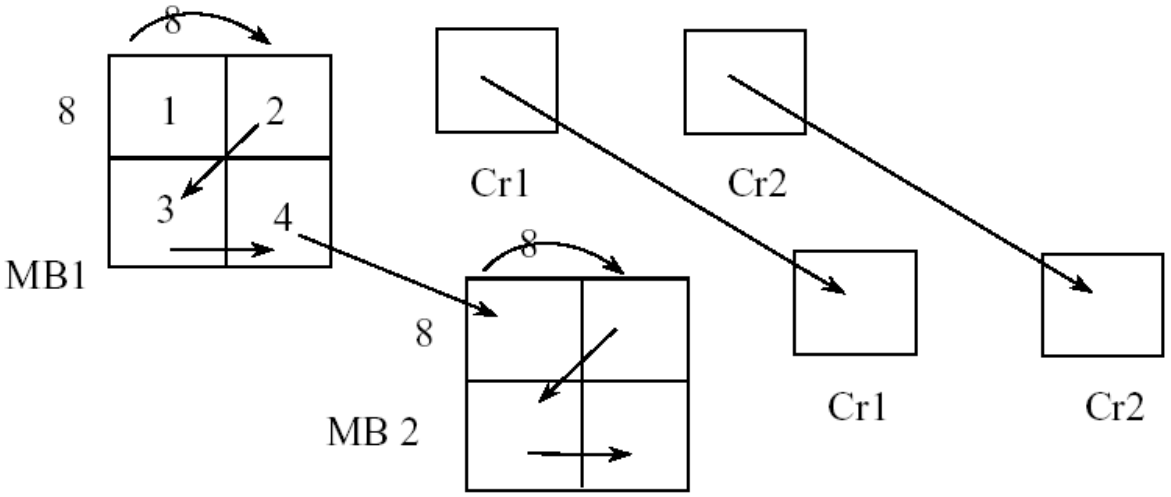
MPEG1 default quantization matrix

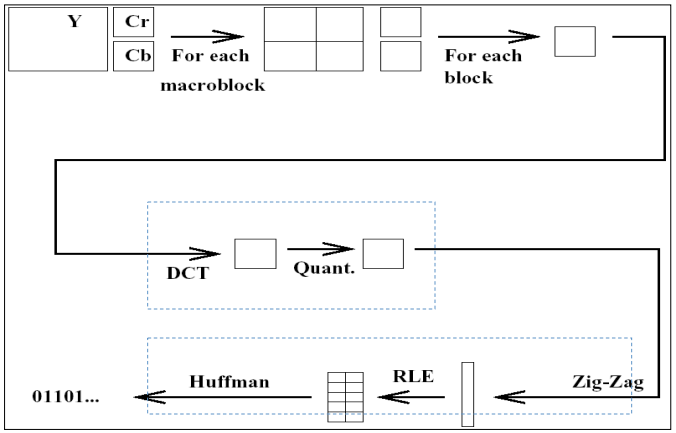
- zig-zag scanning is used to create a 1D stream

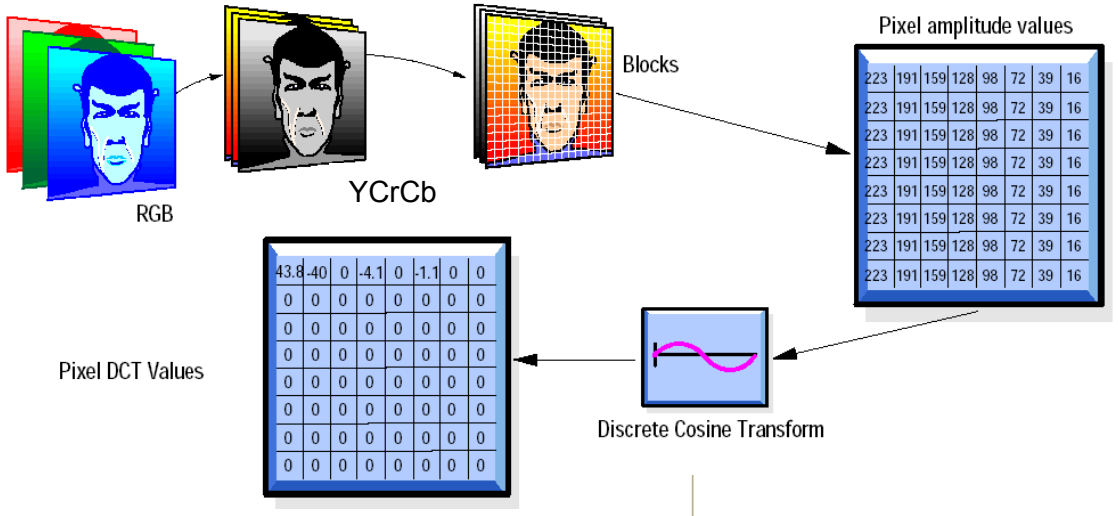


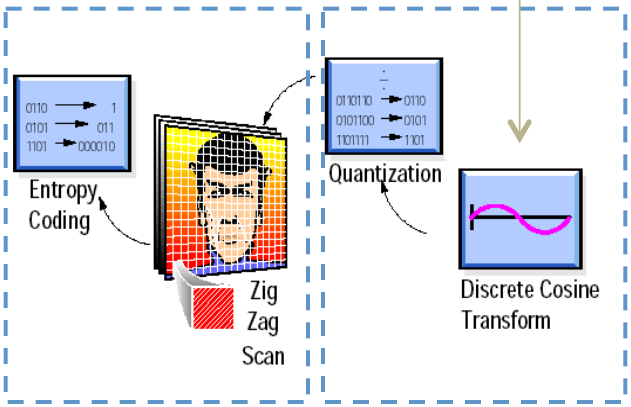
- AC coefficients are encoded losslessly according to *run length encoding* and Huffman coding (VLC: variable length coding). *Run length* and *level tables* are formed on a statistical *basis.* Different tables for Y and CbCr.

- DC coefficients encode differences between blocks of the macroblock (see below)

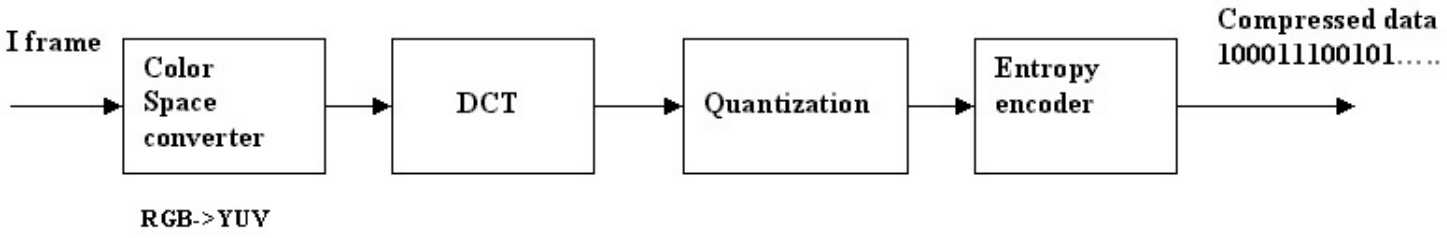


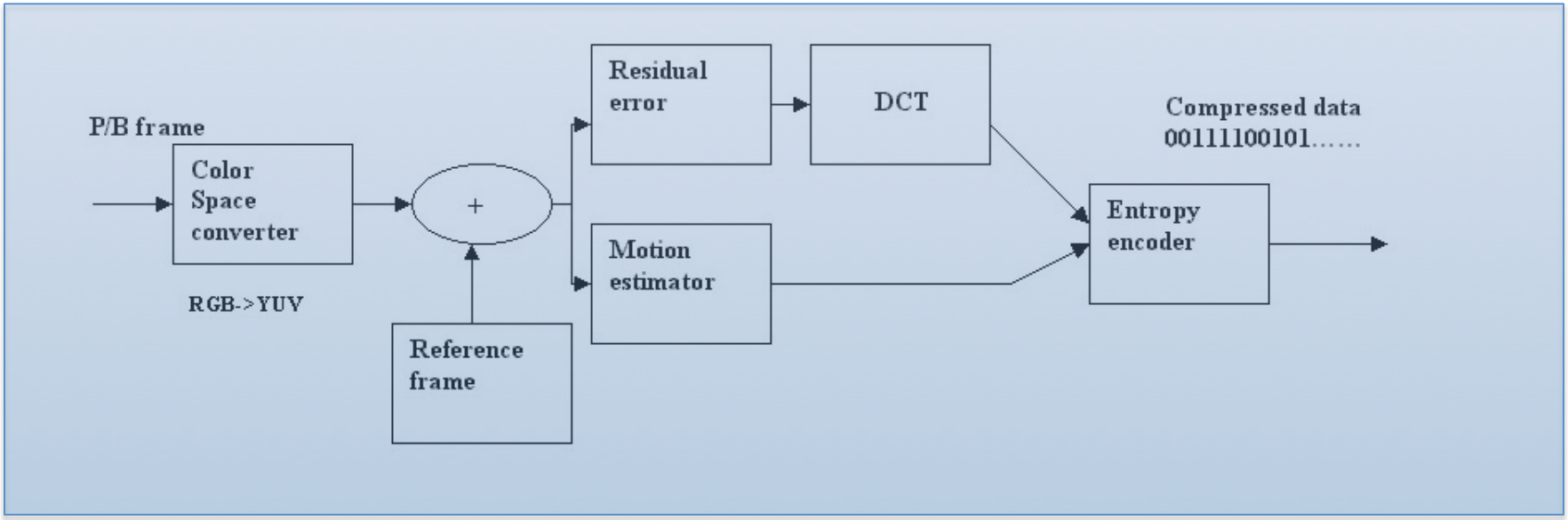

Y
Cr
Cb
For each
macroblock
For each
block
DCT
Quant.
Huffman
RLE
Zig-Zag
01101...


RGB
YCrCb
Blocks
Pixel amplitude values
223 191 159 128 98 72 39 16
Discrete Cosine Transform
Pixel DCT Values
43.8 -40 0 -4.1 0 -1.1 0 0


Entropy
Coding
Zig
Zag
Scan
Quantization
Discrete Cosine
Transform

P/B macroblock coding

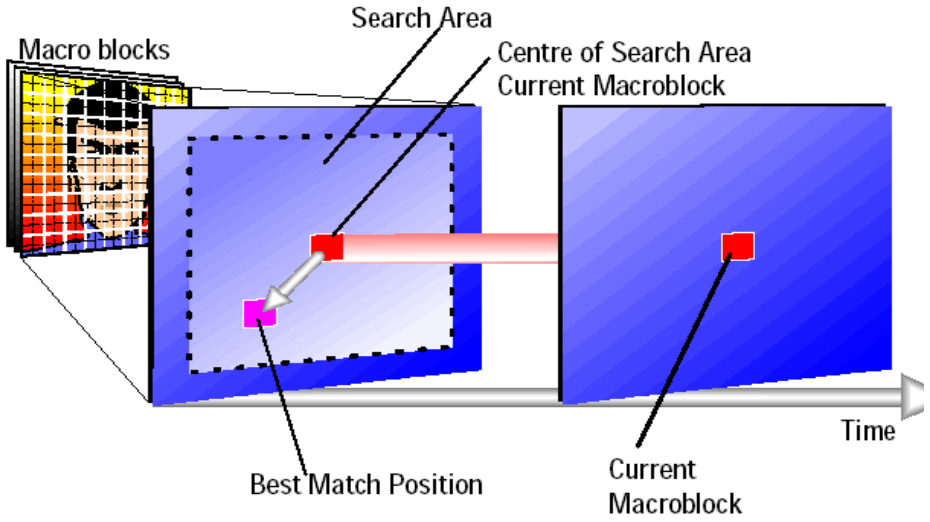




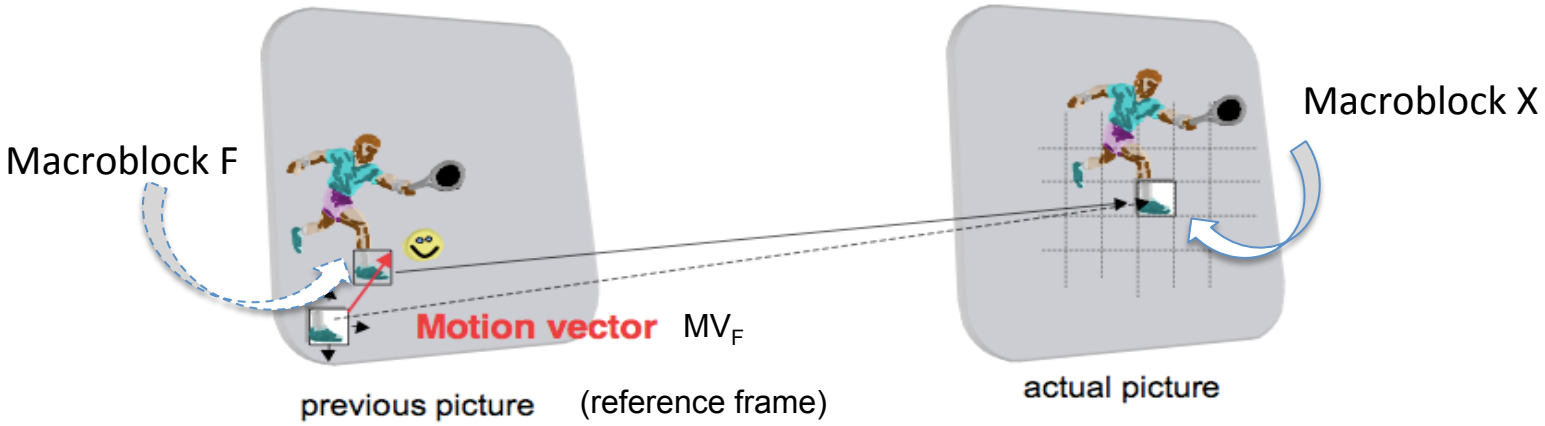
Block motion compensation

- P and B macroblock coding is based on block motion compensation. This is the process of replacing blocks with a *motion vector* and the *error block.* A motion vector describes the transformation between the same (similar) blocks in adjacent frames in a video sequence.

- 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 vector calculation



- Calculation of motion vectors is performed by matching similar blocks of pixels common to two or more successive frames and replacing them by a pointer. Motion vectors are assumed to be constant over a macroblock

- Instead of sending quantized DCT coefficients of macroblock X:
  - Finds the best-matching block in the reference frame, by searching an area in the reference frame and compare. Each block can be assigned a match from either a *backward* (B) or *forward* (F) reference
  - Sends quantized DCT coefficients of X-F (prediction error). If prediction is good, error will be near zero and will need few bits.
  - Encodes and sends the motion vector MV_F. This will be differentially coded with respect to its neighboring vector, and will code efficiently.

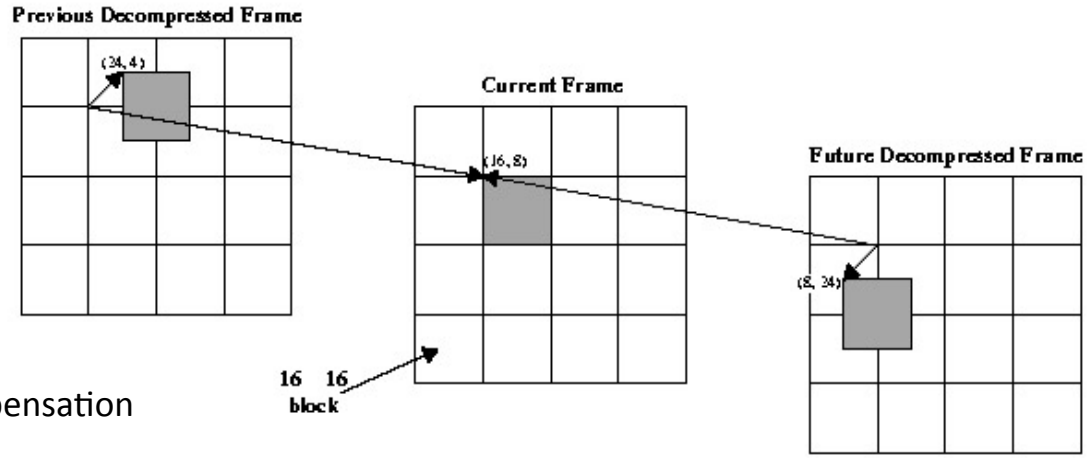
- With predictive video encoding the data transmitted is reduced by detecting the motion of objects. This will typically result in 50% - 80% savings in bits.

Motion vector representation

- A motion vector is specified with two components (horizontal and vertical offset).
  Absence of motion vector is indicated with (0,0).
  - Offset is calculated starting from the top left pixel :
    - Positive values indicate top and right.
    - Negative values indicate bottom and left.
  - 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.
  If motion vector is (0,0) the predictive base is the same macroblock in the reference frame

- Example: the match of the shaded macroblock of the current frame in the previous frame is in position (24,4). Then the forward predicted motion vector for the current frame is (8, -4)

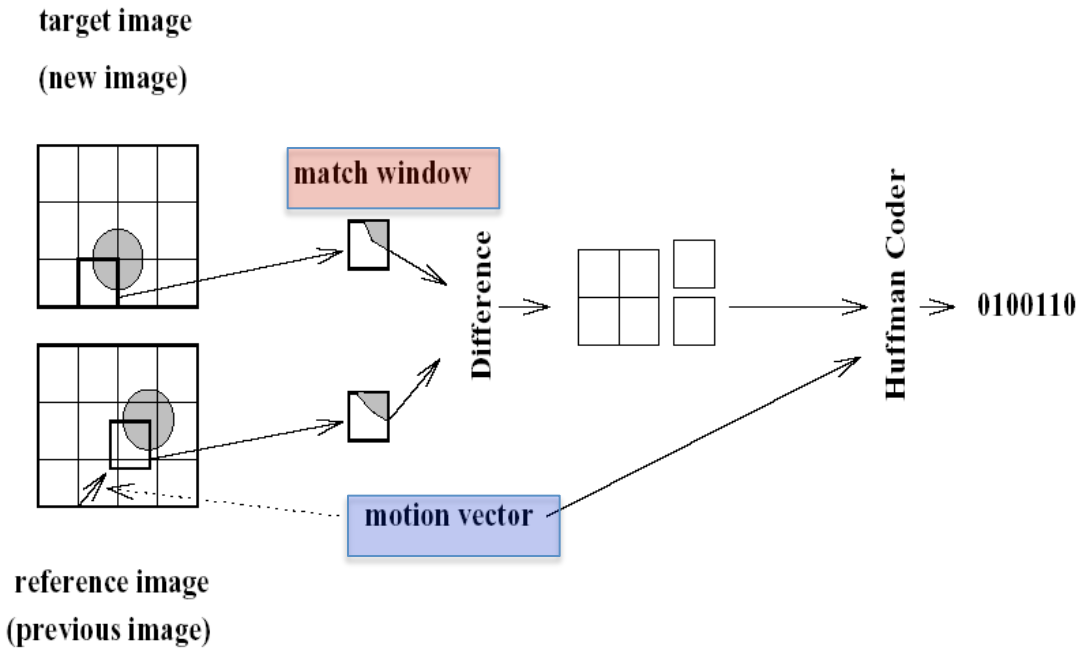


Block motion compensation

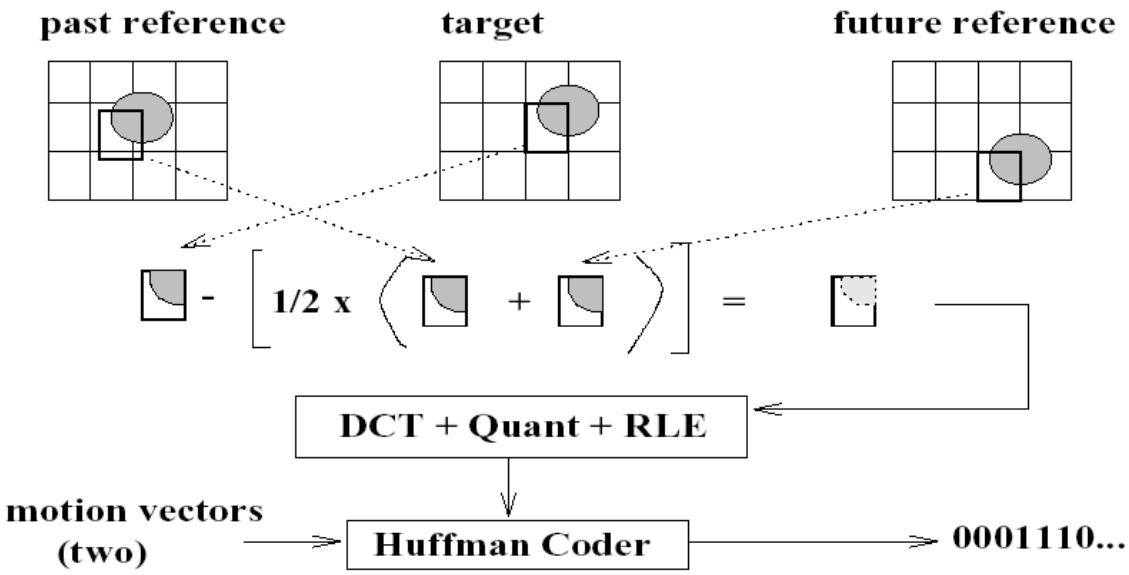
Error blocks

- The error block is obtained as the difference between two motion compensated blocks in adjacent frames. It is encoded as a normal block.

- For a P macroblock:



- For a B macroblock:

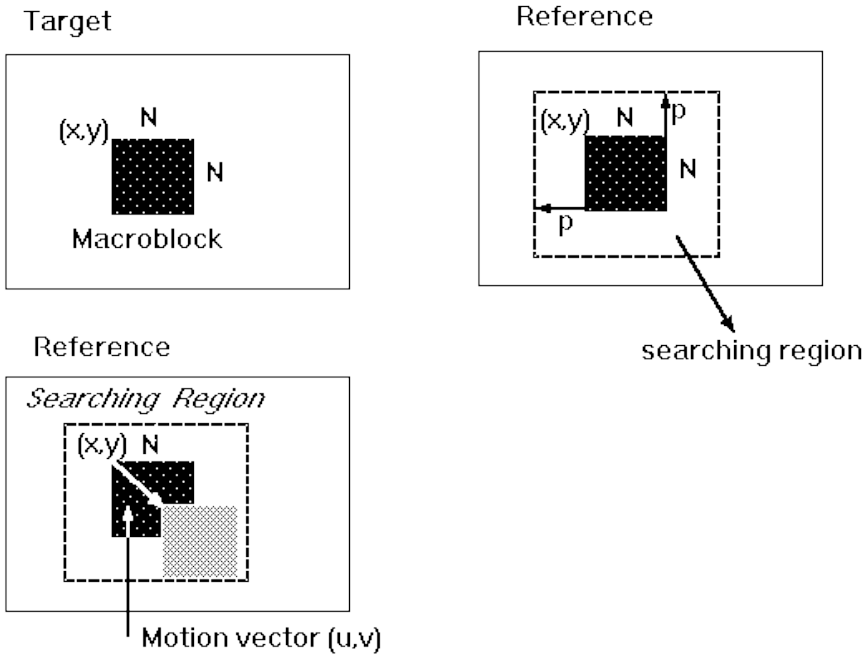


- For P/B error blocks a different quantization matrix is used wrt I-blocks:
  "16" value is set in all the matrix positions as error blocks have usually high frequency information
- Zig-zag scanning, RLE encoding and Huffman encoding follow.
  DC component and AC component are managed in the same way (there is no differential encoding as in I blocks)

- When a new P/B block is found DC component are reset. Motion vectors are reset when a new I macroblock is found.

Motion estimation by block matching

- Motion estimation is performed by applying block matching algorithms. Motion estimation can be performed in different ways depending on:
  - Different block matching techniques:
    - Mean Squared error (MSE).
    - Mean Absolute Error (MAE)
    - Sum of Squared Differences (SSD)
    - Sum of Absolute Errors (SAE)
  - Different search methods. Search methods often limit the search area for matching:



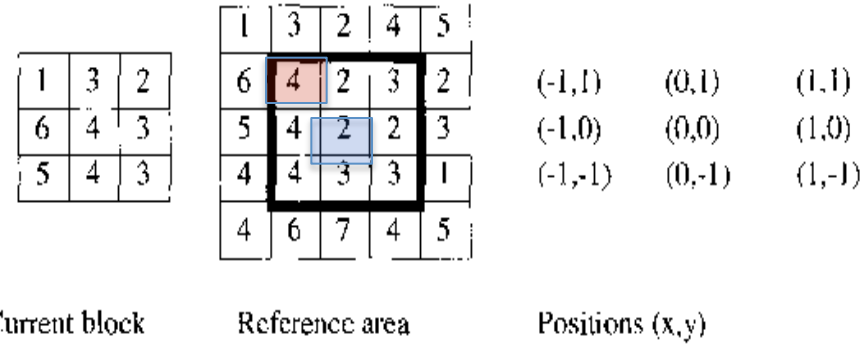
Mean Squared Error block matching

- 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:



MSE is:

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

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 block matching

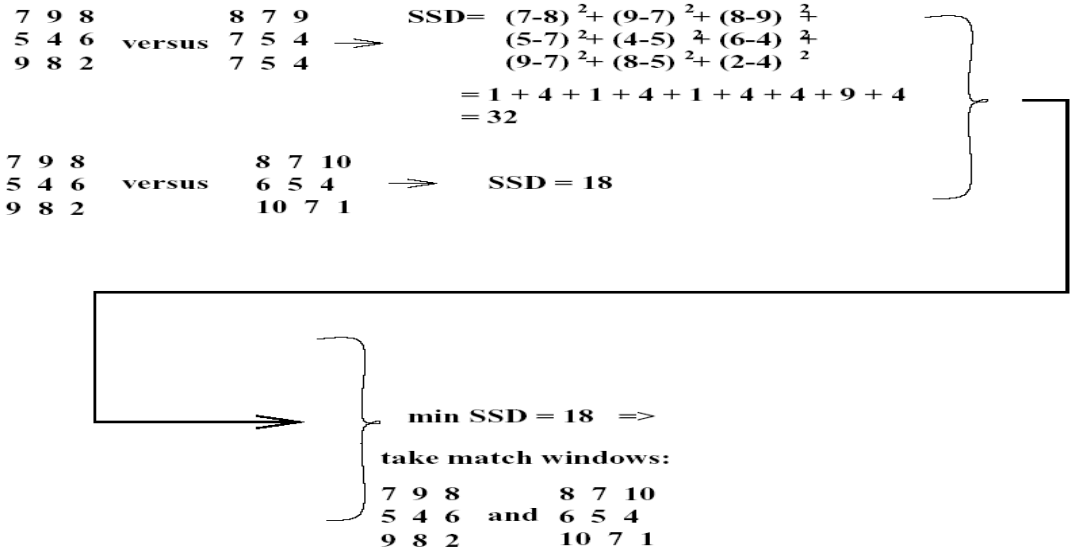
- Mean absolute error / difference (MAE/MAD)
  - Easier wrt MSE:

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

Sum of Squared Differences block matching

- Sum of Squared Differences (SSD):

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



Sensitive to outliers

Sum of Absolute Errors block matching

- Sum of absolute errors / differences (SAE/SAD):

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

Less sensitive wrt outliers wrt SSD

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	

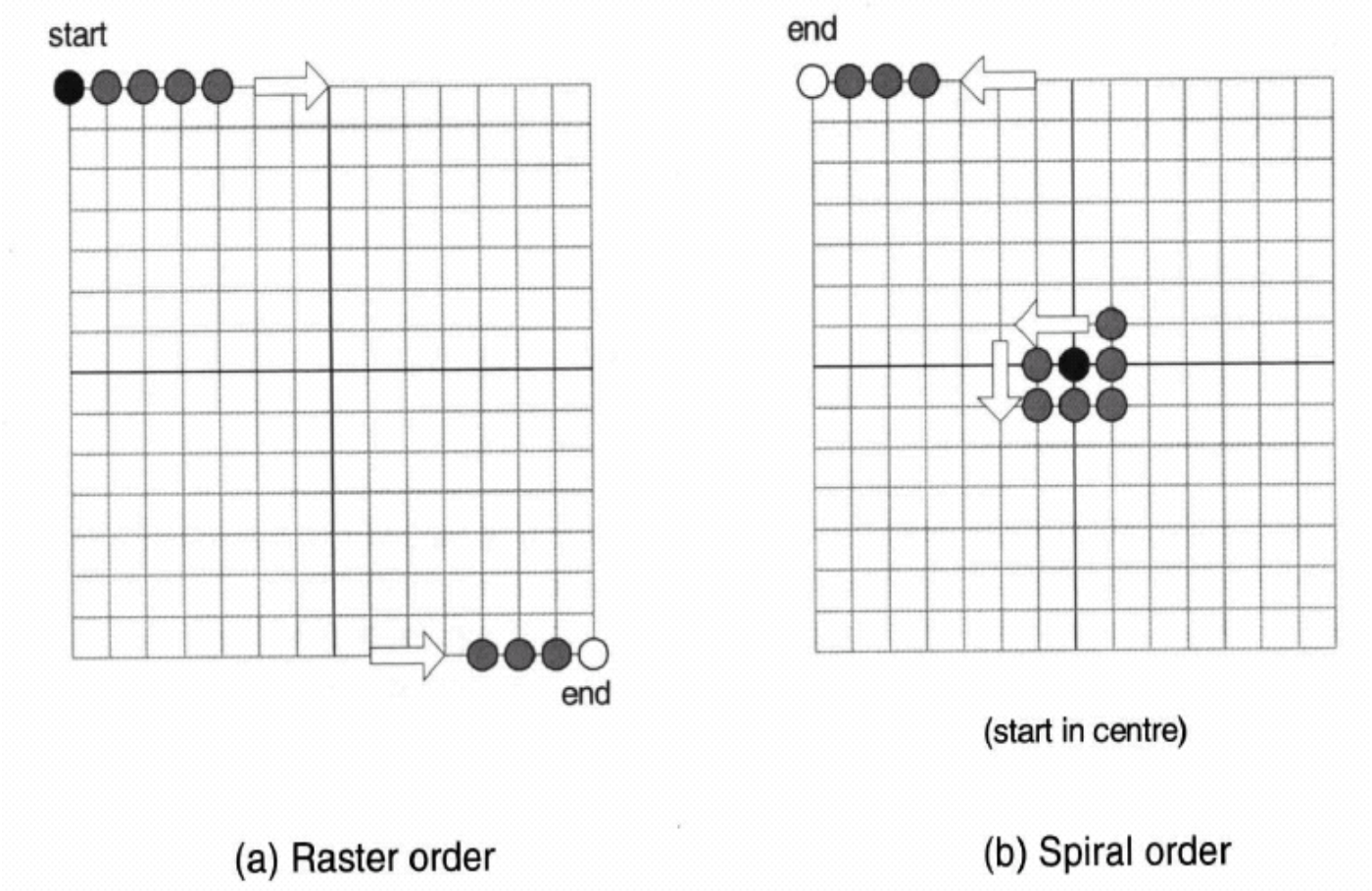
Search methods

- Full search methods check all the positions within the window with a pre-defined order criterion:
  - Raster
  - Spiral

  Full search detects the global minimum. It is computationally expensive, and only suited for hardware implementation.

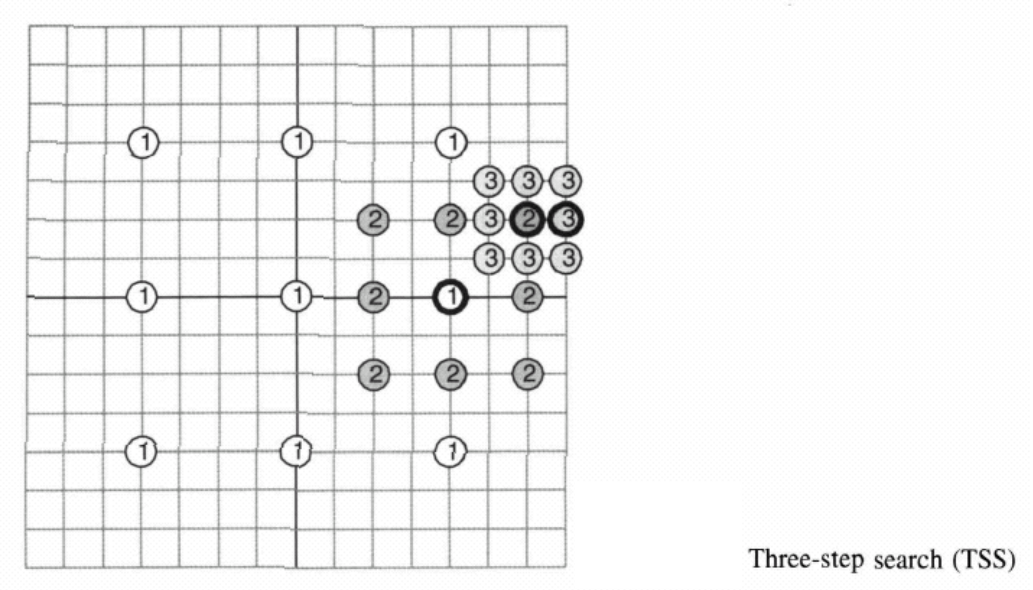
- Several methods employ a reduced number of comparisons wrt full search. Fast search methods may fall into local minima:
  - Three step Search
  - Logarithmic Search
  - One-at-a-Time Search
  - Nearest Neighbours Search

Raster, Spiral full search



Three step reduced search (TSS)

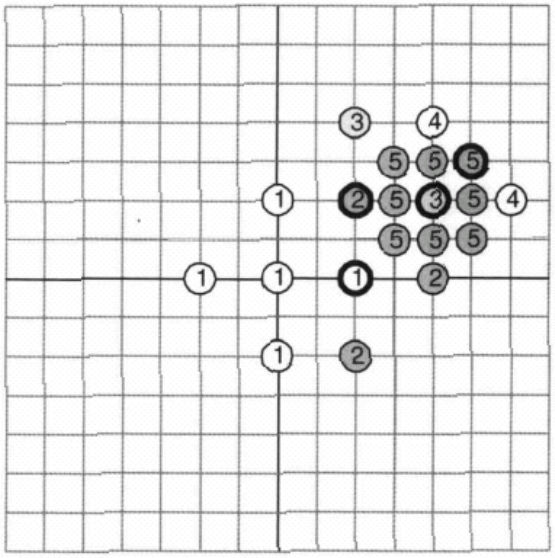
1. Start search from (0, 0).
2. Set S = 2^{N-1} (step size).
3. Look within 8 locations at +/-S pixel distance around (0, 0).
4. Select minimum SAE/SAD location between the 9 that have been analyzed
5. This location is the center for the new search
6. Set *S* = S/2.
7. Repeat from 3 to 5 until *S* =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



Three-step search (TSS)

Logarithmic reduced search

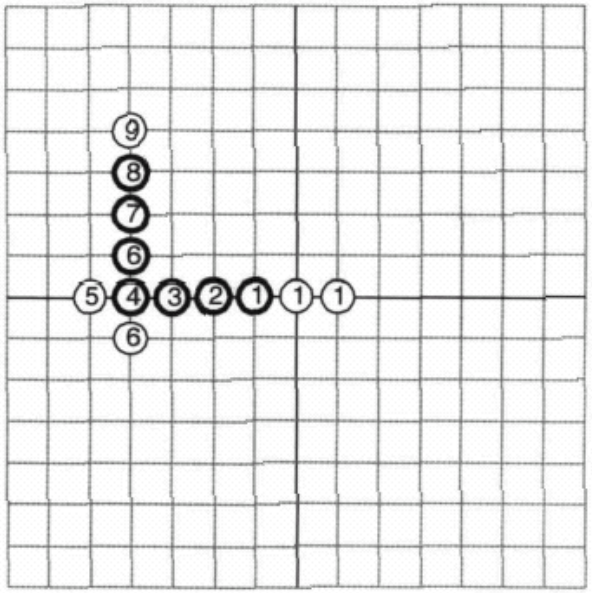
1. Start search from (0, 0).
2. Search in the 4 adjacent positions in the horizontal and vertical directions, at S pixel distance from (0,0) (S search step). The 5 positions model a ‘ + ’.
3. Set the new origin at the best match. If best match is in the central position of ‘+’ then *S* = S/2, otherwise *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 reduced search

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

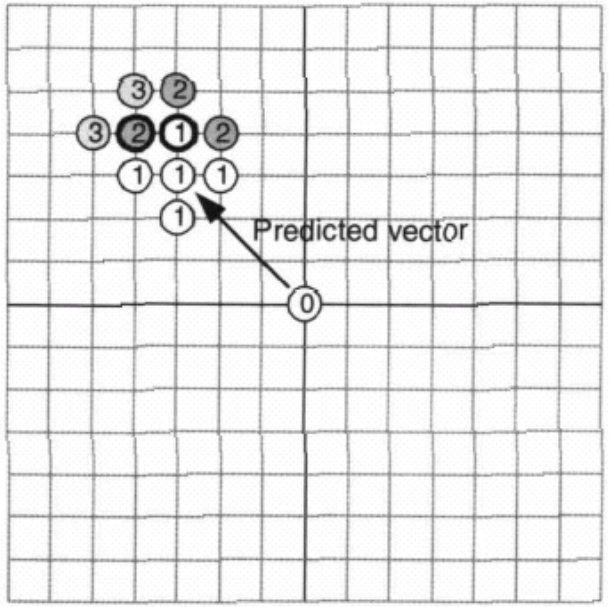


One-at-a-time search

Nearest Neighbours reduced search

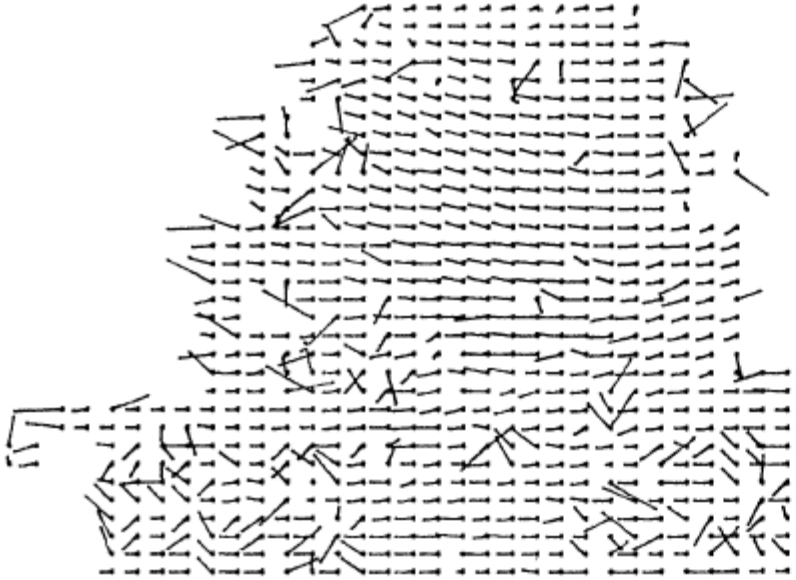
- Assumes that near macroblocks have similar motion vectors. Motion vectors are predicted by the near vectors already coded:
  1. Start from (0, 0).
  2. Set origin in the position of the predicted vector and start from there
  3. Search in the nearest '+'.
  4. If the origin is the best then take this 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.

- Used in H.263 e MPEG-4

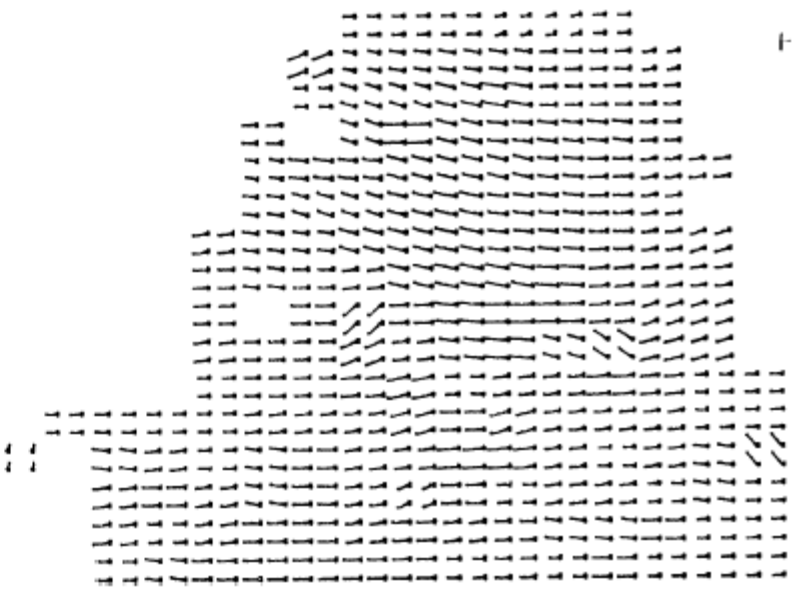


- Results into some smoothing of motion vectors

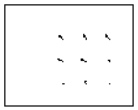




Full search block matcher



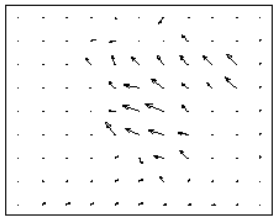
Hierarchical block matcher



(a)



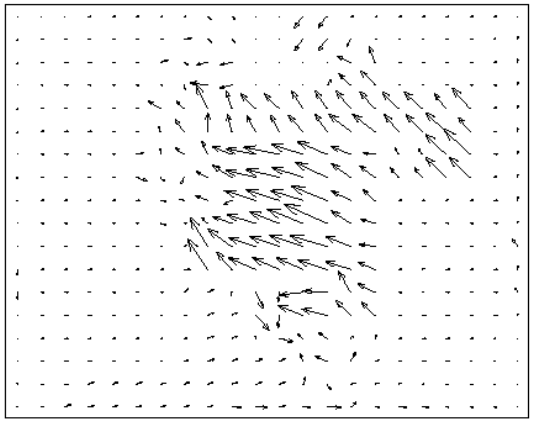
(b)



(c)



(d)



(e)



(f)

Search algorithms comparison

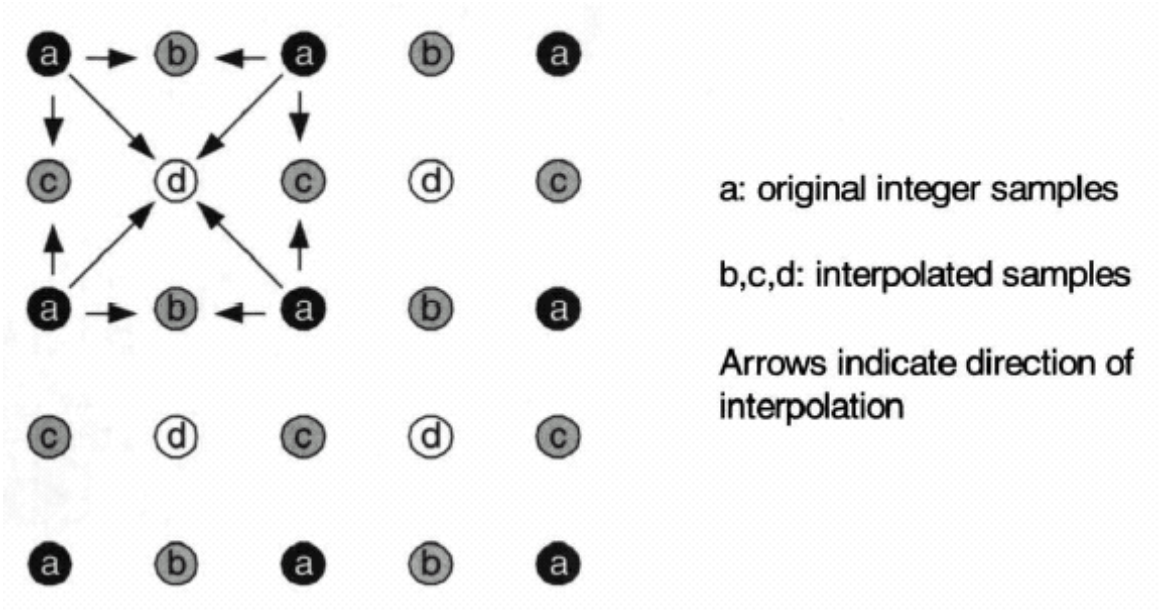
- Logarithmic search and one-at-a-time have low computational complexity and low matching performance as well.

- 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.

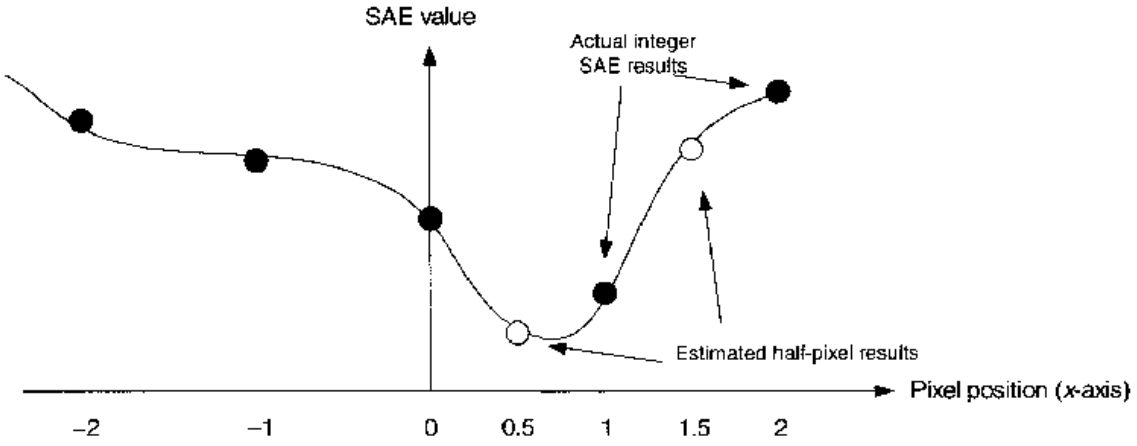
- Searching is performed as follows:
  1. Pixels are interpolated in the image search area so that a region is created with higher resolution than the original.
  2. Best match search is performed using both pixel and subpixel locations in the interpolated region
  3. Samples of the best matched region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



Half pixel interpolation

- As sub-pixel interpolation grows a better block matching performance is obtained at the expense of higher computational cost. Usually best matching is searched at integer position (full pixel) and hence refined at sub-pixel in the neighbourhood

- Motion compensation with half-pixel accuracy is supported in H.263, MPEG-1 e MPEG-2 standard
- Half pixel interpolation is used in MPEG-4. Higher interpolation (>1/4 pixel) is proposed for H.26L/ H.264 standard.



MPEG encoding – decoding

- In MPEG 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.

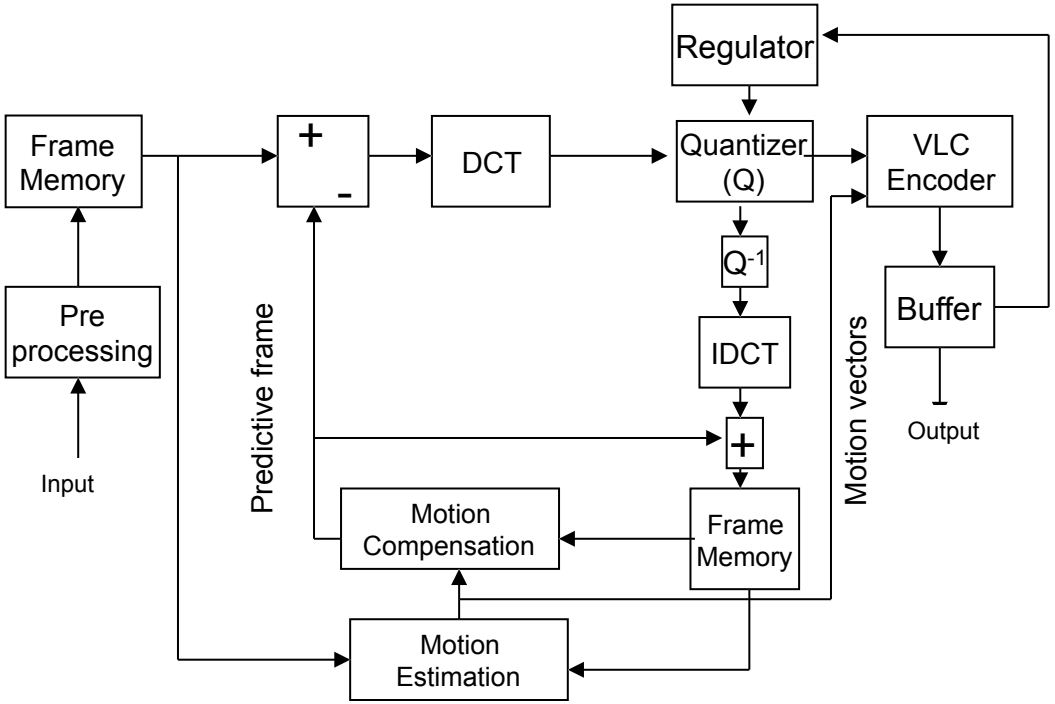
- Example: (a 12 picture long GOP)

  - 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)

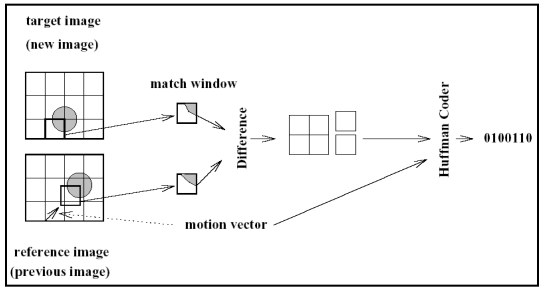
  - Encoding order and order in the coded bitstream:
    I(1) P(4) B(2) B(3) P(7) B(5) B(6) P(10) B(8) B(9) I(13) B(11) B(12)

  - 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)

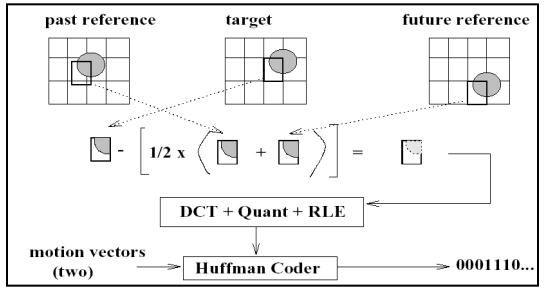
The MPEG encoder



P macroblock



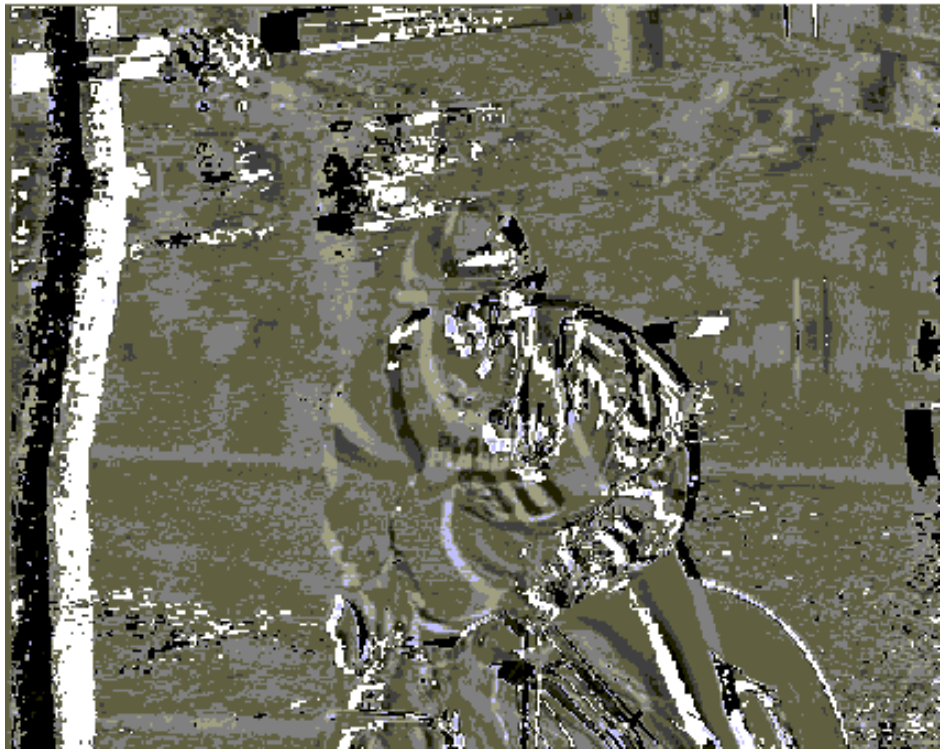
B macroblock

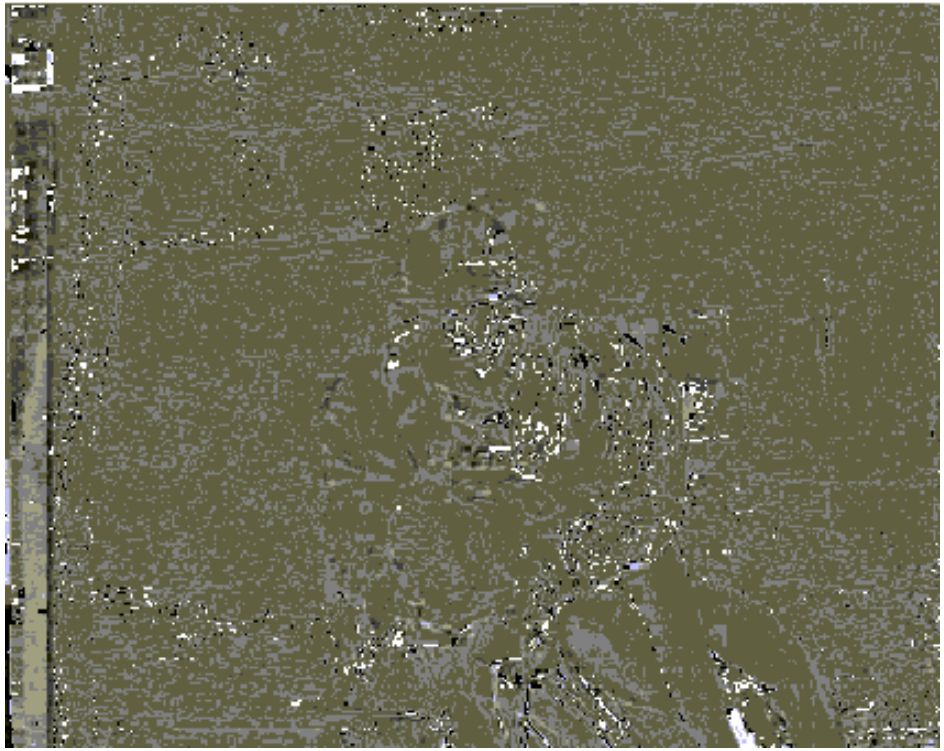






- Frame N to be encoded
- Frame at t= N-1 used to predict content of frame N

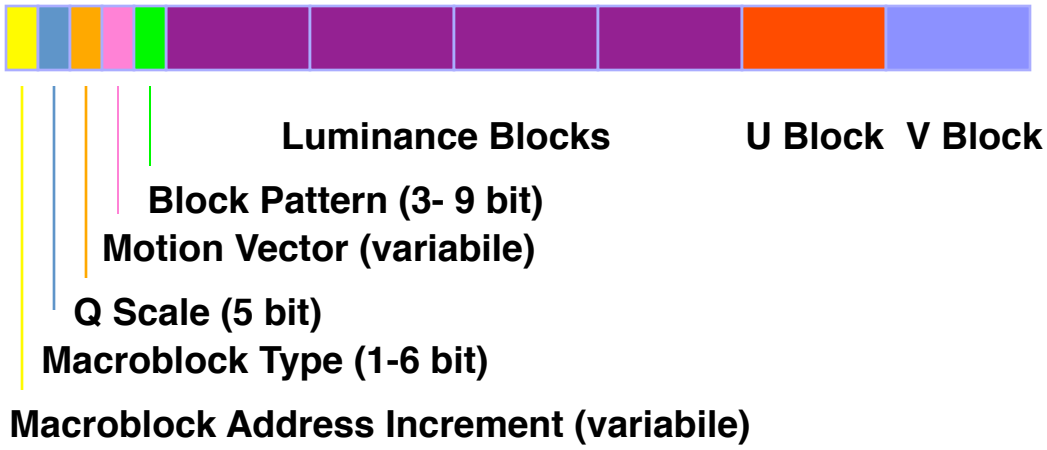




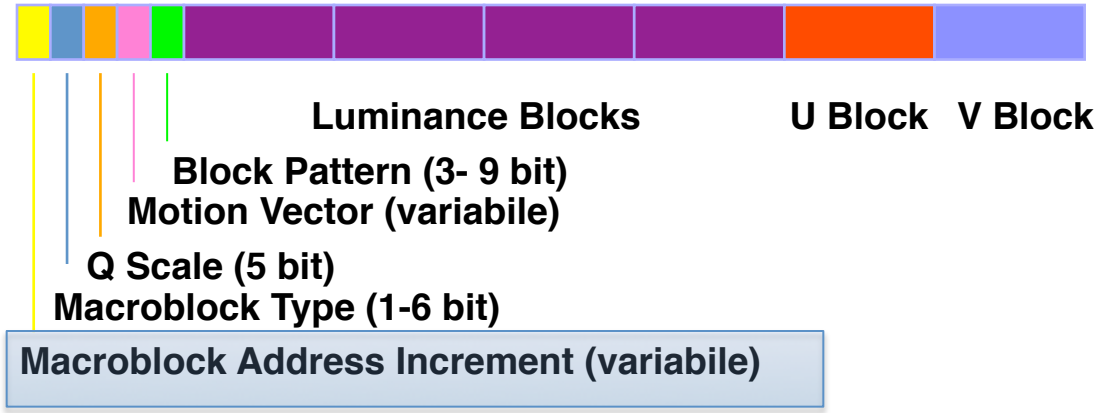
- Prediction error without motion compensation.
- Prediction error with motion compensation

Macroblock coding

- Macroblock information is encoded into a string:



Address Increment



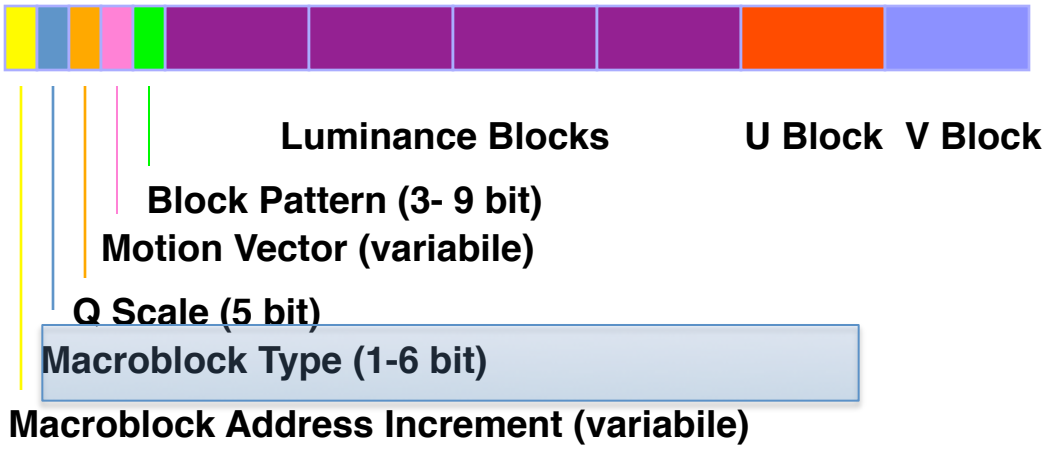
- Every macroblock has its own address:
  - MB_ADDR = MB_ROW * MB_WIDTH + MB_COL
    - MB_WIDTH = luminance width / 16
    - MB_ROW = # row top left pixel/ 16
    - MB_COL = # column top left row / 16

- Decoder maintains the address of the preceding macroblock PREV_MBADDR.
  - Set to **-1** at the start of each frame
  - Set to **(SLICE_ROW * MB_WIDTH-1)** at the start of each slice.

- The increment address is summed up to PREV_MBADDR to obtain the address of the current macroblock

- Address Increment is encoded with Huffman, based on a predefined table (the same used for I frame):
  - 33 codes (1-33).
    - 1the smallest (1-bit)
    - 33 the largest (11-bit)
  - 1 ESCAPE code
    - ESCAPE: add 33 to the following increment address (several ESCAPE can be used)

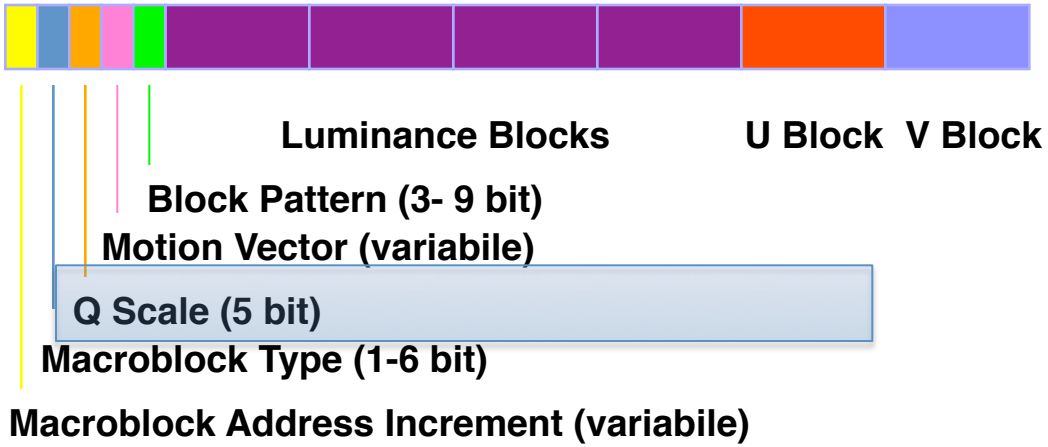
Macroblock Type



- Macroblock Type indicated whether macroblock is Intra or not if Q Scale, Motion Vector, and Block Pattern exist. It is coded with Huffman.

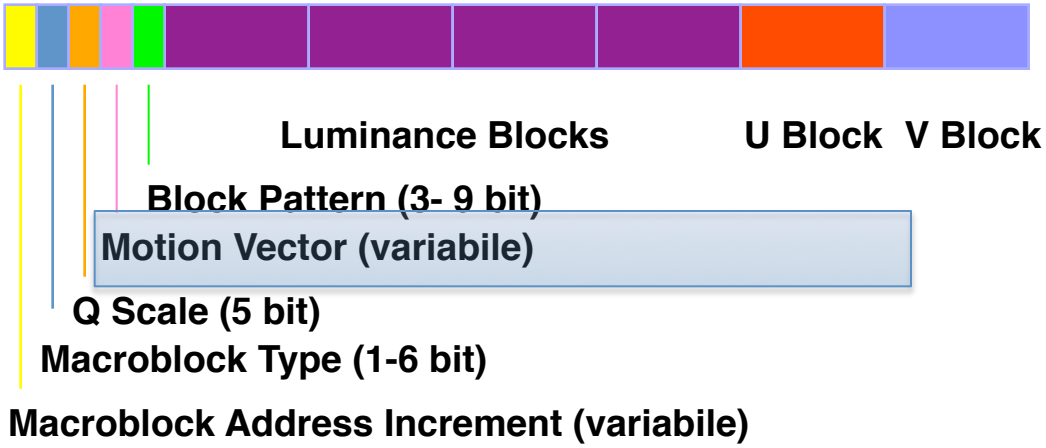
- 8 possible macroblock type (1 - 6 bit).

Quantization Scale



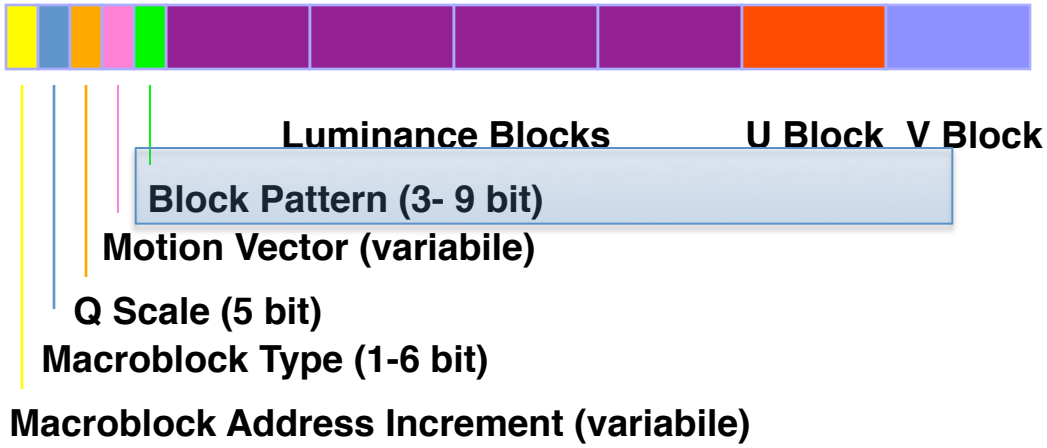
- Quantization scale has value 1 - 31 that are interpreted as 2 - 62 (only even values). 5 bit.
- Decoder uses the current Q-scale unless specified

Motion Vector



- Motion Vector is used to define a predictive base for the current macroblock from the reference image.
- Prediction is used to determine motion vectors. Difference between the predicted value and the actual value is encoded with Huffman

Block Pattern



- Block Pattern indicates which blocks have high error wrt the reference block so to be compensated. Block compensation is necessary to have a predictive base that is as much similar as possible to the current macroblock.

- If block pattern is not present then matching between the current block and its corresponding block is sufficiently good and there is non need for coding

Part II - MPEG 2

MPEG2: why another standard

- MPEG-1 was suitable for storage media. Was aimed at VHS quality at 1.5 Mbps.

- 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 broadcast by terrestrial, cable, and direct broadcast satellite TV systems. It also specifies the format of movies and other programs that are distributed on DVD and similar discs.

- MPEG-2 Video is similar to MPEG-1, but also provides support for interlaced video format used by analog broadcast TV systems. MPEG-2 video is not optimized for low bit-rates (less than 1 Mbit/s), but outperforms MPEG-1 at 3 Mbit/s and above

- MPEG2 supports higher bit rates and a larger number of applications:
  - 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

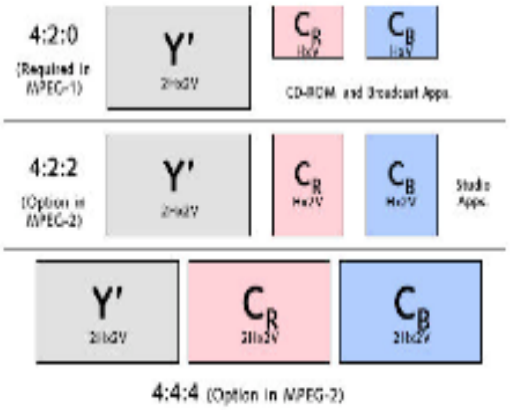
Color subsampling

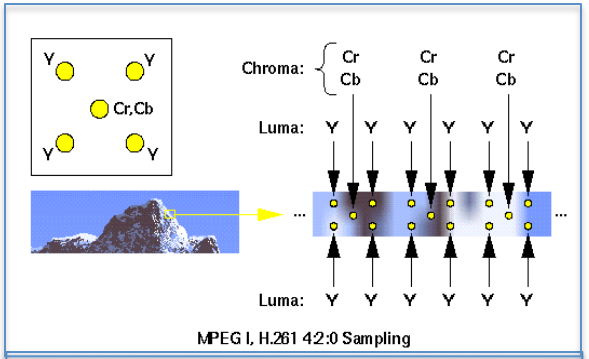
- MPEG2 supports different color subsamplings:
  - 4:2:0 (as MPEG1)

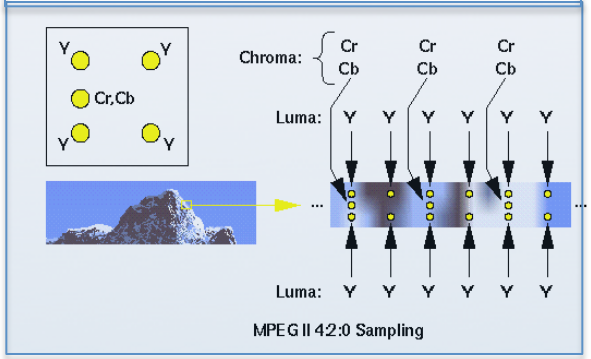
    In MPEG1 chrominance samples are horizontally and vertically positioned in the center of a group of 4 luminance samples.
    In MPEG-2 chrominance samples co-located on luminance samples

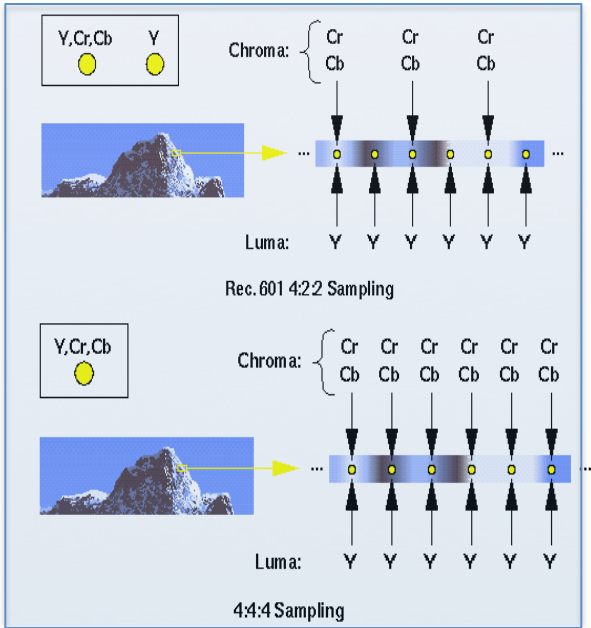
  - 4:2:2, 4:4:4

    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





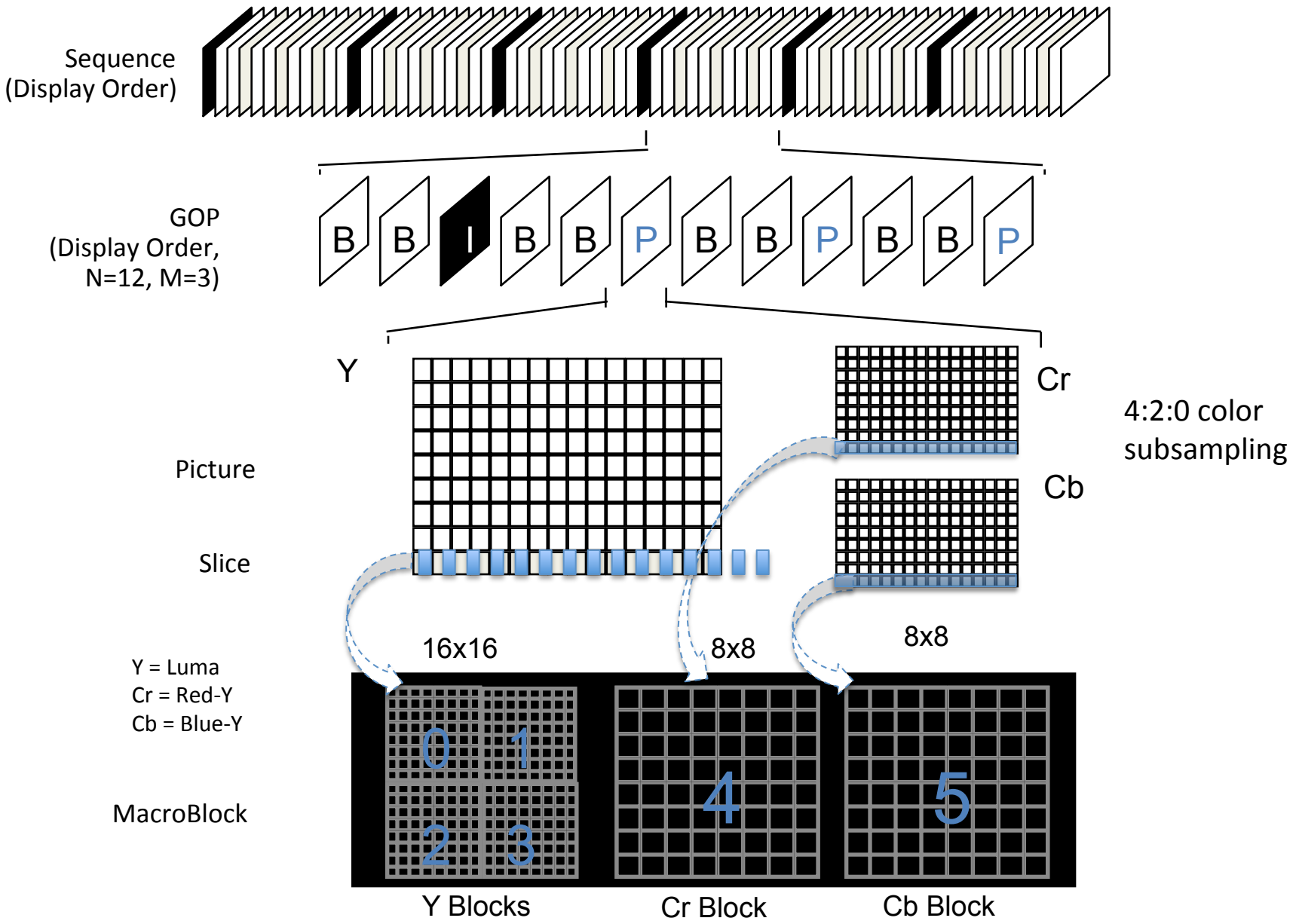




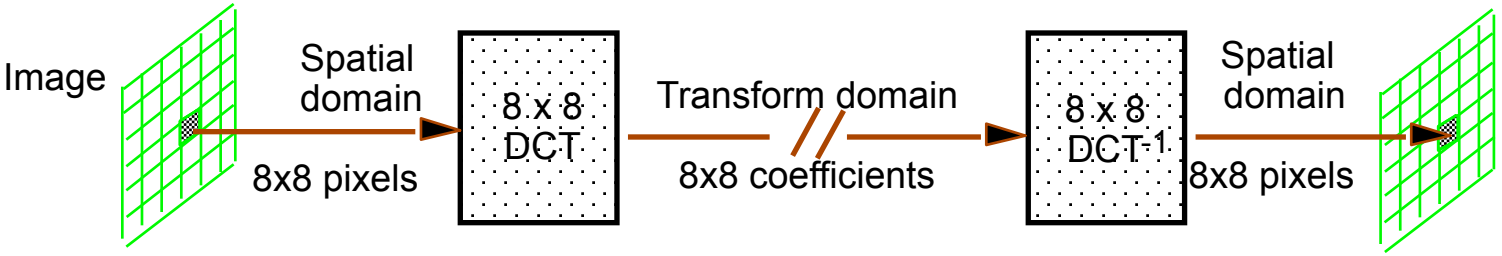
I, P, B frame encoding

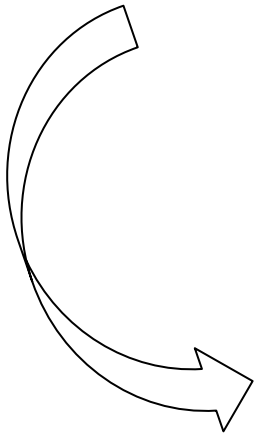
- Same as MPEG1. I, P and B frames (pictures) are encoded on a macroblock basis. DCT coding is used.
  - P-pictures have interframe predictive coding
    - Macroblocks may be coded with forward prediction from references made from previous I and P pictures or may be intra coded (no prediction).
    - 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;
      - or may be intra coded (no prediction).
    - Backward prediction is done by storing pictures until the desired anchor picture is available before encoding the "current" (stored) frames. The encoder forms a prediction error macroblock from either or their average. The prediction error is encoded using a block-based DCT
  - No D pictures

The MPEG2 stream

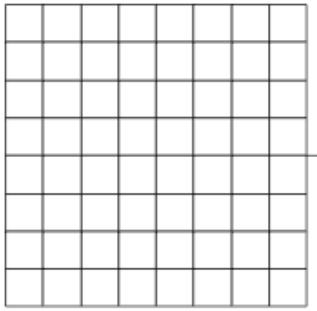


Discrete Cosine Transform and quantization scale





Input pixel block 8x8, 8-9 bits/sample



DCT

Coefficient block 8x8, ≥ 12 bits/sample; default quantization matrix coefficients

DC	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

Increasing vertical frequency

Increasing horizontal frequency

Non linear quantization scale is also available

8 x 8 DCT:

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

8 x 8 IDCT:

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

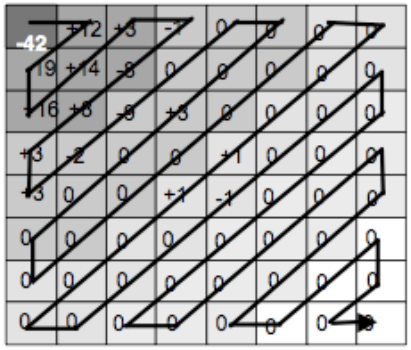
Multiple scanning options

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

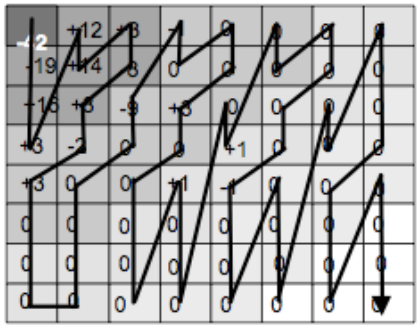
-42	+12	+3	-1	0	0	0	0
-19	+14	-8	0	0	0	0	0
+16	+8	-9	+3	0	0	0	0
+3	-2	0	0	+1	0	0	0
+3	0	0	+1	-1	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

Increasing vertical frequency

Increasing horizontal frequency



Zigzag Scan

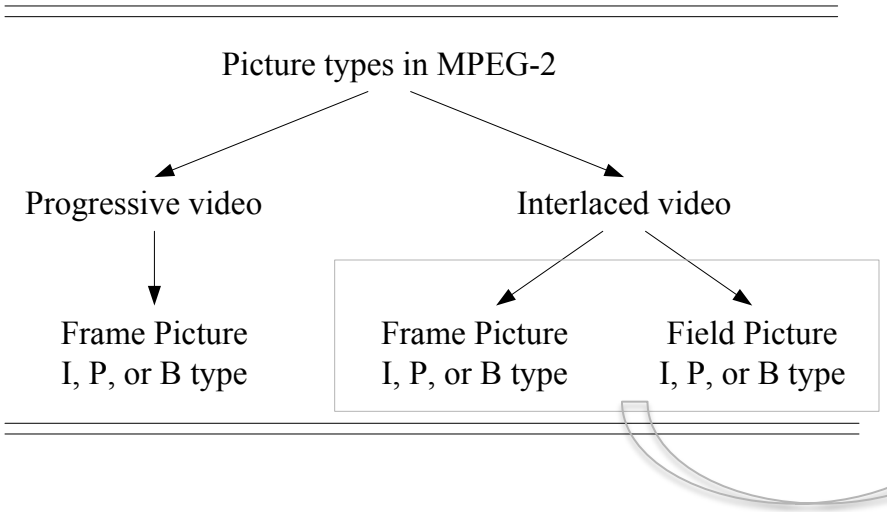


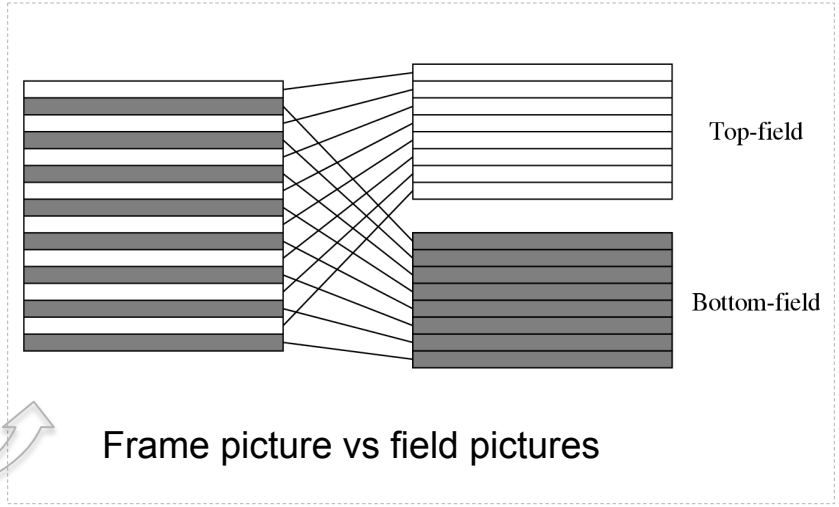
Vertical Scan

Frame vs field-based coding

- MPEG2 supports both *progressive* and *interlaced* video.
  - *Progressive* frames are encoded as *frame pictures* with frame-based DCT coded macroblocks only and the 8x8 four blocks that compose the macroblock come from the same frame of video
  - *Interlaced* frames may be coded as either a *frame picture* or as *two separately coded field pictures.* The encoder may decide on a frame by frame basis to produce a frame picture or two field pictures. Field-based DCT coding can be applied only to interlaced sequences.
    - In the case of a frame picture is produced**,** frame or field-based DCT macroblock coding can be used (on a macroblock-by-macroblock basis)
    - In the case of field pictures are produced, field-based DCT macroblock coding is used and all the blocks come from one field

  Frame-based DCT is suited for macroblocks with little motion and high spatial activity. Field-based DCT is suited for high motion macroblocks.



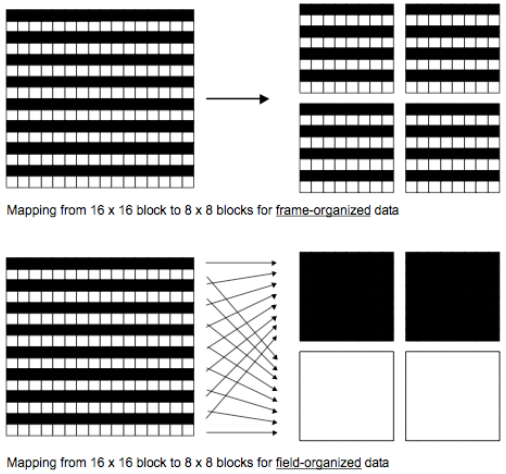


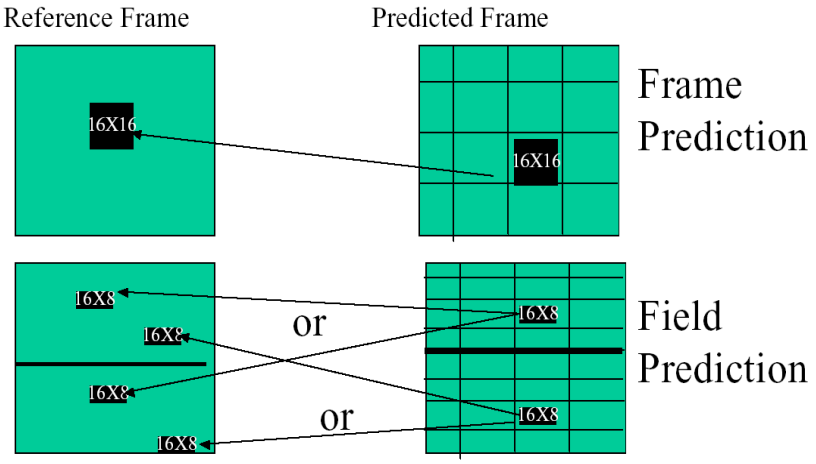
Frame picture vs field pictures

Interlaced frame production: frame and field-based prediction

- For interlaced sequences with frame production it is possible to use either *frame-based* or *field-based* prediction:
  - *Frame prediction for frame-pictures*: Identical to MPEG-1 prediction methods. Frame-based prediction uses a single motion vector for each 16x16 macroblock.
  - *Field prediction for frame-pictures:* the top-field and bottom-field of a frame-picture are treated separately.

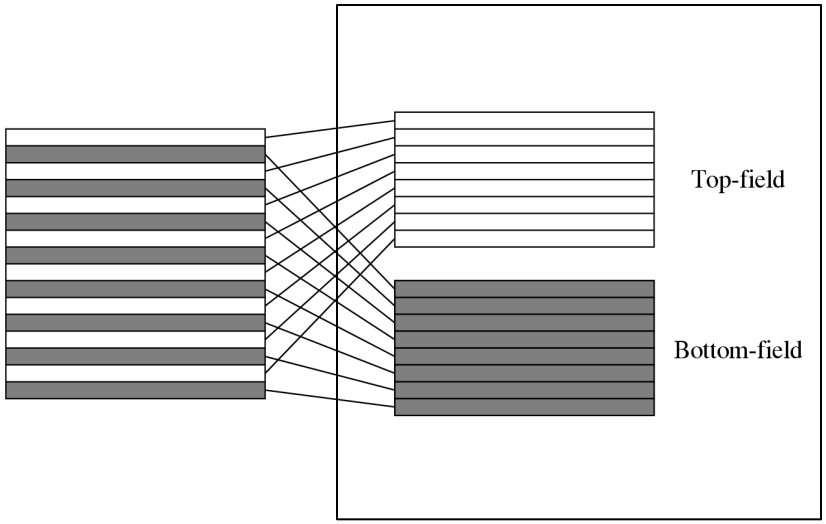
- The size of 16×16 in the Field picture covers a size of 16×32 in the Frame picture. It is too big size to assume that behavior inside the block is homogeneous. Therefore, 16×8 size prediction was introduced in Field picture. Two Motion Vectors are used for each macroblock and come from the two most recent fields: the first is applied to the 16×8 block in the field 1 and the second is applied to the 16×8 block in field 2.

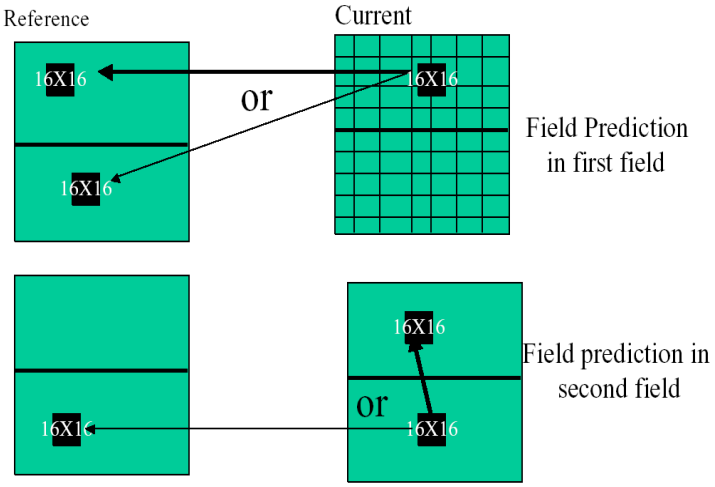




Interlaced field production: field-based prediction

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





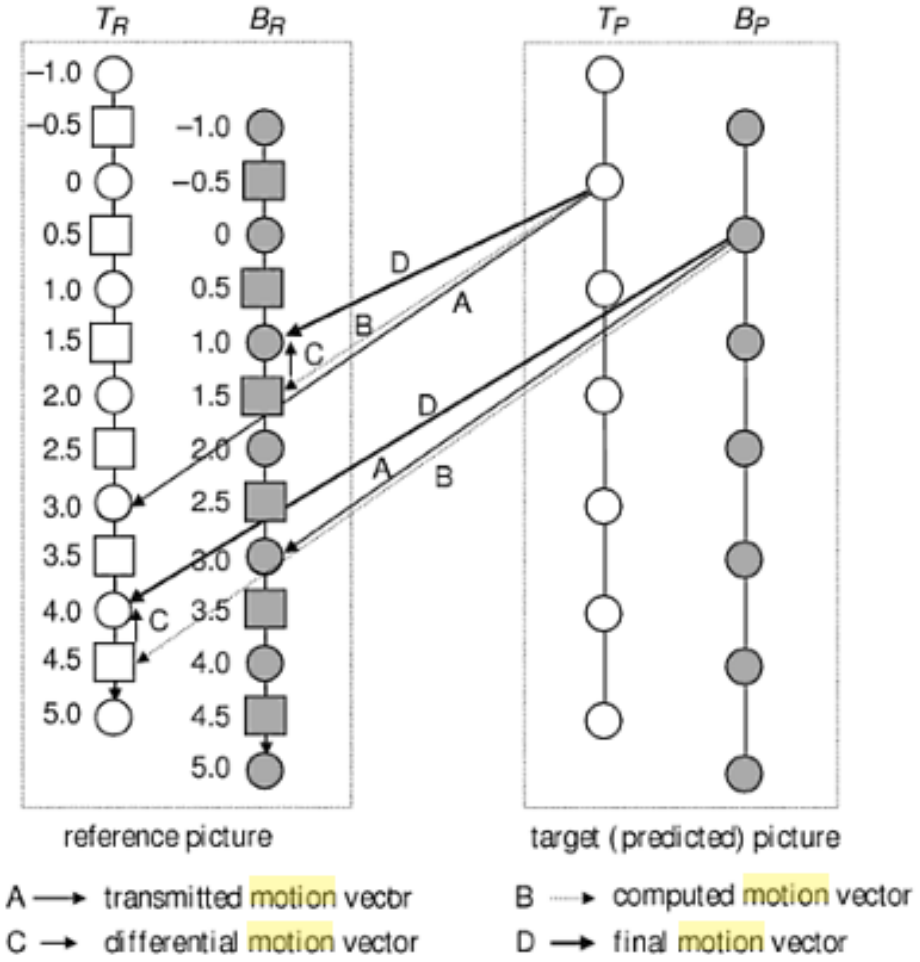
- Like macroblocks in frame-coded pictures 16x8 predictions can also be used (the upper and lower 8 lines of the macroblock have different predictions). This compensate for the reduced temporal prediction precision of field picture macroblocks as a result of the fact that fields possess half of the lines of frames.

Interlaced frame/field production: dual-prime prediction

- The idea of Dual Prime adaptive motion prediction is to send minimal differential Motion Vector information for adjacent field Motion Vector data
- Dual-Prime Prediction is a prediction mode in which two forward field-based predictions are averaged. The predicted block size is 16x16 luminance samples. Only one motion vector is encoded with a small differential motion correction
- It is only used in interlaced P-pictures when there have been no B-pictures between the P-picture and its reference frame. This is the only mode that can be used for either frame-pictures or field-pictures. It avoids the frame re-ordering needed for bi-directional prediction but achieves similar coding efficiency.

- For field pictures, two motion vectors are derived from this data and are used to form two predictions from two reference fields. These two predictions are combined to form the final prediction.
- For frame pictures, this process is repeated for each of the two fields. Each field is predicted separately, giving rise to a total of four field predictions which are combined to form the final two predictions.

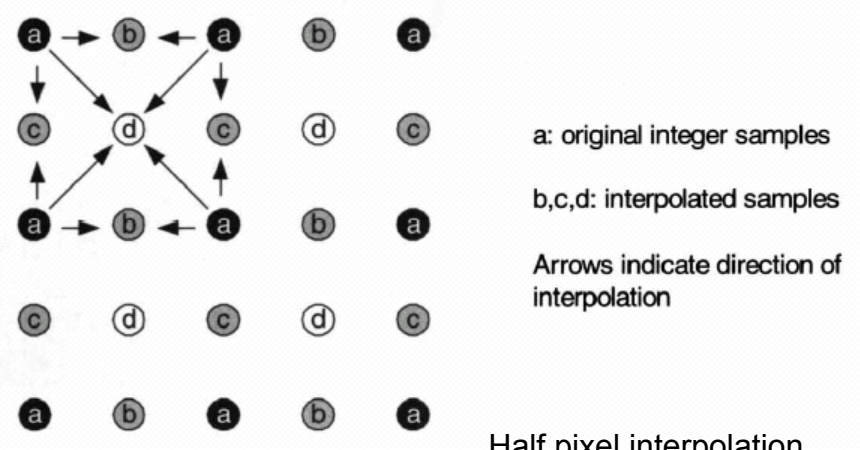
dual-prime prediction for field prediction for frame pictures



Half pixel interpolation for motion estimation

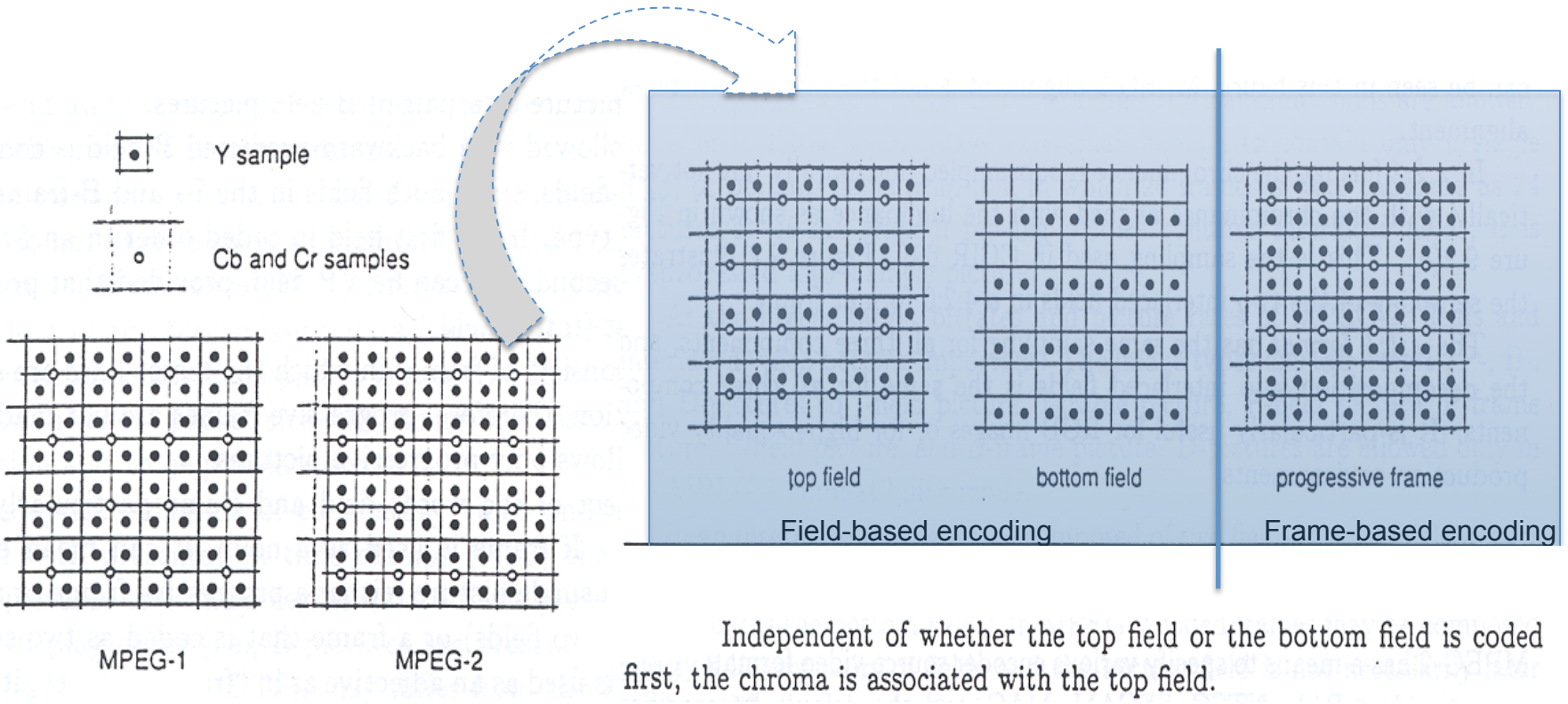
- MPEG2 uses half-pixel interpolation for motion vector estimation.

- 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.



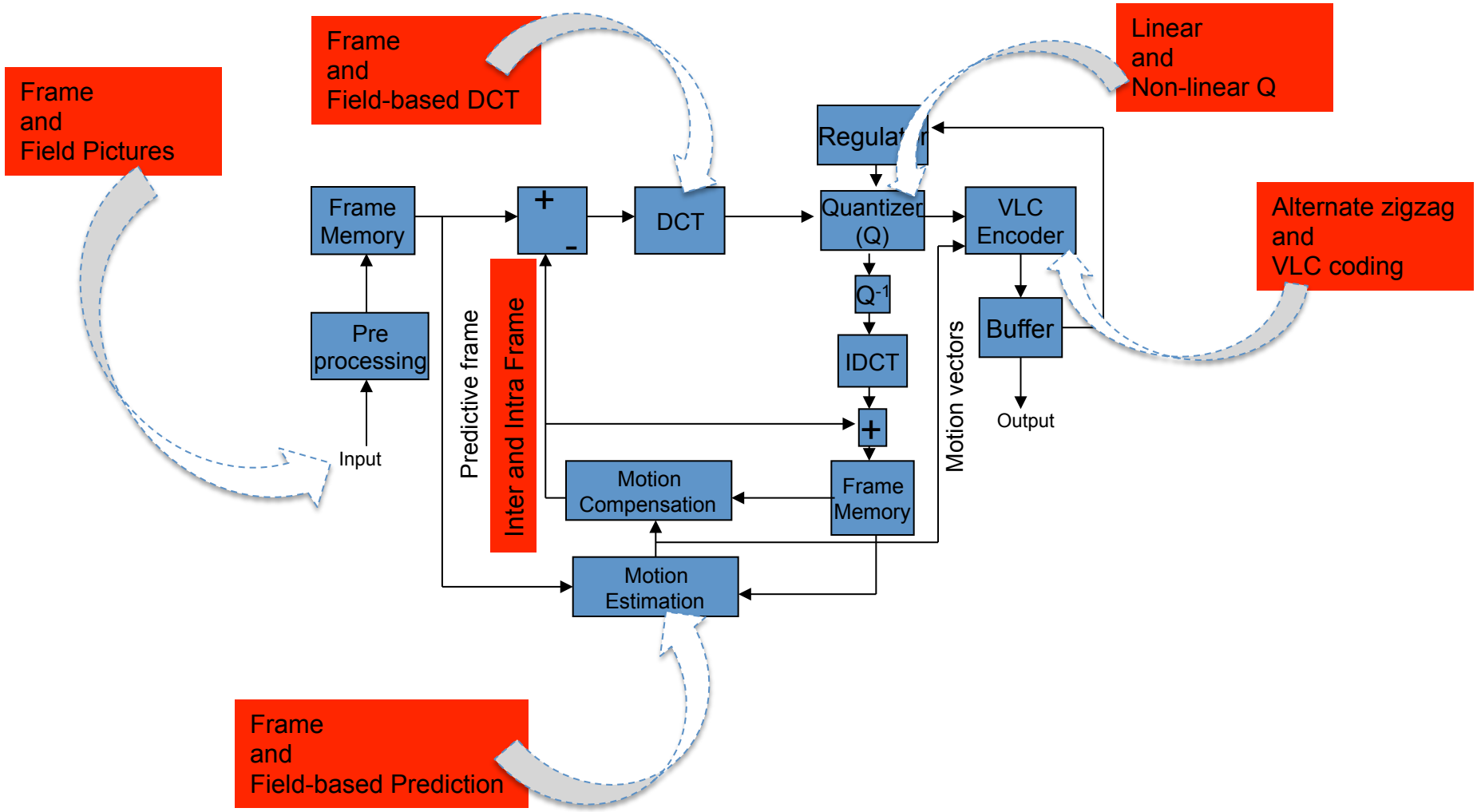
Half pixel interpolation

- Example of 4:2:0 format: chrominance is sampled 2:1 both vertically and horizontally but exist different alignments depending on whether frame-based or field based coding is used



Independent of whether the top field or the bottom field is coded first, the chroma is associated with the top field.

MPEG2 Enhancements

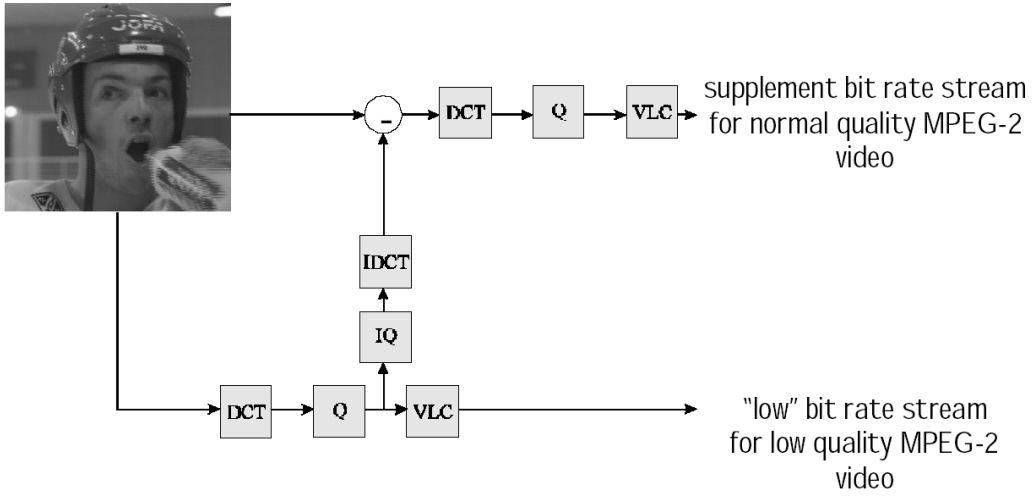


Scalability

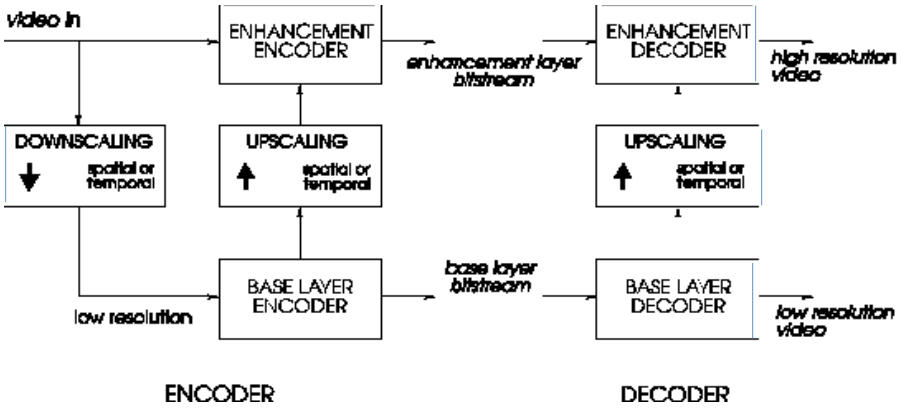
- Scalability is the ability of decoding only part of the stream to obtain a video of the resolution desired. It is possible to have:
  - SNR scalability,
  - Spatial scalability
  - Temporal scalability

- 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.

- 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

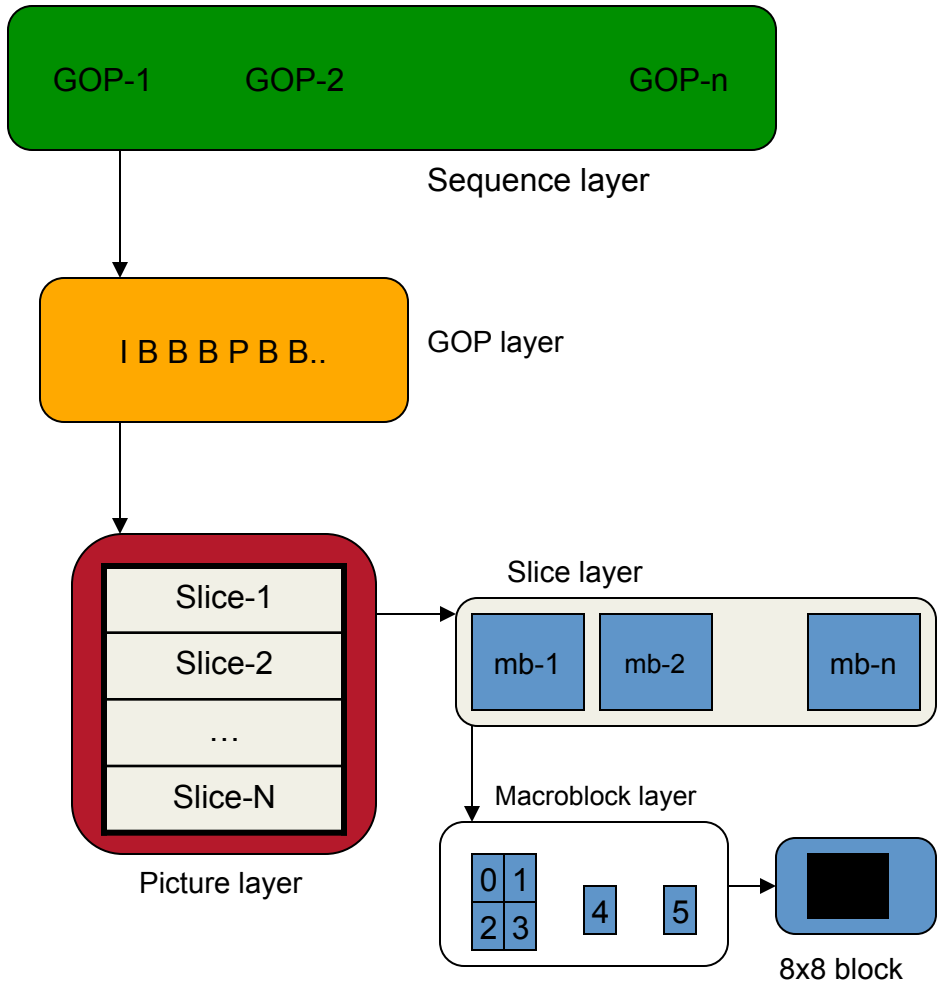
- Low Level
  - MPEG1 CPB (Constrained Parameters Bitstream): max. 352x288 @ 30 fps
- Main Level
  - MPEG2 CPB (720x576 @ 30 fps)
- High-1440 and High Levels
  - Typical of HDTV

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 - MPEG 4

MPEG4

- MPEG4 has been designed for.
  - Real-time communication (videoconferencing)
  - Digital television
  - Interactive graphic applications (DVD, ITV);
  - World Wide Web applications

- 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 digital video composition, manipulation, indexing, and retrieval
  - 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 spartial resolution and low frame rate (up to 15 fps).

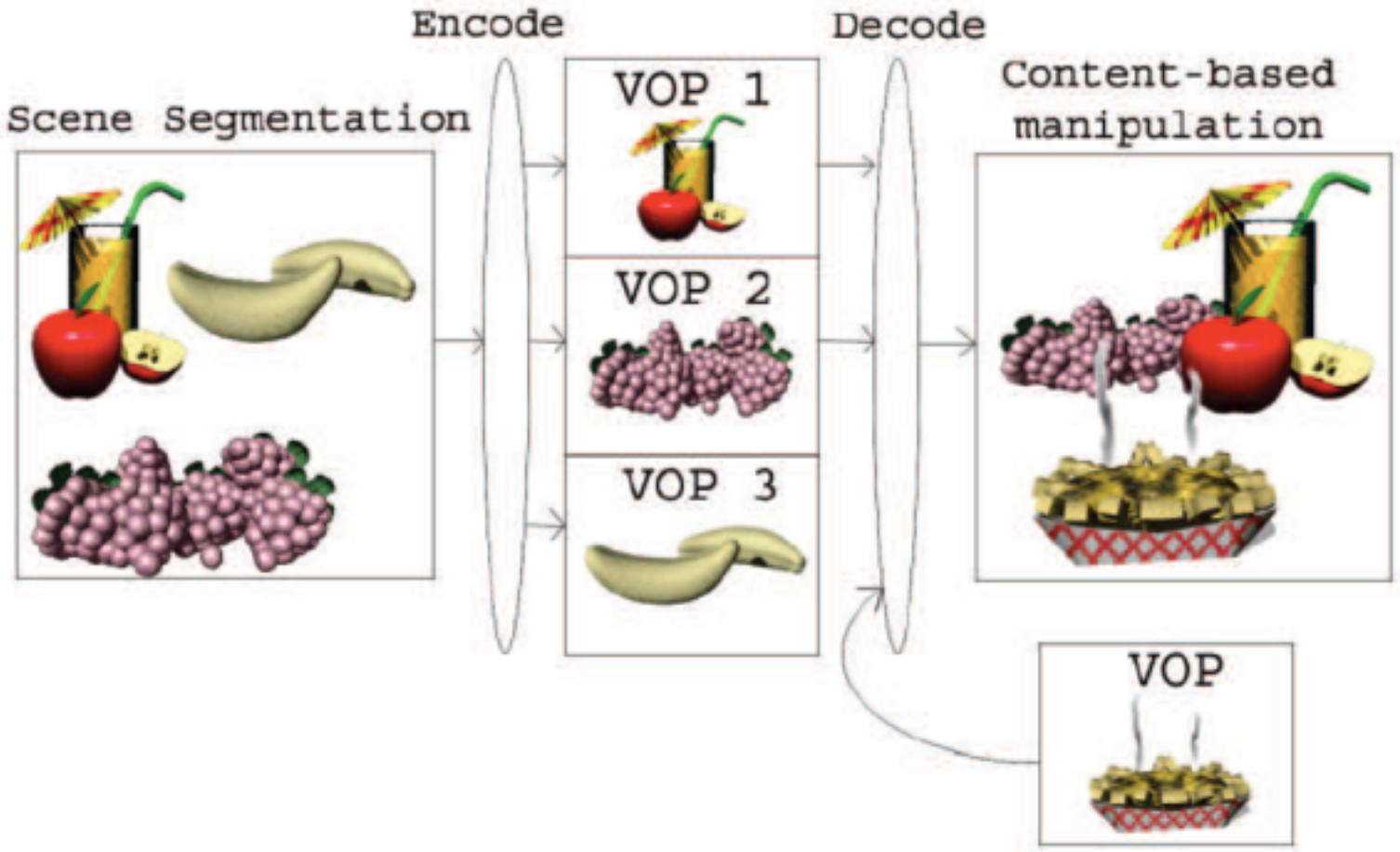
Distinguishing elements

- 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

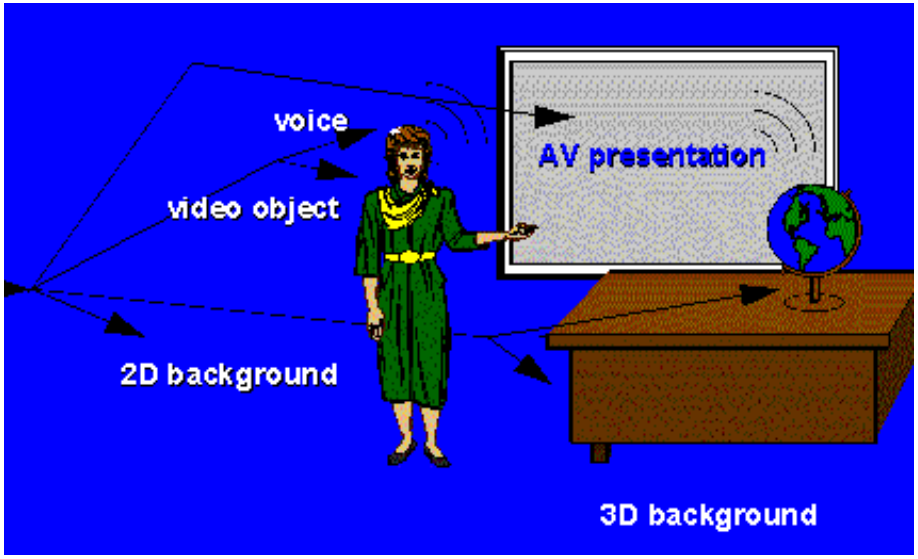
- MPEG4 includes technologies to support:
  - *server side*
    - Encoding based on and audio-visual objects. When a VOP is the rectangular frame it corresponds to MPEG2
    - Audio-visual objects manipulation
    - Hierarchical scene composition (audio-visual objects local coordinates, temporal syncronization….. 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 natural and artificial 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…)

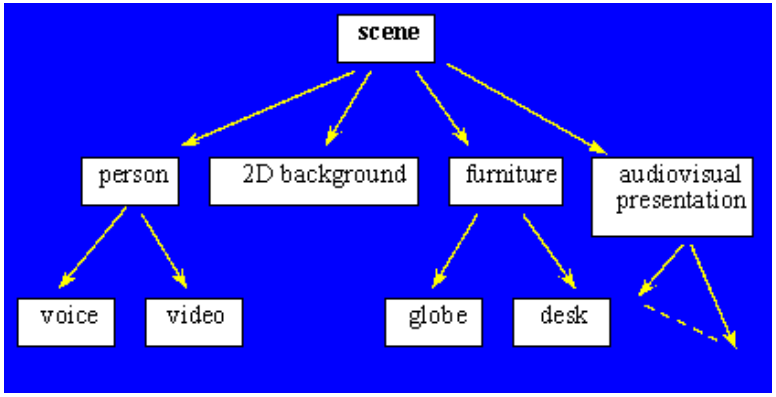
MPEG4 operations



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…)

MPEG4 encoding

- MPEG4 provides algorithms and tools to:
  - Compress images and video
  - Compress textures to be mapped onto 2D and 3D meshes
  - Compress geometric streams that change through time for 2D mesh animation
  - Access to any visual object
  - Manipulation of images and video sequences
  - Object-based coding of image and video content
  - Scalability based on content of textures of images and video
  - Spatial, temporal and quality scalability

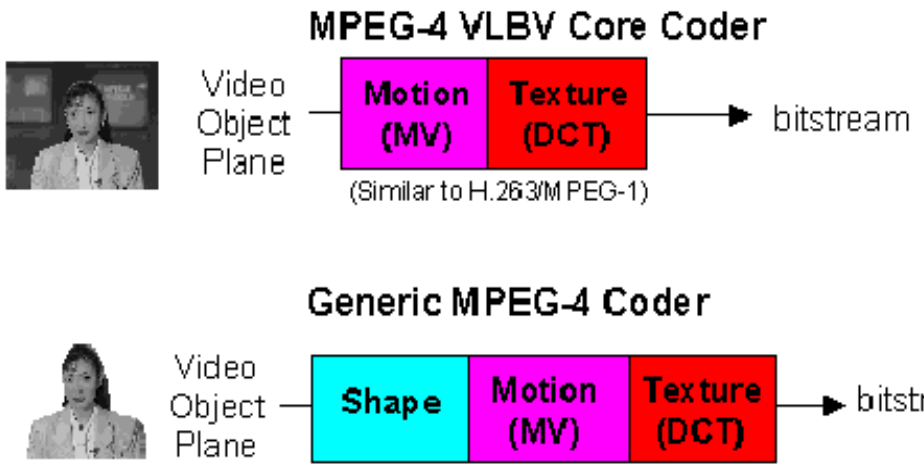
Compression

- MPEG4 compression is the same as MPEG1 and MPEG2 compression. Rectangular frames at different:
  - Bitrate
  - Frame rate
  - Input format
  - Quality Scalability
  - Spatial Scalability
  - Temporal Scalability

- Specifically it supports:
  - Progressive and interlaced video
  - SQCIF/QCIF/CIF/4*CIF/CCIR 601, up to 2048*2048
  - YCbCr/Alpha
  - 4:2:0 color quantization (4:2:2 e 4:4:4 for studio quality)
  - Continuous variable frame rate

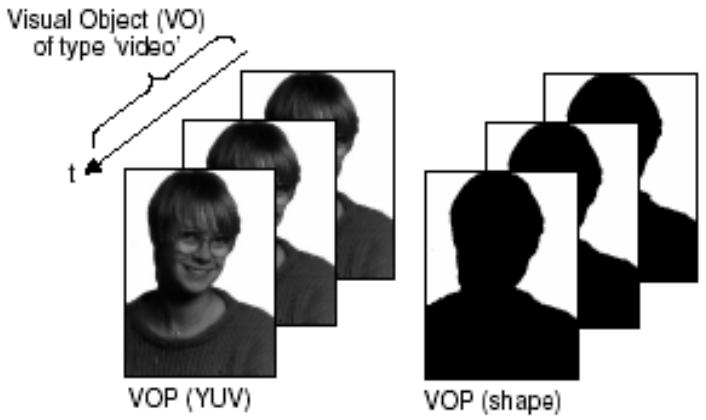
Object-based coding: 2D natural audiovisual objects

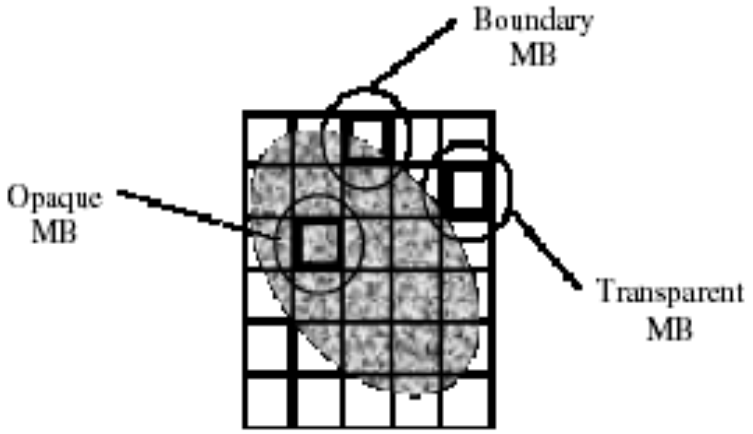
- In MPEG4 video is regarded as a composition of 2D objects (they can be placed in a 3D space).
- In object-oriented coding 2D objects can be of any arbitrary shape and texture. Both shape and texture must be encoded
- If shape is not considered, MPEG4 encoder is based on motion compensation as in MPEG1 and MPEG2, using macroblocks



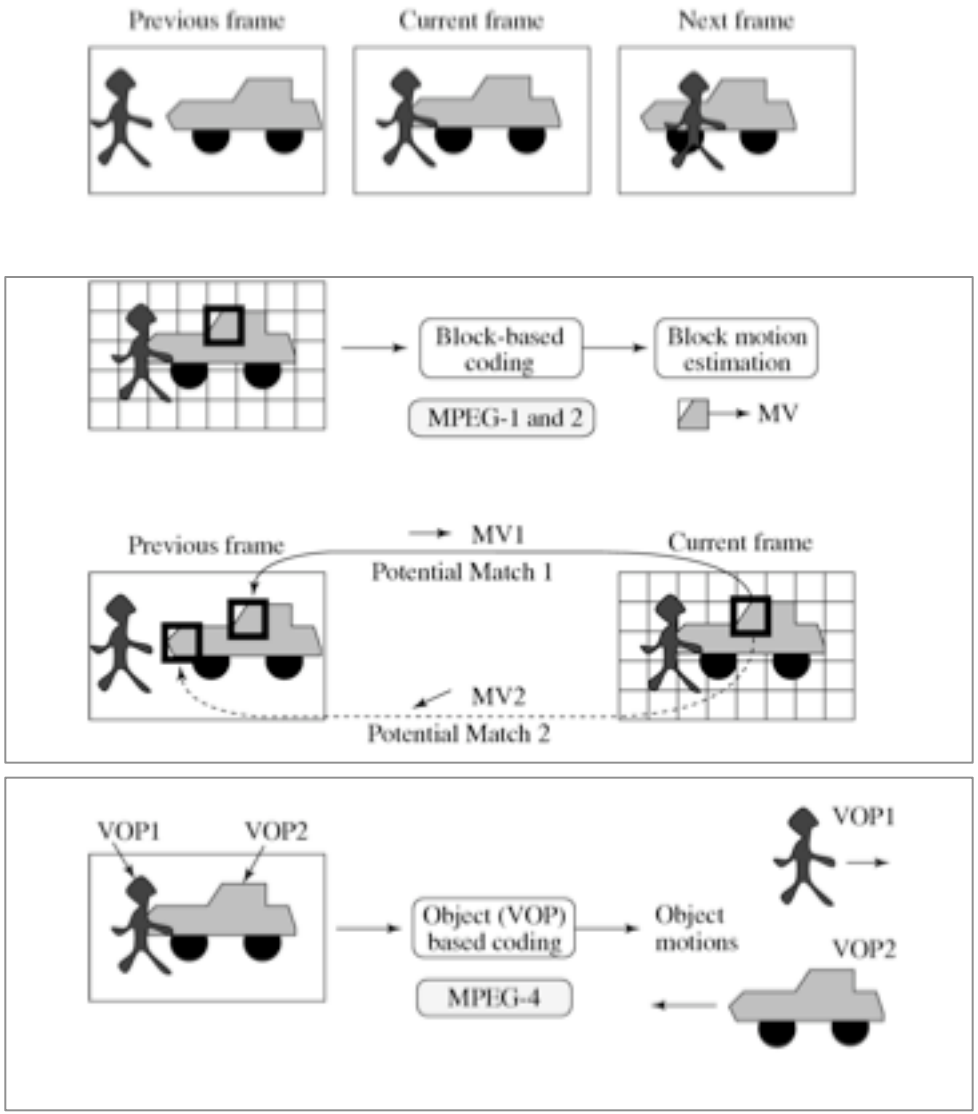
- *Shape coding*
  - Shape coding is still based on blocks. The object bounding box is used for shape encoding. It is eventually divided in 16x16 macroblocks. Shape can be encoded as 8 bit alpha channel or bitmask
  - Macroblocks inside object must be treated differently than boundary blocks (padding, different DCT etc)
  - Algorithms to detect the object shape are not defined (only the bitstream is defined); there can be used several algorithms (either automatic or assisted)

- *Texture coding*
  - Texture coding based on motion compensation and 8x8 DCT standard or shape adaptive





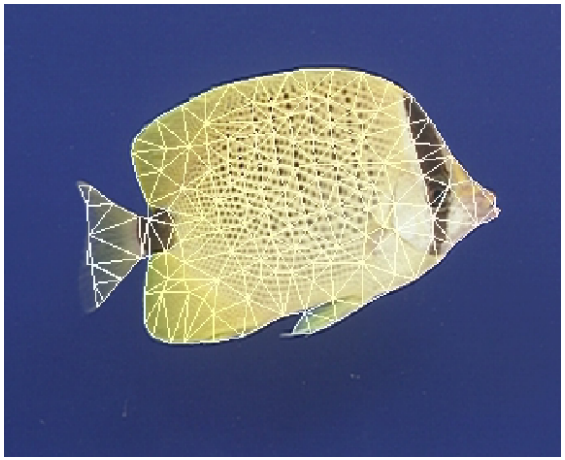
Comparison between block-based and object-based coding

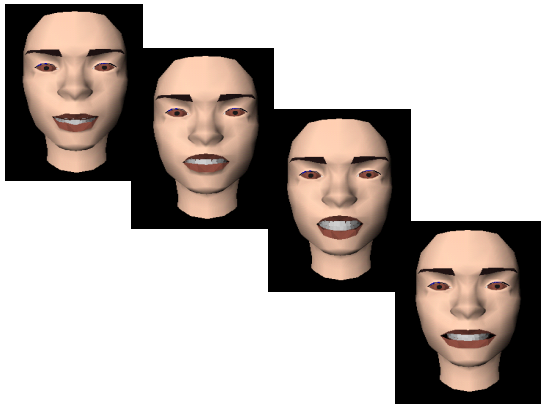


Object-based coding: synthetic 3D Audio Visual Objects

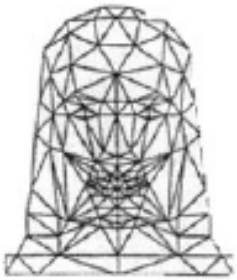
- MPEG4 supports coding of 3D synthetic audiovisual objects:
  - Animated faces
  - Animated bodies
  - 2D meshes with animation

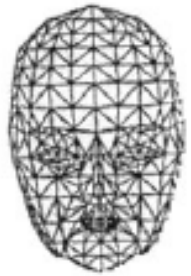
- It has allows special compression algorthms for 3D mesh compression and 2D texture mesh compression









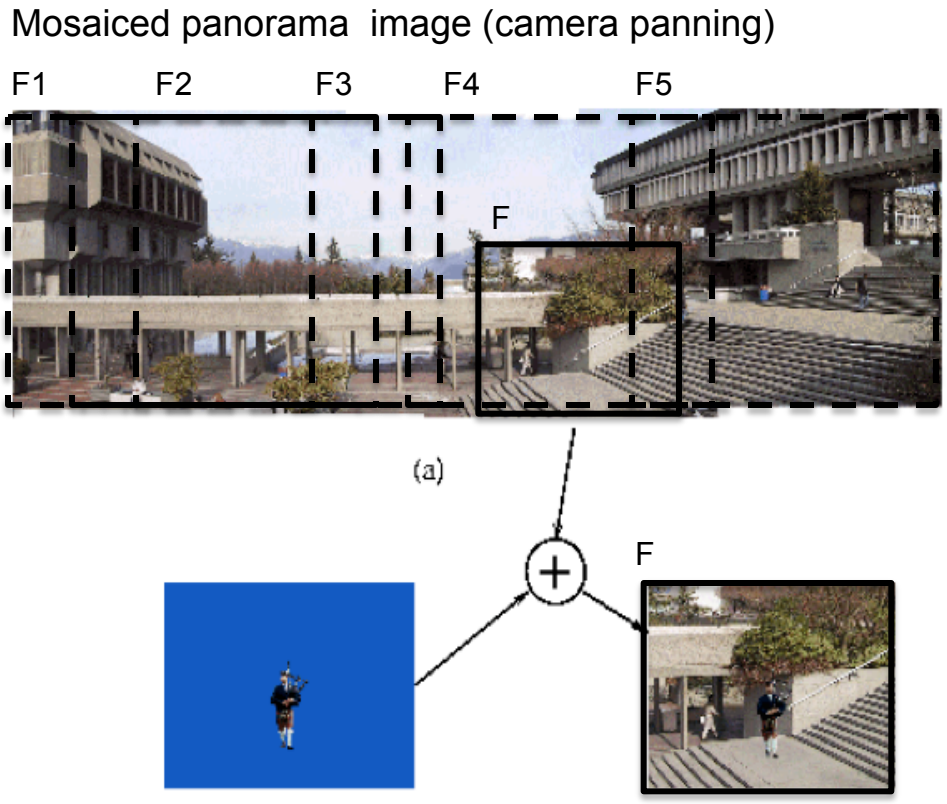


Global motion compensation

- Background objects must be separated from foreground objects: to separate the foreground object from the background, sprite panorama images are considered i.e. a still image that describes the static background over a sequence of video frames.

Sprite panorama is encoded and sent to the decoder only once at the beginning of the video sequence

When the decoder receives foreground objects (separately coded) and parameters of the camera movements, it can reconstruct the scene

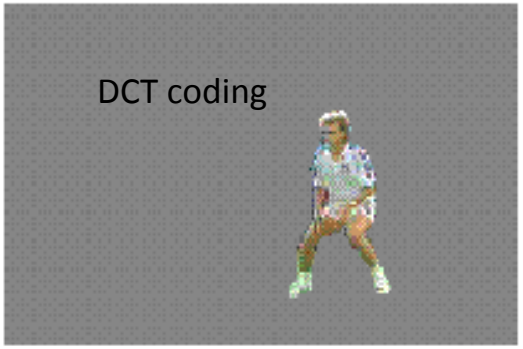


Server side

Global motion compensation for background images

Compression can be adapted for each object





Client side



MPEG4 Improvements

- Improvements in coding are obtained with appropriate object based motion prediction.
  - Compression can be adapted for each object
  - Motion compensation with ¼ di pixel interpolation for objects
  - Global motion compensation for background images
  - B-VOP motion prediction
  - DCT coding (as MPEG2 or with a different quantization)
  - Wavelet coding of images and textures that are applied to meshes

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…

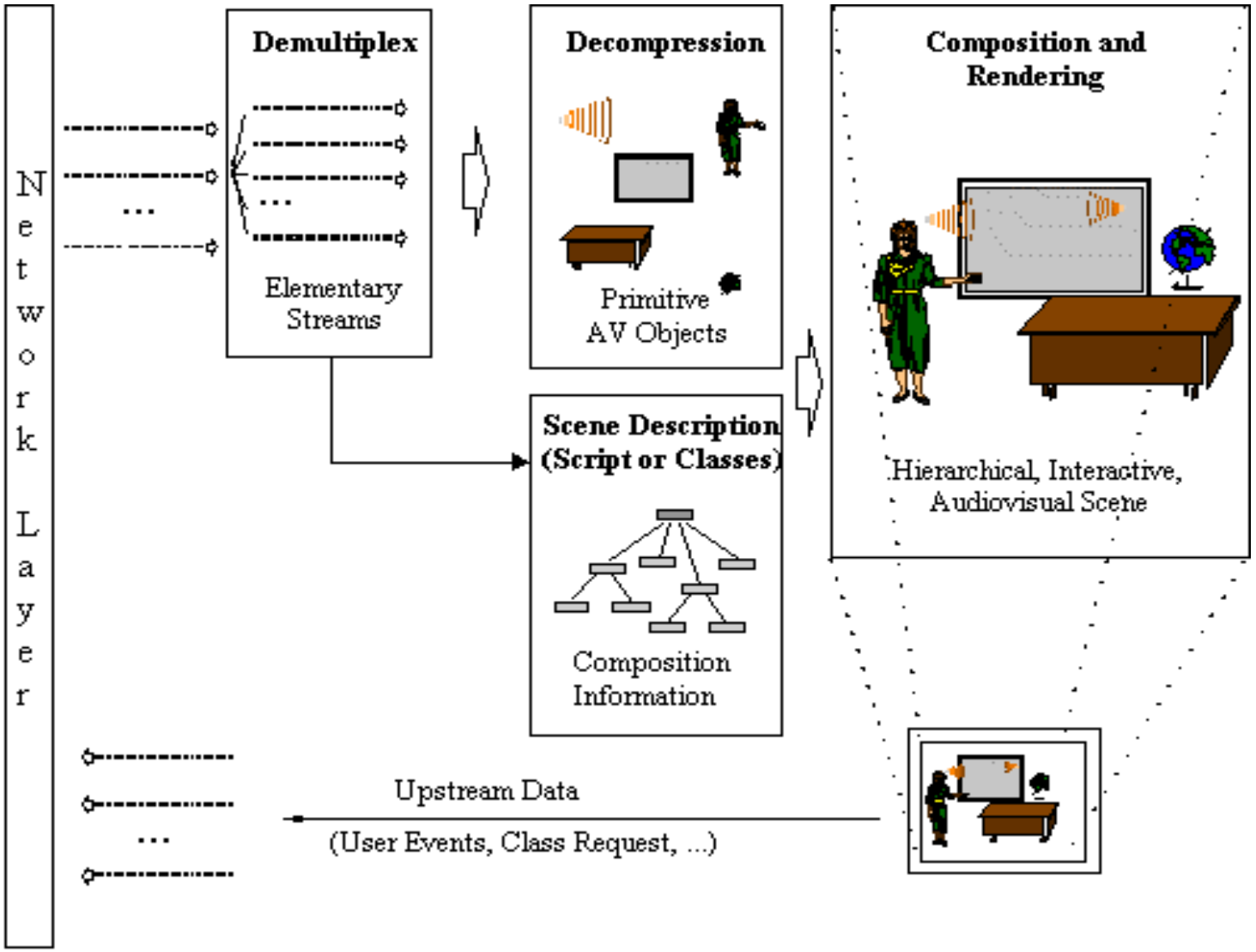
- Levels define different degrees of computational complexity and quality



Profiles@levels

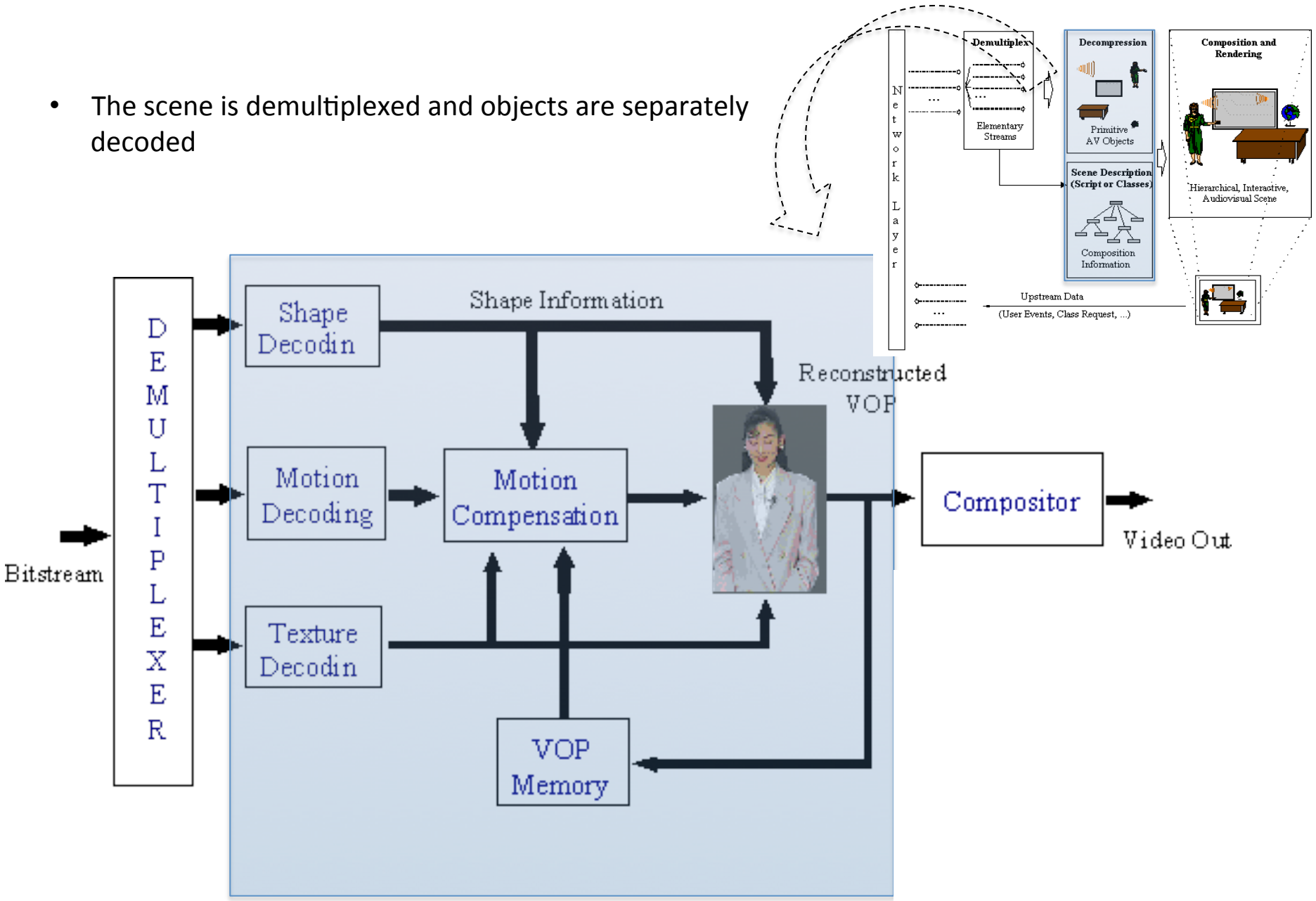
Profile	Level	Typical picture size	Bit-rate (bits/sec)	Max number of objects
Simple	1	176 × 144 (QCIF)	64 k	4
	2	352 × 288 (CIF)	128 k	4
	3	352 × 288 (CIF)	384 k	4
Core	1	176 × 144 (QCIF)	384 k	4
	2	352 × 288 (CIF)	2 M	16
Main	1	352 × 288 (CIF)	2 M	16
	2	720 × 576 (CCIR601)	15 M	32
	3	1920 × 1080 (HDTV)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

MPEG4 decoding



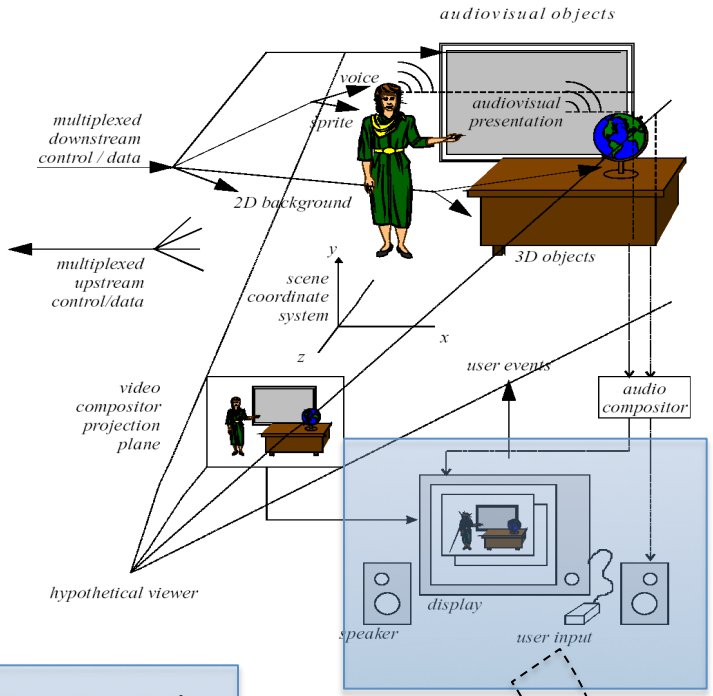
Object decoding (client side)

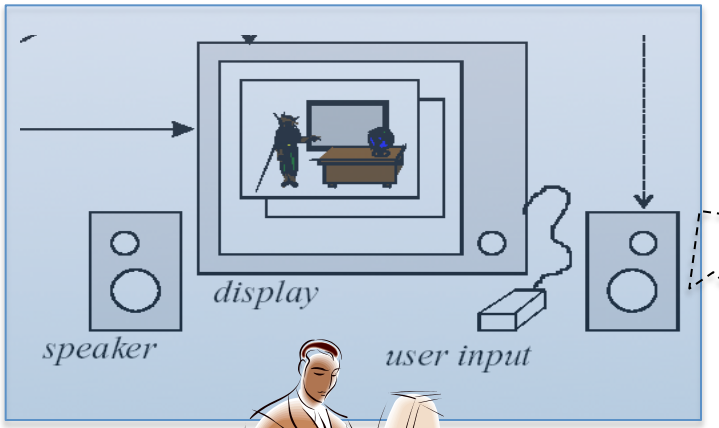
- The scene is demultiplexed and objects are separately decoded



Interactive display of MPEG4 scene (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





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.