Part II – Video

- General Concepts
- MPEG1 encoding
- MPEG2 encoding
- MPEG4 encoding
- H.264 encoding
- H265 encoding

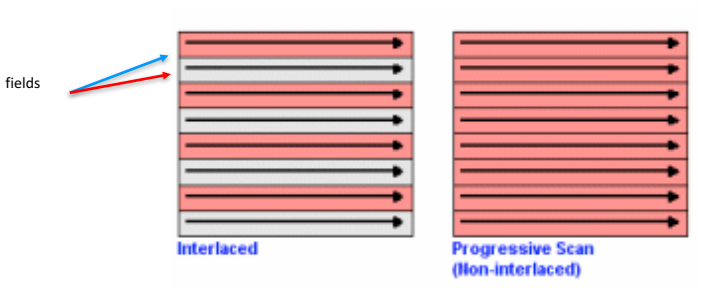
Video General Concepts

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

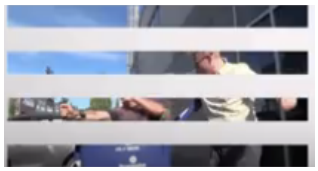
- Digital video cameras come in two different image capture formats: interlaced and progressive scan
  - *Interlaced cameras* record the image in alternating sets of lines: the odd-numbered lines first, and then the even-numbered lines. One set of odd or even lines is referred to as a *field*, and a 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.



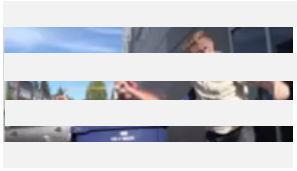
Progressive and interlaced video

- Video comes into two forms:
  - Progressive: one image after another is captured transmitted and played in rapid succession. It is specified by [format p framerate] (es. 1080p 30)
  - Interlaced: alternate sets of lines are captured transmitted and played : first even lines and then odd lines. Half-frames are called fields. It is specified as [format i half-framerate] (es 1080i 60)
- This distinction was originated by the broadcasting industry mainly due to bandwidth restriction of cable and satellites

1080p



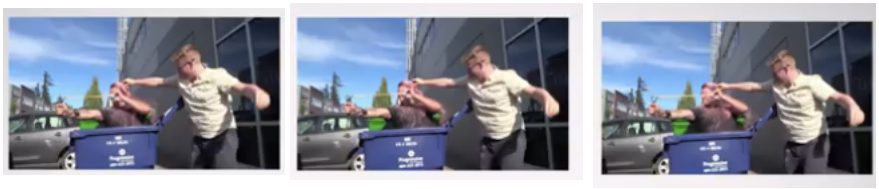
1080i even lines



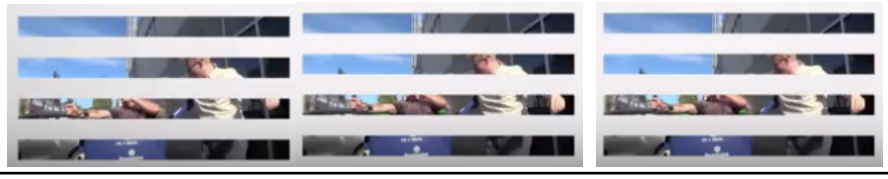
1080i odd lines

- For a fixed bandwidth, interlaced video has twice the display refresh rate versus progressive scan video at a similar frame rate (for instance 1080i at 60 half-frames per second, vs. 1080p at 30 full frames per second)
  - The higher refresh rate improves the appearance of an object in motion, because it updates its position on the display more often.
  - When an object is still, persistence of human vision combines information from multiple similar half-frames to produce the same perceived resolution as a progressive full frame

30 full frames per second



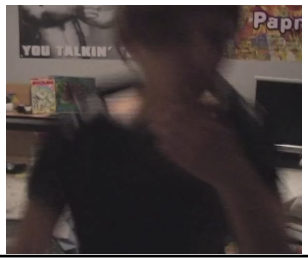
60 half frames per second



Today

- With digital video compression, interlacing introduces inefficiencies:
  - Because each interlaced video frame is two fields captured at different times, frames can exhibit motion artifacts known as *interlacing effects*, or *combing*, if recorded objects move fast enough to be in different positions when each individual field is captured
  - when the subject contains vertical detail that approaches the horizontal resolution of the video format (a finely striped jacket on a news anchor) there can be *interline twitter* like a shimmering effect

- LCD monitors and flat panels TV support progressive display. To display interlaced video on a progressive display requires a process called deinterlacing. EBU European Broadcasting Union promoted HDTV 1080p 50 progressive to replace 1080i 50
- Most modern computer monitors do not support interlaced video. Playing back interlaced video on a computer display requires some form of deinterlacing in the software player

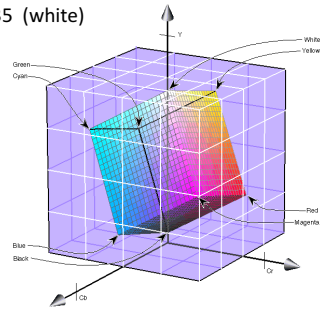


Digital video color spaces

- 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

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

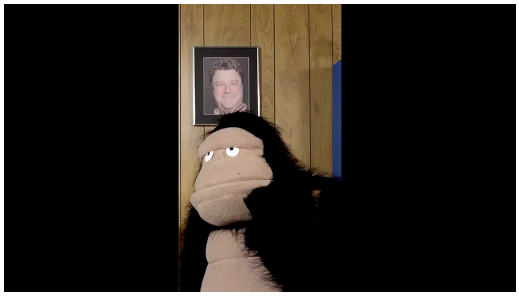


Digital video aspect ratio

- 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.



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 FULL HD quality
  - 50 Mbit/s 4K ULTRA HD quality
- Assuming that the viewer has the download rate and processing speed to view something at the same bit rate, bitrate means video quality

Constant and variable video bitrate

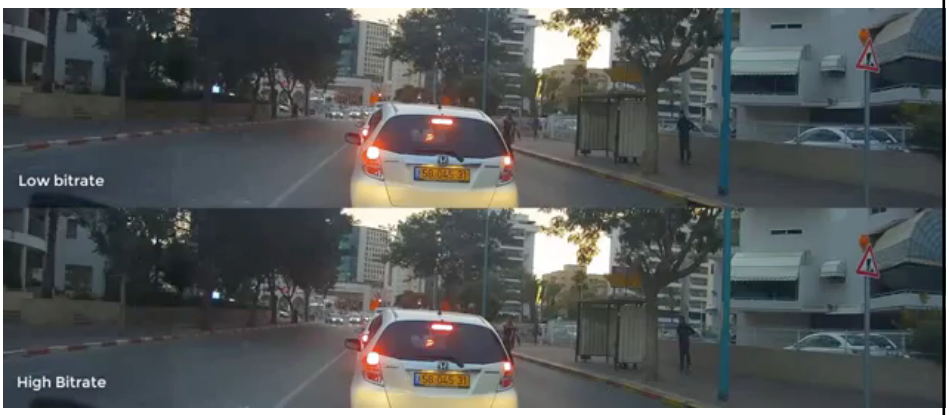
- Video can be transmitted at Constant or Variable bitrate
  - *Constant bitrate* (CBR) maintains the same bit rate of transmission for the entire video. CBR can produce uneven quality. The variation in data density is handled by compressing the more intense parts of the video more than the simpler parts.

    For real-time and non-buffered video streaming when the available bandwidth is fixed – e.g for videoconferences, satellite and cable broadcasting – CBR must be used

  - *Variable bitrate* (VBR) is a strategy to maximize the visual video quality and minimize the bitrate. On fast-motion scenes, a variable bitrate uses more bits than it does on slow-motion scenes of similar duration, yet achieves a consistent visual quality .

    VBR is commonly used for video DVD/Blu ray creation and video in programs.
    VBR is the better choice for live streaming and for streaming video in general.

- Every codec can give a varying degree of quality for a given set of frames within a video sequence. Numerous factors play a role in this variability.
  - First, the bitrate control mechanism that is responsible for determining the bitrate and quality on a per-frame basis: *variable bitrate* (VBR) and *constant bitrate* (CBR) create a trade-off between a consistent quality over all frames and a more constant bitrate
  - Second, codecs differentiate between different types of frames, such as key frames and non-key frames, differing in their importance to overall visual quality and the extent to which they can be compressed.
  - Third, quality depends on prefiltrations, which are included on all present-day codecs.



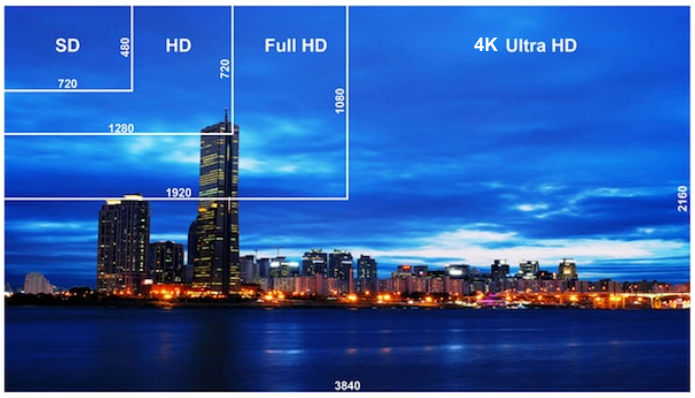
Video bitrate, resolution, compression

- There's a reciprocal relationship between bit rate and resolution: resolution is the density of the video itself, while bitrate has to do with its transmission. A higher bitrate allows transmission of higher-resolution video without as much loss of data.
- Transmitting video over the Internet always requires a certain amount of file compression. The higher the video resolution and the lower the bitrate, the more compression is required and the more data will be lost
- A high video resolution with a low bitrate can actually result in a poorer quality video at the viewing end, compared to a lower resolution video

Resolution

4K (Ultra HD) is going to be the standard for video production. Google, YouTube, Vimeo support upload of 4K video

4K video can be useful when we need a crop in high resolution of a large image. But it requires wideband transmission

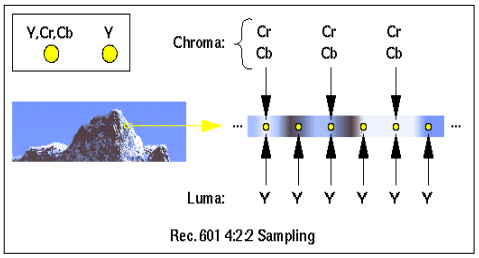


Video sampling

- 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)

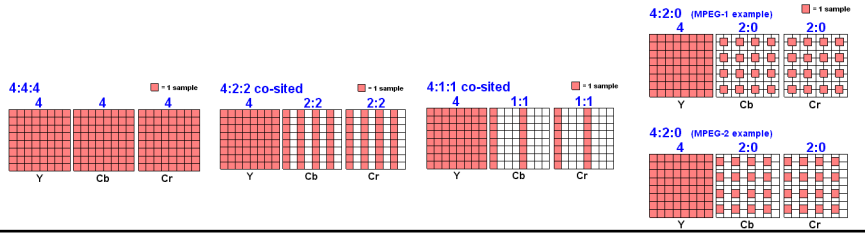
  Es. 4:2:2 means that every 4 samples of luminance, there are 2 chroma samples both in the odd and the even lines. It compresses frames as it drops data. 4:2:0 provides higher compression

For each line



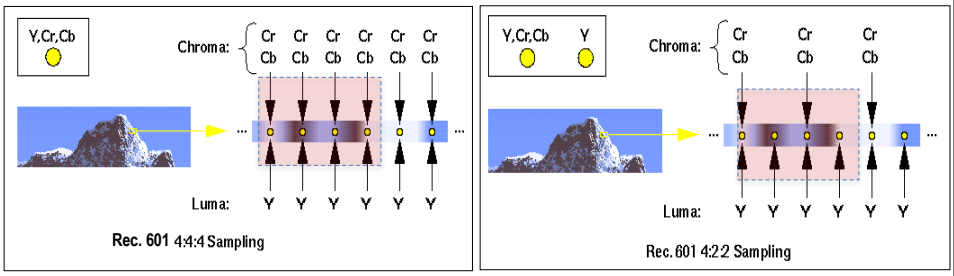
Digital video sampling rates

- 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.



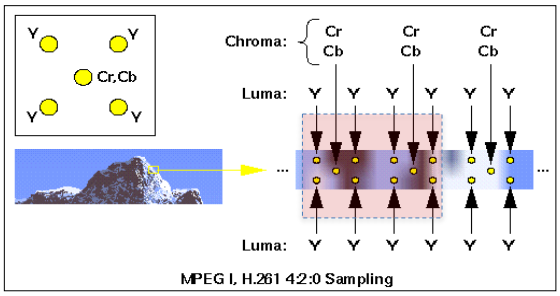
Digital video format ITU-R BT.601

- 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



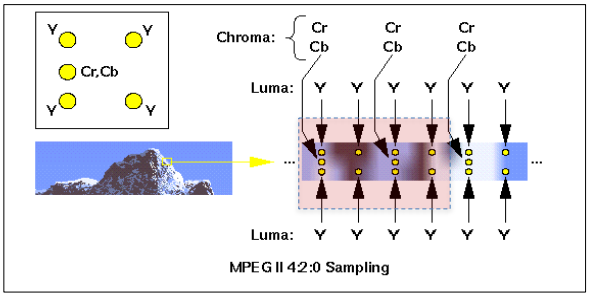
Digital video format MPEG 1

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



Digital video format MPEG 2

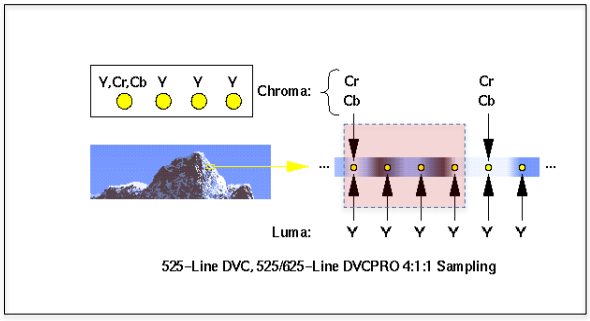
- MPEG2 was defined to provide a better resolution than MPEG1 and manage interlaced data. Based on fields instead of frames. Used for DVD and HDTV:
  - 4:2:0 sampling
  - Bitrate 5 Mbit/s.
  - Frame size: 720x480



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 (4:2:0 – 4:2:2)
  - with JPEG compression we reduce to 1-8 bpp
  - with MPEG we go below 1 bpp

Codecs and containers

- A codec – or coder/decoder – is an encoding tool that processes video and stores it in a stream of bytes. Codecs use algorithms to effectively shrink the size of the audio or video file, and then decompress it when needed.
- A container has the purpose of bundling all of the audio, video, and codec files into one organized package. The container often contains chapter information for DVD or Blu-ray movies, metadata, subtitles, and/or additional audio files such as different spoken languages.

Codecs history

- Codecs standards have been originated by two experts groups: the MPEG (Moving Pictures Experts Group) and the ITU-T VCEG (Video Coding Expert Group)
- MPEG and VCEG, work side-by-side. MPEG specialized in broadcast (television), while the VCEG focused on telecommunications (phone, internet). Today their goals mostly overlap as everything is going the way of the Internet and this association is likely to continue.
- In 1988, H.261 was created by the VCEG, mainly for ISDN/Videoconferencing work. It had a maximum bit rate of 2 Mbps, and was limited with a chroma sub-sampling of 4:2:0.
- In the same time JPEG became a popular codec for images.
- In 1993 the MPEG following the broadcast industry's needs adopted H.261 and JPEG and defined the MPEG-1 suite. It was limited to 1.5 Mbps, 4:2:0 and stereo audio only.
  At the time, there was PAL, NTSC and VHS. MPEG suites have sub-divisions, called Parts.
  Traditionally, Part 1 is for the System (file format). Part 2 is for video, and Part 3 is for audio.

Progress of MPEG standard

The story of two groups - MPEG and VCEG						
Year	**MPEG**	**Part**	**Layer/Profile/Type**	**Usage**	**VCEG**	Variants
1984	Not formed	Practically not useful			H.120	
1988	Not formed	Videoconferencing			H.261	
1993	MPEG-1	VHS and Television Recording				
		Part 1	Systems			
		Part 2	Video	VCD	H.261	
		Part 3	Audio			
			Layer 1			
			Layer II			
			Layer III	**MP3**		
1999	**MPEG-2**	Broadcast, Distribution, DVD				
		Part 1	Systems			
			Program Stream			
			Transport Stream			
		Part 2	Video		H.262	HDV, XDCAM
		Part 3	Audio			
			Layer 1			
			Layer II			
			Layer III	**MP3**		
2004	**MPEG-4**	Broadcast, Internet, Blu-ray				
		Part 1	Systems			
		Part 2	Video		H.263	HDCAM SR
		Part 3	Audio			
		Part 10	Advanced Video Coding	**MPEG-4 AVC**	**H.264**	AVCHD, XAVC
		Part 14	MP4 Container	**MP4**		
2013	MPEG-H	Part 2	Video	**HEVC**	**H.265**	

MPEG-1 H261, MPEG-2 H262, MPEG-4 H264

- MPEG-1 H261 No more in use
  - VHS Quality at 1.5 MBits/s
  - Basis of Video-CD
  - MP3 audio
- MPEG-2 H262 Still most widely used codec today for broadcast
  - designed for broadcasting and storage
  - Bitrates: 4-9 MBits/s
  - Satellite TV, DVD
- MPEG-4 part 10 H264 The most widely used codec today
  - encoding method for devices with limited resources (media players and mobile phones)
  - Internet and very low-bitrate transmission: 4kbps to 240Mbps (Part 10/H.264)
  - Drives the Blu-ray disc
  - Coding of media objects

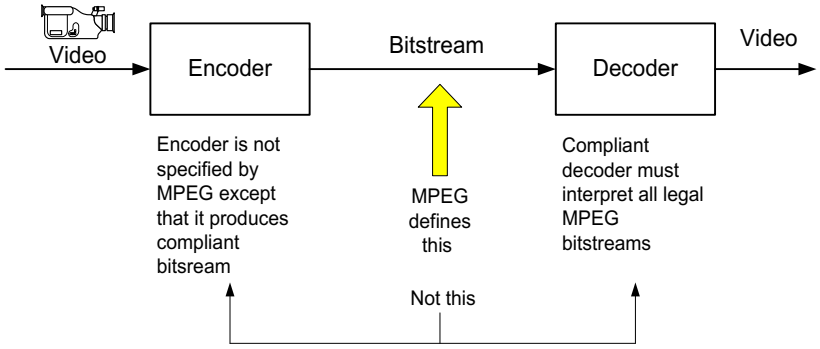
MPEG-H H265

- MPEG-H H.265 or HEVC (High Efficiency Video Coding). Clearly the future
  Key standard for videosurveillance with Full HD e 4K cameras. There are many vendors already claiming compatibility with this format.
- Key features:
  - Support up to ULTRA HD 8K (7680 × 4320 pixel - total 33,2 Mpixel).
  - Bitrate 65Mps
  - Supports up to 300 fps
  - 12-bit color bit depth
  - 4:4:4 and 4:2:2 chroma sub-sampling
  - File size subjectively half the size of H.264 with better quality

Video coding standard	Average bit rate reduction compared to H.264/MPEG-4 AVC HP			
	480p	720p	1080p	4K UHD
HEVC	52%	56%	62%	64%

Codecs: 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



Containers

- Most popular Containers:
  - **MKV** (open source)
    MKV is a rapidly growing format. The container itself supports almost any audio or video format. However, they are not often supported from editing software and on some operating systems an extra playback program is required.
  - **MP4**
    MP4 is the recommended format for uploading video to the web, and services such as Vimeo and YouTube have it listed as their preferred format. The MP4 container utilizes MPEG-4 encoding, or H.264, as well as AAC or AC3 for audio. It is widely supported on most consumer devices, and is the most common container used for online video.
  - **AVI**, **WMV**, **ASF** (Microsoft)
    The **AVI** container is probably the best known one, but is also very old and cannot handle modern codecs like H. 264. **WMV** and **ASF** containers are not very flexible. If the video is played on a non-Windows system it usually needs an extra video player.
  - **MOV** (Apple)
    The **MOV** container supports about all codecs, but some operating systems require an extra video player
  - **FLV**, **SWF** (Flash Video) Flash is an aged container

MPEG-1 H261

MPEG-1

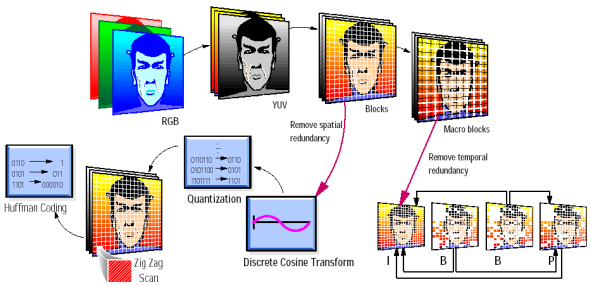
- MPEG-1 is an ISO standard (ISO/IEC 11172) developed to support VHS quality video.
  It was developed for *progressive video* (non interlaced) so it manages only frames.
- The MPEG1 typical video resolution is 352x240 or 320x240 with 4:2:0 sampling at a bitrate of ~1.5 Mbps

- 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

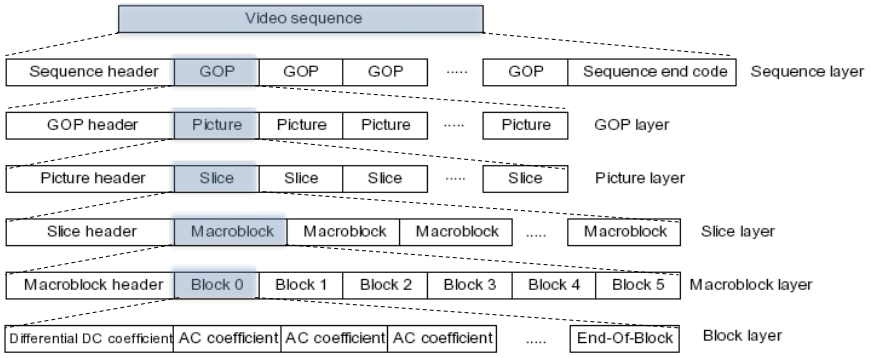
Basic principles

- MPEG-1 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-1
  - *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)



6 layers

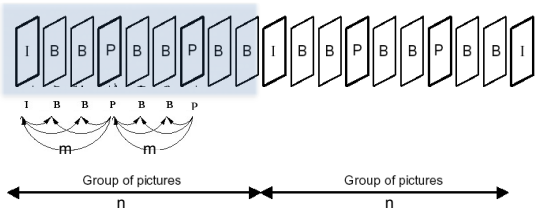
- Sequence: unit for random access
- GOP: unit for video random access (the smallest unit of independent coding)
- Picture (frame): primary coding unit
- Slice: syncronizzation unit
- Macroblock: motion compensation unit
- Block: unit for DCT processing



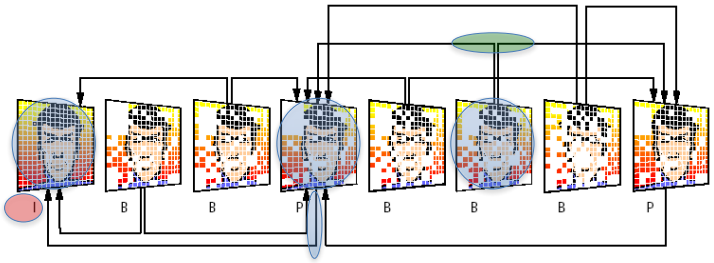
GOP

- 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:
  - **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



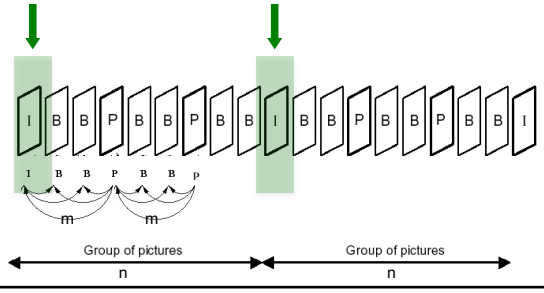
Frames



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

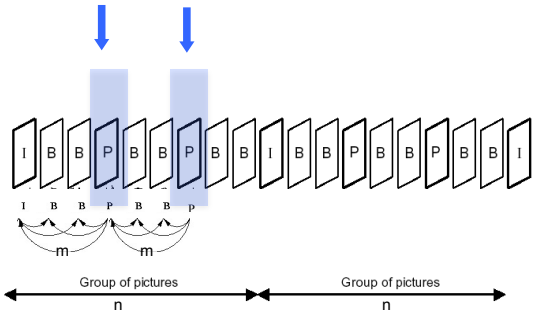
I-frames

- *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).



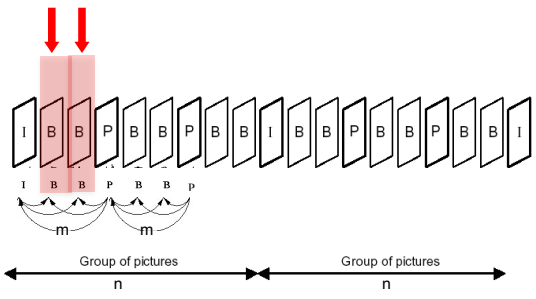
P-frames

- *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

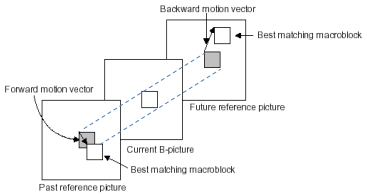
- *Bi-directional-Coded frames* are coded with *bidirectional (past and future) motion compensation* using I and P frame (no B frame).
- Motion is inferred by averaging past and future predictions. B pictures are expected to provide relatively the largest amount of compression under favorable predict
- 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. Not frequently used.

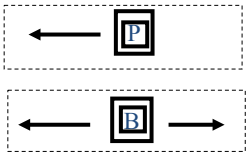


Macroblocks

- Each video *frame* contains *macroblocks* that are the processing unit of MPEG-1 compression. Macroblocks are set of 16x16 pixel and are necessary for purposes of the calculation of motion vectors and error blocks for motion compensation.

- individual prediction types can be selected on a macroblock basis rather than being the same for the entire picture 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)

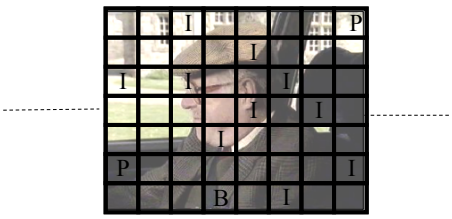




Frames and macroblocks

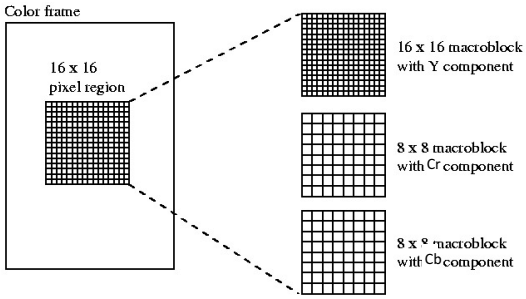
- Therefore frames can contain different types of macroblocks:
  - P frames: contain *Intra-coded* (I) *macroblocks* or *forward-predicted* (P) *macroblocks*
  - B frames: contain *Intra-coded* (I) *, forward* (P) or *backward-predicted* (B) *macroblocks*
  - I and D frames: contain *Intra-coded* (I) *macroblocks* with blocks that contain direct encoding from the image samples

B Frame with macroblocks



Macroblock components

- Each macroblock is encoded separately.

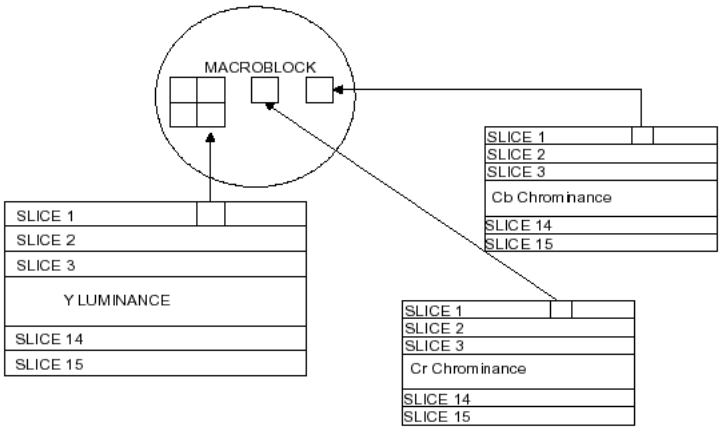


4:2:0 sampling

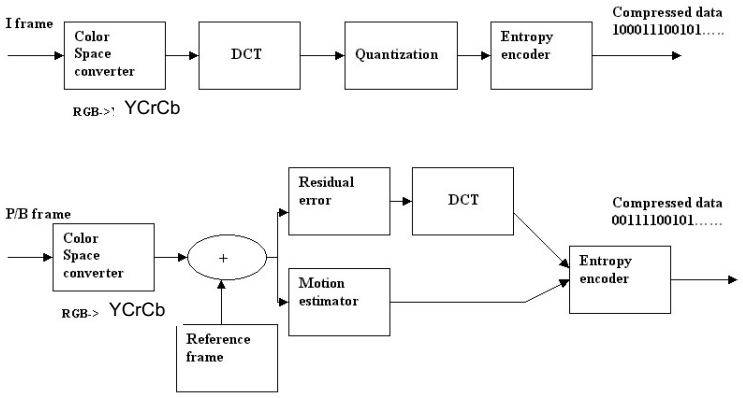
- The Y component of a macroblock is used for motion compensation. Cr and Cb are chrominance components.

Slices

- Macroblocks are organized into slices

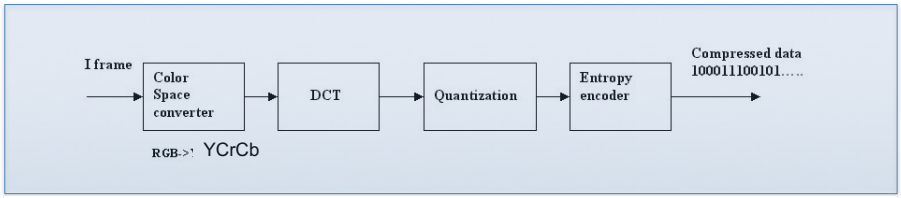


Encoding macroblocks

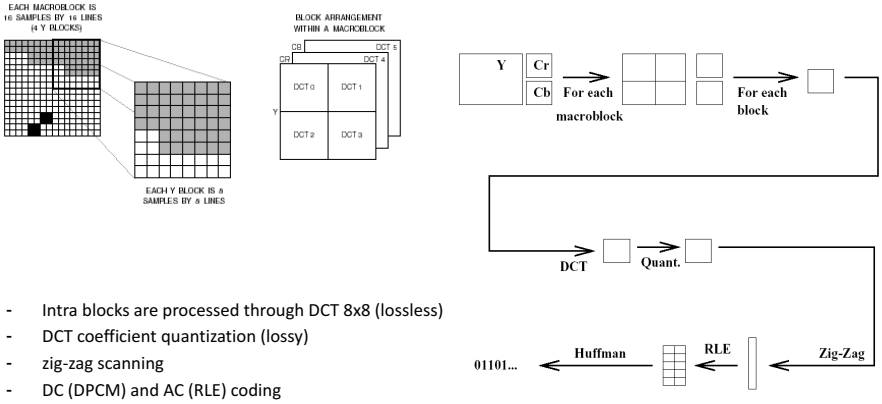


The block diagram of the MPEG encoder

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)

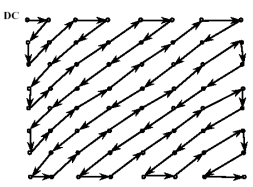
Spatially-Adaptive Quantization

- Spatially-adaptive quantization is made possible by the *factor quantizerscale*. 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.

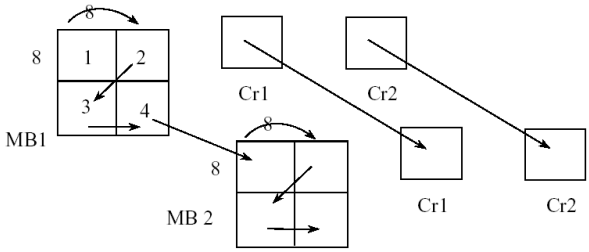
8	16	19	22	26	27	29	34
16	16	22	24	27	29	34	37
19	22	26	27	29	34	34	38
22	22	26	27	29	34	37	40
22	26	27	29	32	35	40	48
26	27	29	32	35	40	48	58
26	27	29	34	38	46	56	69
27	29	35	38	46	56	69	83

MPEG1 default quantization matrix

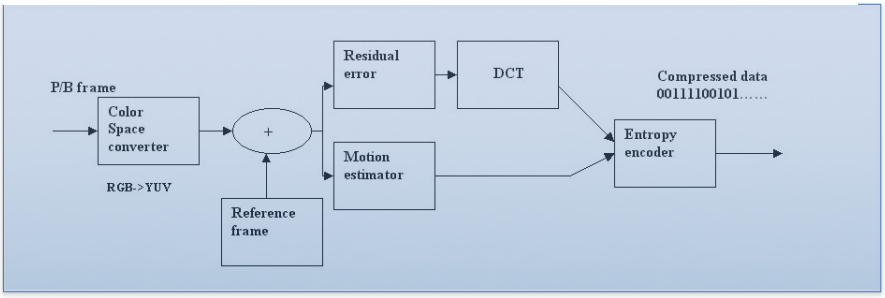
- zig-zag scanning is used to create a 1D stream



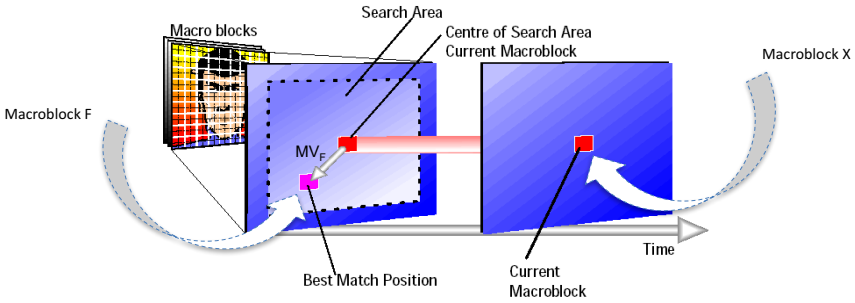
- AC coefficients are encoded losslessly according to *run length encoding* and Huffman coding. *Tables* are formed on a statistical *basis.* Different tables for Y and CbCr.
- DC coefficients encode differences between blocks of the macroblock:



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.

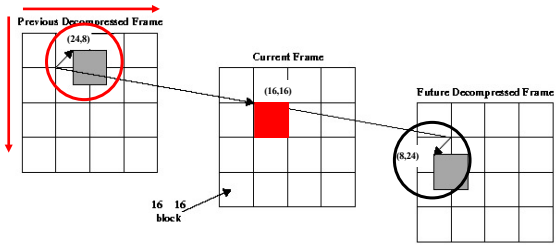
Motion vectors

- 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): in this case the predictive base is the same macroblock in the reference frame
- Motion vectors are reset when a new I macroblock is found.

Example:

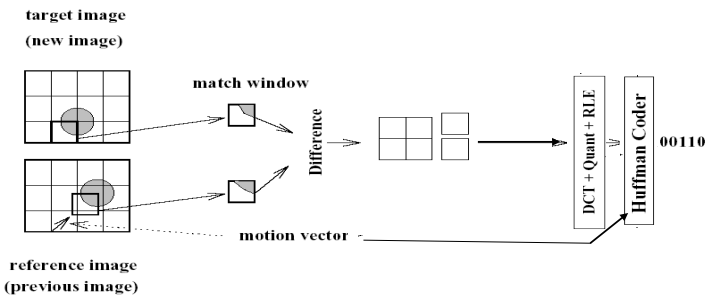
the match of the shaded macroblock of the current frame in the previous frame is in position (24,8). The forward predicted motion vector for the current frame is (8,- 8)

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)

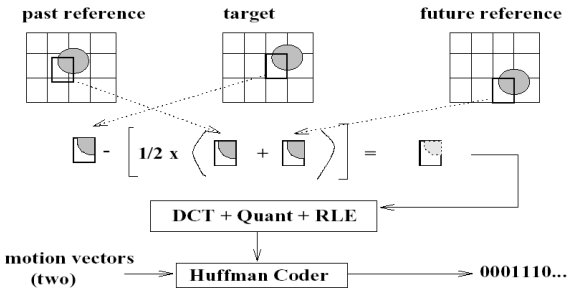


Error blocks

- 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:

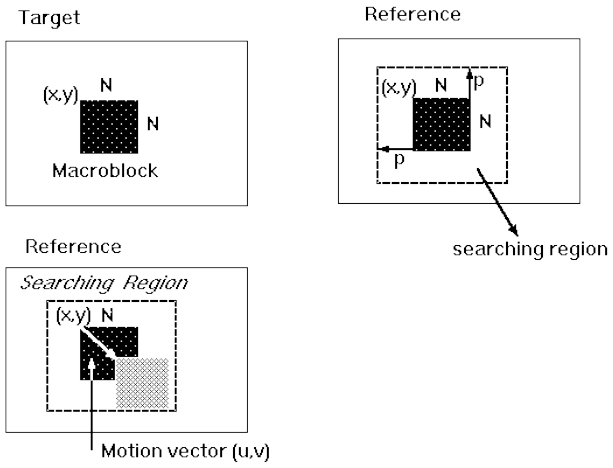


- For a B macroblock:



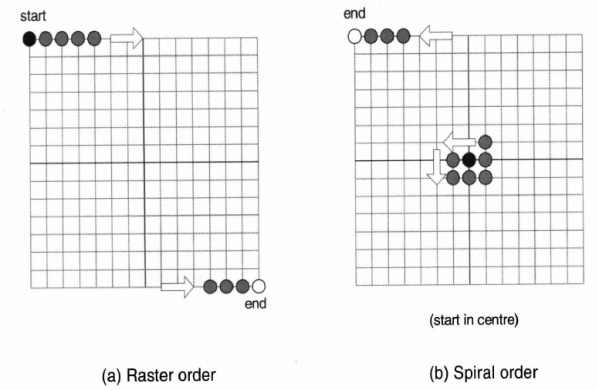
Motion estimation by block matching

- 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



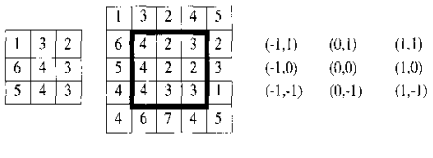
Mean Squared Error (MSE) criterion

- 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

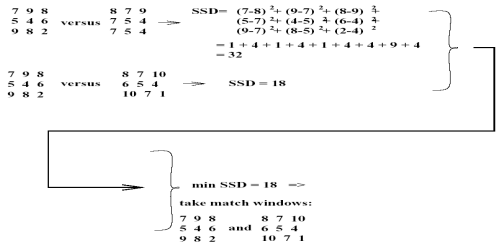
- Mean absolute error/difference (MAE/MAD): Easier wrt MSE

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

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

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

- Sum of Squared Differences (SSD): $SSD = \sum_i (x_i - y_i)^2$ Sensitive to outliers



- 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:

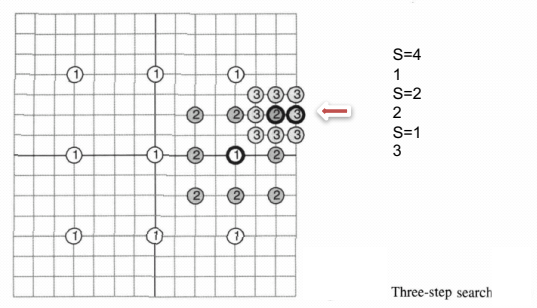
```
7 9 8          8 7 10
5 4 6  versus  6 5 4      -> SAD = 211
9 8 2          10 7 202
```

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

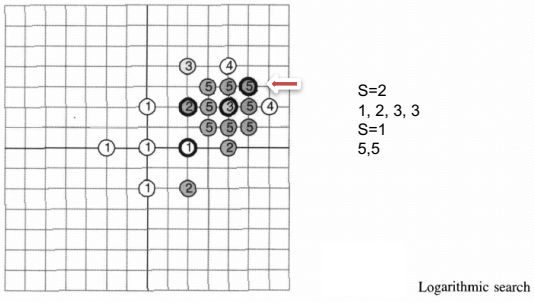
Three step search

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.



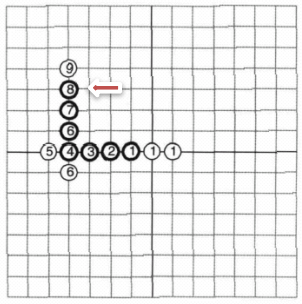
Logarithmic search

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



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.



Horiz
1, 2, 3, 4, 4
Vert
6,7,8,8

One-at-a-time search

Block matching algorithms comparison

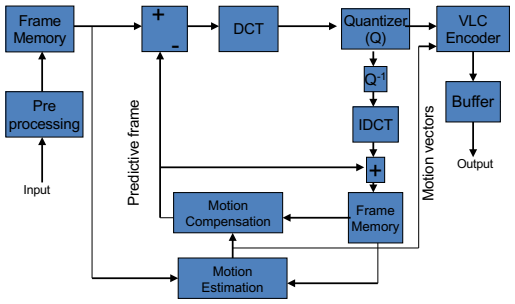
- Logarithmic search, Three step search e one-at-a-time have low computational complexity and low matching performance as well.

MPEG-1 encoding – decoding

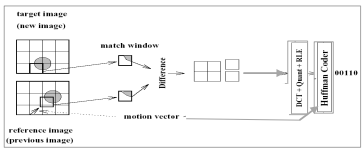
- In MPEG-1 pictures are coded and decoded in a different order than they are displayed. This is due to bidirectional prediction for B pictures. The encoder needs to reorder pictures because B-frames always arrive late.

- 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)

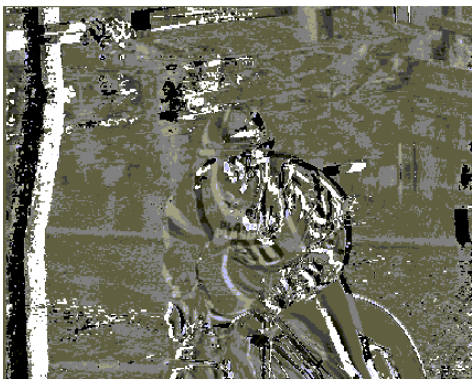
The MPEG-1 encoder

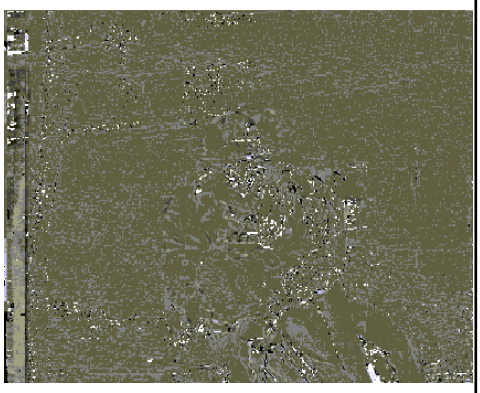


P macroblock



- 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)

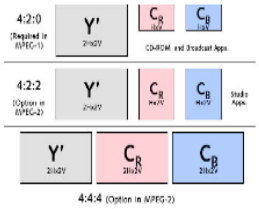
MPEG-2 H262

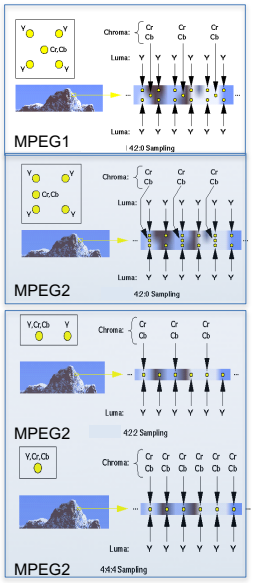
MPEG-2 standard

- MPEG-2 was designed as a superset of MPEG1 with support for broadcast video with 720x576 resolution.at 4-15 Mbps (average 8 Mbps), HDTV, 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
- MPEG-2 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 MPEG-1
  - Flexible quantization schemes (can be changed at picture level)
  - Scalable bit-streams
  - Profiles and levels

Color subsampling

- MPEG-2 supports different color subsamplings:
  - 4:2:0 (as MPEG-1)
    In MPEG-1 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

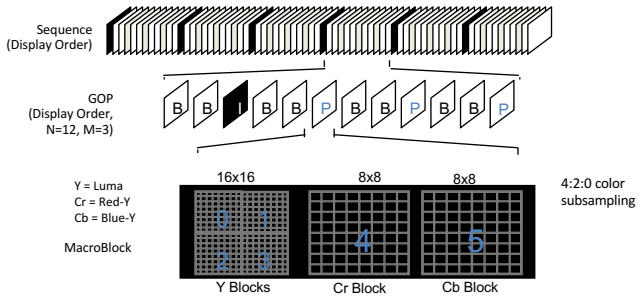




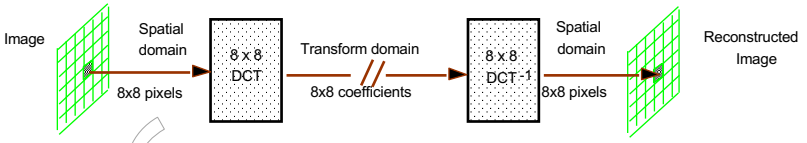
I, P, B frame encoding

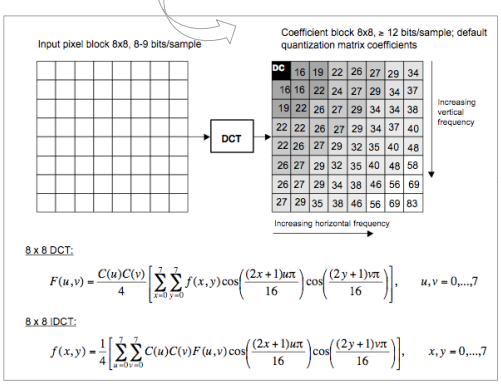
- Same as MPEG-1: I, P and B frames (pictures) are encoded on a macroblock basis using DCT:
  - P-pictures have interframe predictive coding:
    - Macroblocks may be:
      - coded with forward prediction from previous I and P pictures
      - intra coded
    - For each macroblock the motion estimator 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 MPEG-1 it has no D pictures

The MPEG-2 stream



Discrete Cosine Transform and quantization scale

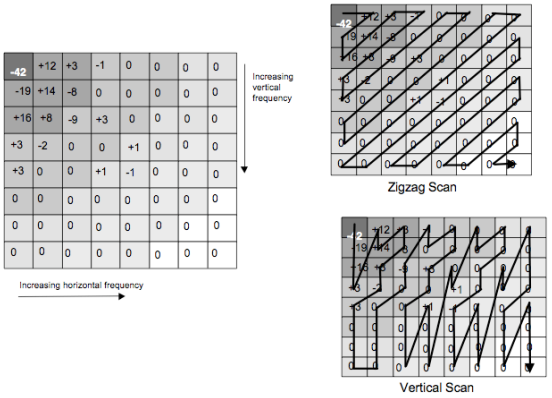




Non-linear quantization scale is also available

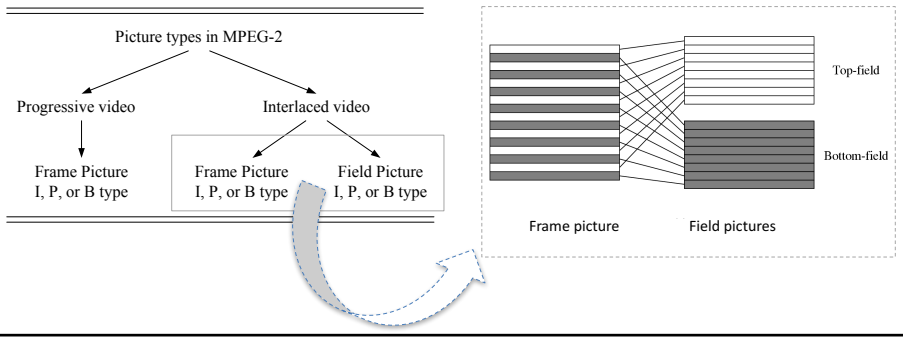
Multiple scanning options

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



Support of progressive and interlaced video

- With MPEG-2:
  - 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 MPEG-1
  - 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.

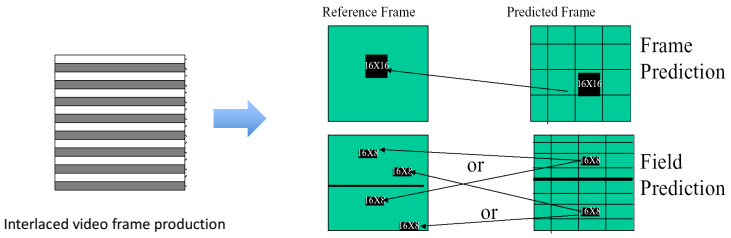


Frame and field-based prediction

- With interlaced video it is possible to choose whether the luminance in the two fields must be encoded jointly or separately. *Frame-based prediction (*for produced frames) or *field-based prediction* (for produced frames and produced fields) are applied on a macroblock-by-macroblock basis:
  - *Frame-based prediction* is suited for macroblocks with little motion and high spatial variations.
  - *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

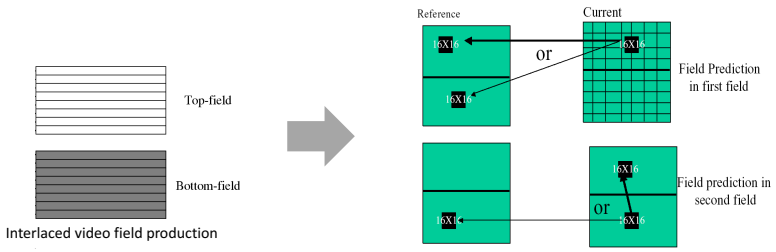
Frame and field prediction for produced frames

- *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.
  - 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.



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.



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.

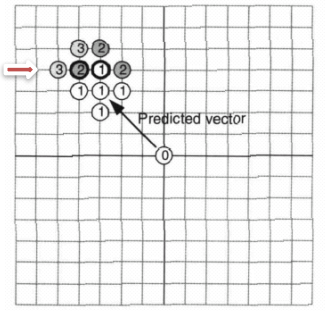
Note that the size of 16 × 16 in the field picture covers a size of 16 × 32 in the frame picture. It is too big size to assume that behavior inside the block is homogeneous. Therefore, 16 × 8 size prediction was introduced in field picture.

Nearest neighbours search

- Used also in H264. 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 (near vectors already coded) 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.

> 1,2,2

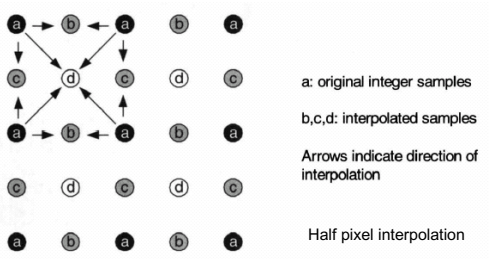


Block matching algorithms comparison

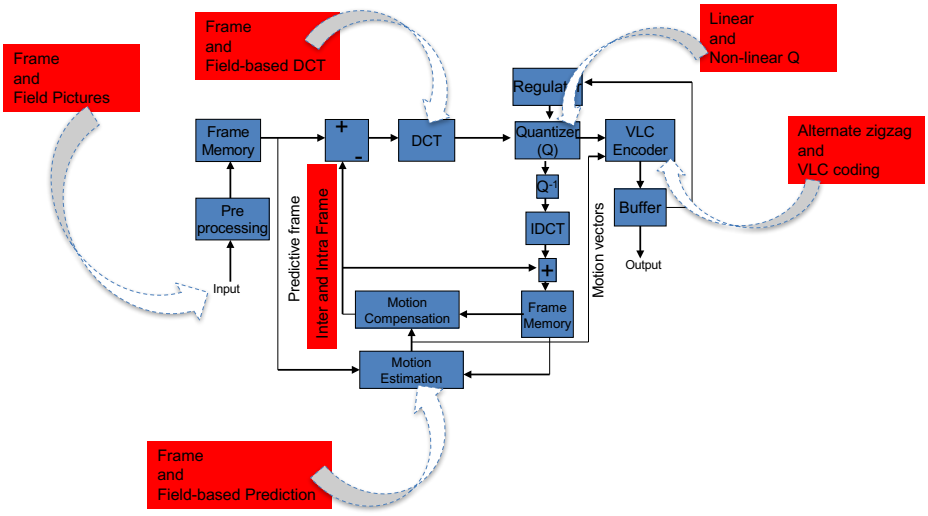
- Logarithmic search, Three step search e one-at-a-time have low computational complexity and low matching performance as well.
- Nearest-neighbours search, has good performance, similar to full search, and moderate computational complexity

Half pixel interpolation for motion estimation

- MPEG-2 uses half-pixel interpolation for motion vector 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:
  - 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 region
  - samples of the best matched region are subtracted from the samples of the current block to obtain the error block



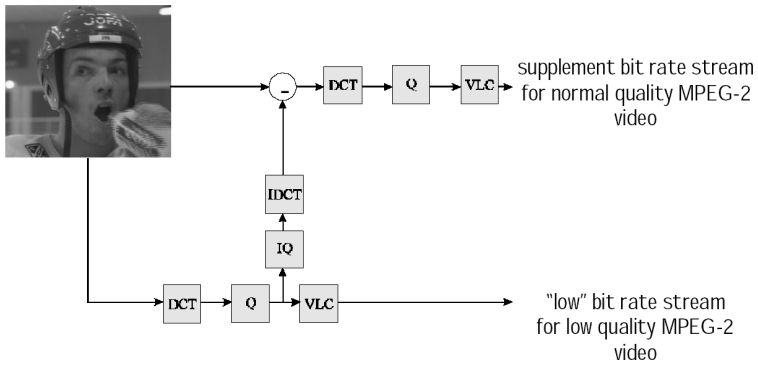
MPEG2 Enhancements



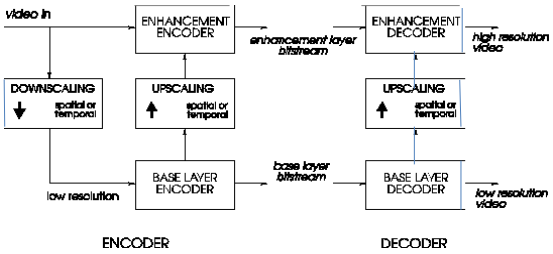
Scalability

- 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 (MPEG-1 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: 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
  - 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 and 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		

MPEG-2 criticals

- There are several conditions that are critical for MPEG-2 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
  - …….

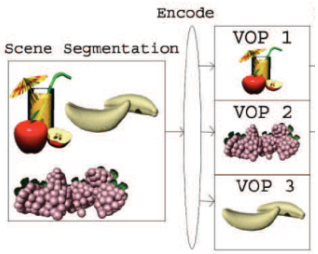
MPEG-4 H264

MPEG-4

- MPEG-4 absorbs many of the features of MPEG-1 and MPEG-2 adding new features such as VRML support for 3D rendering, object-oriented composite files (including audio, video and VRML objects), and various types of interactivity.
- Initially, MPEG-4 was aimed primarily at low bit rate video communications. Its scope as a multimedia coding standard was later expanded. The key parts are MPEG-4 Part 2 (including Advanced Simple Profile) and MPEG-4 part 10 (referred to as H.264)
- MPEG-4 features:
  - Improved coding efficiency over MPEG-2
  - 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)
  - Supports sequences at low spatial resolution and low frame rate (up to 15 fps)
  - Ability to encode mixed media data
  - Ability to interact with the audio-visual scene generated at the receiver

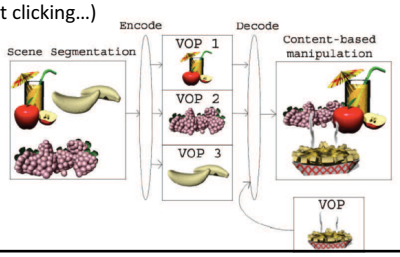
MPEG4 distinguishing elements

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



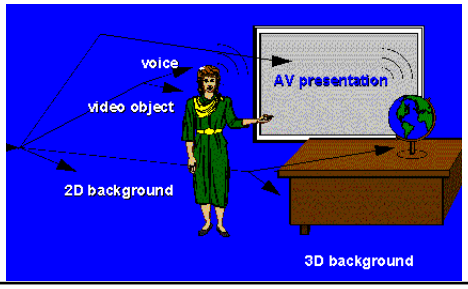
Main features on client and server sides

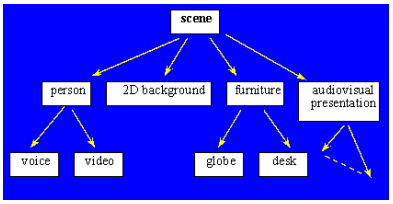
- MPEG4 includes technologies to support:
  - server side
    - Encoding based on and audio-visual objects. When a VOP is the rectangular frame it corresponds to MPEG-2
    - Audio-visual objects manipulation
    - Hierarchical scene composition (audio-visual objects local coordinates, temporal syncronization….. 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



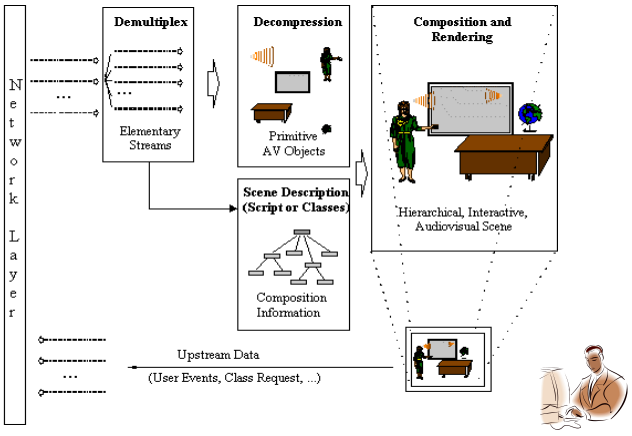


BInary Format for Scene description

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 and interaction (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



Profiles and levels

- Most of MPEG-4 features are optional and their implementation is left to the developer. Most of the software for MPEG-4-coded multimedia files does not support all the features.

- Profiles define resolution, bitrate and number of the objects that can be coded separately
- 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	suited for mobile terminals
	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	suited for internet services
	2	720 × 576 (CCIR601)	15 M	32	
	3	1920 × 1080 (HDTV)	38.4 M	32	

MPEG-4 video compression

- MPEG-4 Part 10 specifies a compression format for video signals which is technically identical to the H264 standard.

- The motivations for H264 standard are that digital representation of the Television signals created many different services for the content delivery (Satellite, Cable TV, Terrestrial Broadcasting, ADSL and Fiber on IP) with contrasting requirements. To optimize these services, there is the need of:
  - -High Quality of Service (QoS)
  - -Low Bit-Rate
  - -Low Power Consumption

  ← ← *Conflicting*

H264

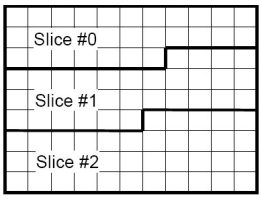
- H264 codec is the standard for:
  - Storage on blu-ray discs
  - Streaming internet (standard for videos on YouTube and iTunes Store)
  - Conversational services over Internet, LAN, wireless and mobile networks,
  - Broadcast services: broadcast satellite television services; cable television services, DSL….

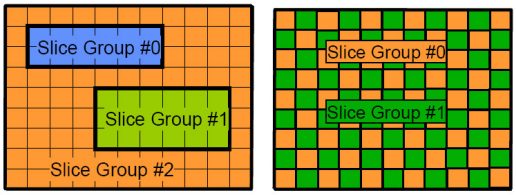
- The H264 standard provides good video quality at a lower bitrate wrt MPEG-2 with no additional cost of implementation or complexity

- H264 is a joint effort between Video Coding Experts Group (VCEG) and Moving Picture Experts Group(MPEG)
- It 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 H264 as the format for QuickTime

Slices and Macroblocks

- In H264 a picture is a collection of one or more Slices. Slices are a sequence of 16x16 Macroblocks which are processed in the order of a raster scan and decoded without use of data from other slices. This flexibility supports variations of the effective bandwidth available to a user
- Slice groups can also be used (Flexible Macroblock Ordering facility).
- H.264 does block based coding. Each 16x16 Macroblock is divided into blocks





Useful for concealment in video conferencing applications

Integer transform

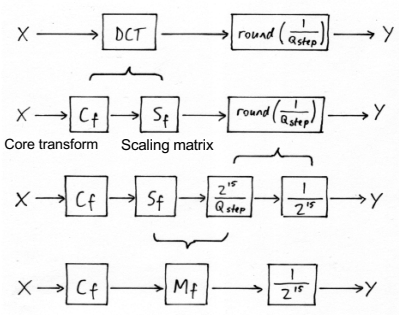
- Sampling structure (same as MPEG-2):
  Y CbCr 4:2:2 ; 4:4:4
- New Transform:
  - similar to DCT: the transform is a scaled approximation to a 4x4 Discrete Cosine Transform that can be computed using simple integer arithmetic

$$\mathbf{A} = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix}, \quad a = \tfrac{1}{2},\ b = \sqrt{\tfrac{1}{2}}\cos\tfrac{\pi}{8} = 0.6532\ldots,\ c = \sqrt{\tfrac{1}{2}}\cos\tfrac{3\pi}{8} = 0.2706\ldots$$

Cosine transform

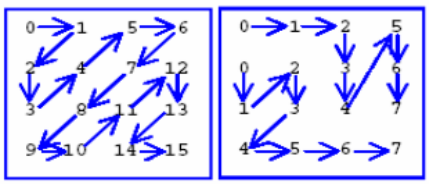
$$\mathbf{C}_f = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix}$$

Integer approximation



Quantization and scanning

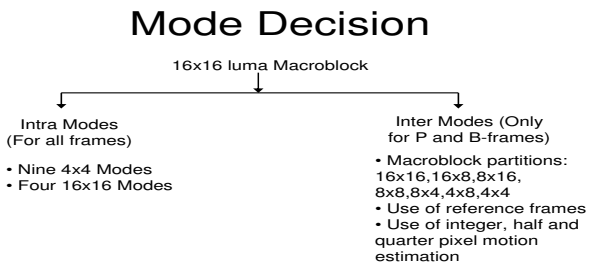
- Different quantizer for luminance, chrominance (new)
  - Thirty-two different quantization step sizes: the step sizes are increased at a compounding rate of approximately 12.5%
  - Matrices selected by the encoder based on perception optimization
- Two different coefficient-scanning patterns (new)
  - The simple zigzag scan
  - The double scan
- Entropy Encoding
  - Context-adaptive variable-length coding
  - Context-adaptive binary arithmetic coding



Each of the 2 4x4 blocks within a 4x4 block are scanned independently in a Z pattern. Suggested for higher quality images

H264 frame coding

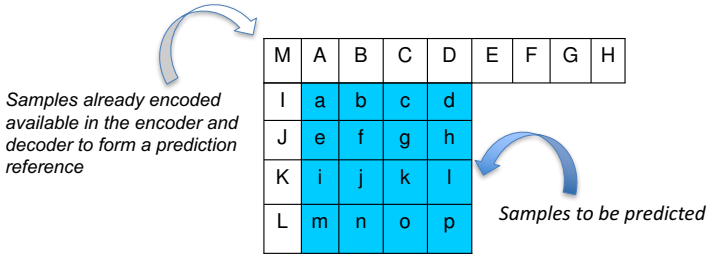
- There are two distinct frame coding:
  - *Intra frame coding*: each Macroblock can be encoded using blocks of pixels that are already encoded within the current frame
  - *Inter frame coding*: Macroblocks can be coded using blocks of pixels in previous or future encoded frames. The process of finding a match of pixel blocks in inter frame coding is called Motion Estimation



Intra-Frame Prediction

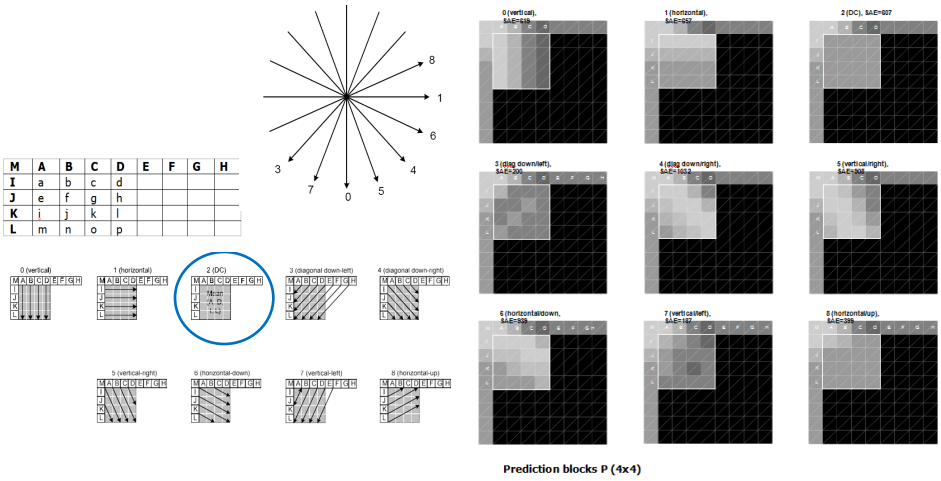
- If a block or macroblock is encoded in intra mode, a prediction block is formed based on previously encoded and reconstructed blocks. This prediction block P is subtracted from the current block prior to encoding.
- Intra blocks: 16 4x4 sub-blocks or 1 16x16 block (macroblocks are 16x16)

- Prediction samples a,b,...,p are predicted from samples A,..., M that have been encoded previously.



Intra Luma Prediction for 4x4 blocks

- 9 modes for Luma samples prediction applied to 4x4 blocks;
- 4 modes forChroma samples prediction applied to 8x8 blocks

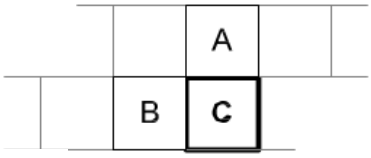


Prediction blocks P (4x4)

In Mode 2, the samples a,..,p are predicted using average of samples A,..,D and I,...,L.

Intra Luma Prediction

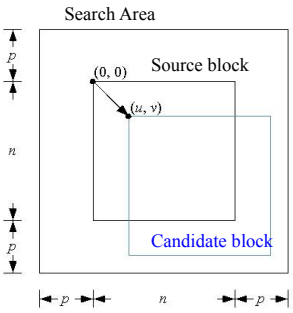
- For each current block C, the encoder and decoder calculate the most_probable_mode. The choice of intra prediction mode for each 4x4 block must be signalled to the decoder and this could potentially require a large number of bits.
- However, intra modes for neighbouring 4x4 blocks are highly correlated. For example, if previously- encoded 4x4 blocks A and B in Figure 8 were predicted using mode 2, it is likely that the best mode for block C is also mode 2.



Adjacent 4x4 intra coded blocks

Inter-frame prediction

- The inter-frame coding includes block-based motion compensation to remove temporal redundancy, the same principle adopted by major coding standards. Important differences include the support for a range of block sizes and fine sub-pixel motion vectors (1/4 pixel in the luma component).
- Each block is associated with a search region in the reference frame

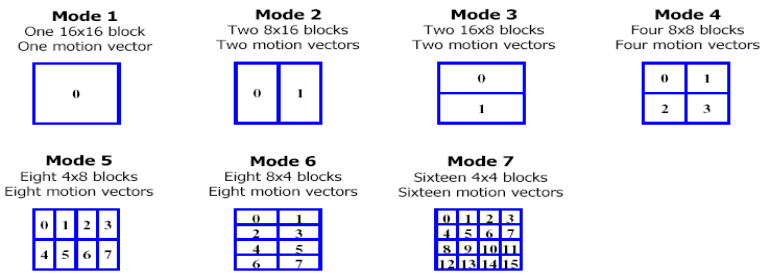


Motion vector: (u, v)

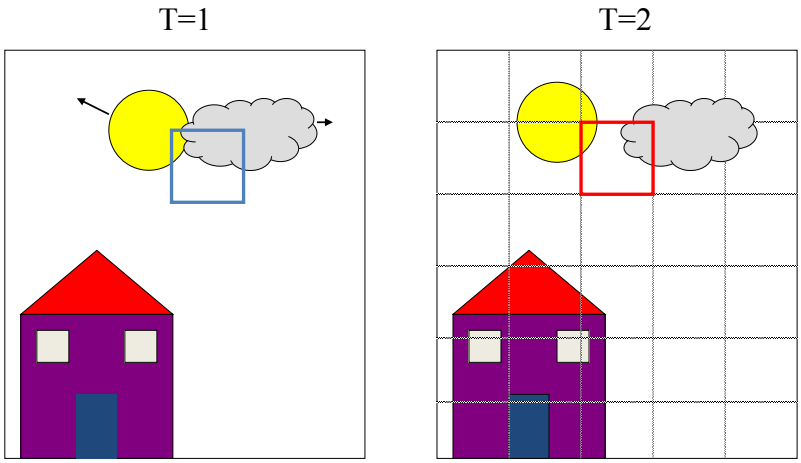
Search Area: $(n+2p)\times(n+2p)$

Block sizes and motion vectors

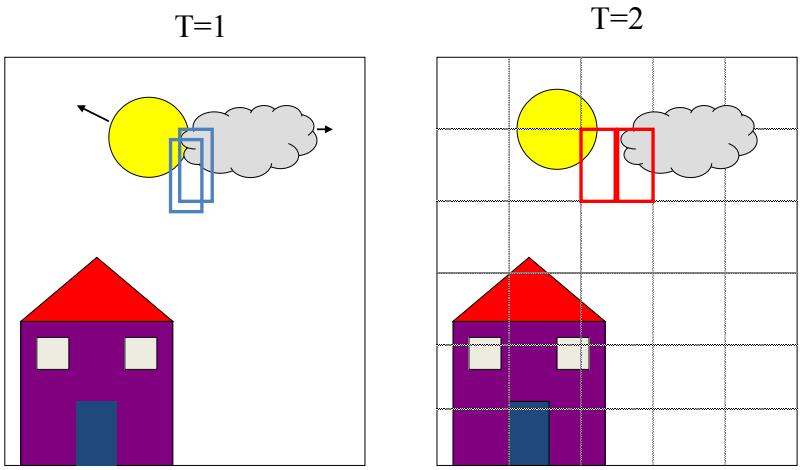
- Motion Estimation is where H264 makes most of its gains in coding efficiency.
- A number of different block sizes are used for motion prediction. Seven optional modes with inter blocks: 16x16, 16x8, 8x16, 8x8 (and subpartitions). Useful when size of moving/stationary objects is variable
- A separate motion vector is required for each partition or sub-partition: multiple motion vectors per macro-block are used



Motion Scale Example



H.264 Variable Block Size Example



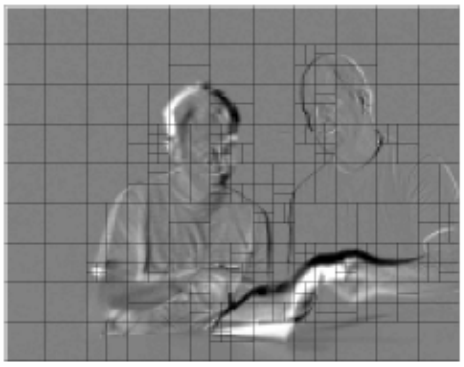
Block sizes and motion vectors

- Each motion vector must be coded and transmitted; in addition, the choice of partition must be encoded in the compressed bitstream.
- The choice of partition size has a significant impact on compression performance:
  - a large partition size (e.g. 16x16, 16x8, 8x16) requires a small number of bits to signal the choice of motion vector(s) and the type of partition; however, the motion compensated residual may contain a significant amount of energy in frame areas with high detail.
  - a small partition size (e.g. 8x4, 4x4, etc.) results into a lower-energy residual after motion compensation but requires a larger number of bits to signal the motion vectors and choice of partition(s).
- In general, a large partition size is appropriate for homogeneous areas of the frame and a small partition size may be beneficial for detailed areas.

Motion vector prediction

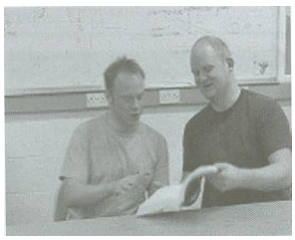
- Encoding a motion vector for each partition can take a significant number of bits, especially if small partition sizes are chosen.
- Motion vectors for neighbouring partitions are often highly correlated and so each motion vector is predicted from vectors of nearby, previously coded partitions.
- The “basic” predictor is the median of the motion vectors of the macroblock partitions or sub-partitions immediately above, diagonally above and to the right, and immediately left of the current partition or sub-partition.

- The encoder selects the best partition size for each part of the frame, to minimize the coded residual and motion vectors.

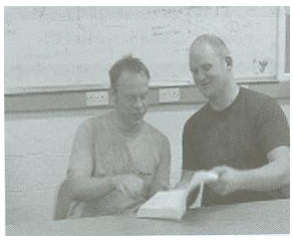


Residual frame (no motion compensation)

In figure: macroblock partitions for each area superimposed on the residual frame:
- little change between the frames (residual appears grey): a 16x16 partition is chosen
- detailed motion (residual appears black or white): smaller 4x4 partitions are more efficient



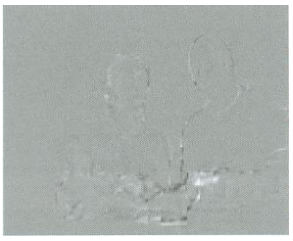
Frame F_n



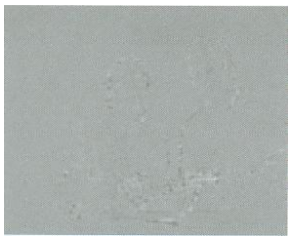
Frame F_{n-1}

Residual (no motion compensation)

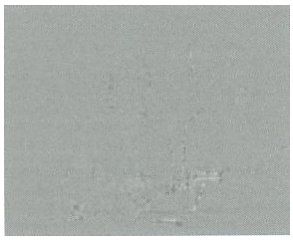
Residual (16x16 bock size)



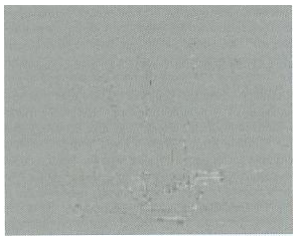
Residual (8x8 bock size)



Residual (4x4 bock size)



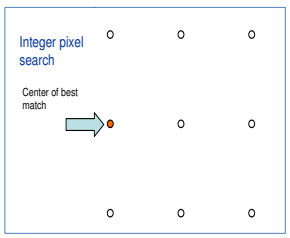
Residual (4x4 bock size; half pixel)



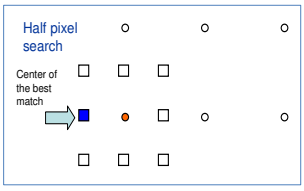
Residual (4x4 bock size; quarter pixel)

Sub-pixel Motion Estimation

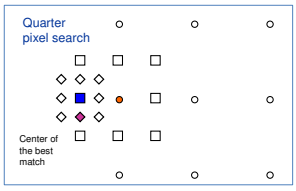
- H264 performs quarter pixel motion estimation



Find the position corresponding to the minimum block distortion



Half pixel motion estimation is done where the best match was found in the integer pixel search step.

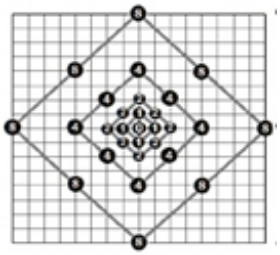


Quarter pixel motion estimation is done where the best match was found from the half pixel search step, giving us the final motion vector.

Diamond search algorithm

- The Diamond search algorithm employs two search patterns.
- Large diamond search pattern (LDSP) comprises nine checking points from which eight points surround the center one to compose a diamond shape.
- Small diamond search pattern (SDSP) consisting of five checking points forms a small diamond shape.

- LDSP is repeatedly used until the minimum block distortion occurs at the center point.



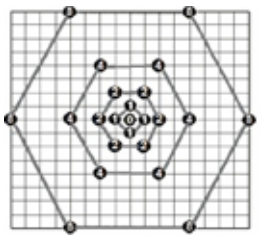
- Apply large diamond to the center of the search window.
- Compute the block distortions at all positions and check if the center position has the minimum distortion.
- Apply large diamond to the new center position. Find the new minimum block distortion.
- If the center is not the minimum, move the center to the minimum point and reapply the large diamond pattern.
- If the center is the minimum block distortion position, then apply a small diamond.

- The minimum block distortion position in this step gives the final motion vector.

Hexagonal search algorithm

- HEXBS algorithm can find a same motion vector with fewer search points than the Diamond Search algorithm, by calculating the minimum cost at 6 corner points of Hexagon

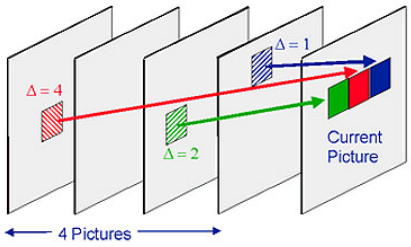
- The larger the motion vector, the more search points the HEXBS algorithm can save



122

Multi-frame motion-compensated prediction

- Multi-frame motion-compensated prediction (new):
  - Multi-frame motion-compensated prediction permits the use of more pictures (up to 16) than the previously decoded one for motion compensated prediction.
  - The decoder maintains a frame memory, in which multiple decoded reference pictures can be stored and used for motion-compensated prediction of following pictures. A reference picture index is transmitted in addition to the motion/displacement vector for each block coded using motion-compensated prediction.
- Useful if object leaves scene and then comes back, if the camera pans to the right, and then back to the left, …..



Scalable coding

- Scalable Video Coding (same as MPEG-2):
  allows construction of bit-streams that contain sub-bit-streams that also conform to standard:
  - SNR (quality) bit-stream scalability
  - Spatial (adaptation to smaller screens)
  - Temporal bit-stream scalability (adaptation to lower frame rates)

Multi-view coding

- Multi-view Video Coding (new):
  construction of bit-streams that represent more than one video of a video scene:
  - Example: stereoscopic (two-view) video
  - Example: free viewpoint television
  - Example: multi-view 3D television

- Multi view coding exploits large amount of inter-view statistical dependencies: cameras capture same scene from different viewpoints
- Combined temporal and inter-view prediction: frame from certain camera can be predicted not only from temporally related frames from the same camera, but also from neighboring cameras

- Two profiles in Multi-view Video Coding:
  - Multi-view High Profile (arbitrary number of views)
  - Stereo High Profile (two-view stereoscopic video)

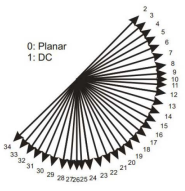
HEVC H265

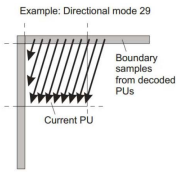
High Efficiency Video Coding

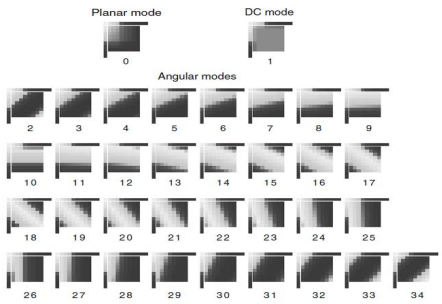
- High Efficiency Video Coding (HEVC) H265 is one of the potential successors to MPEG-4 H264. It offers about double the data compression ratio at the same level of video quality and supports resolutions up to 8192×4320 (ULTRA HD)
- Similarly to H264 it looks for areas that are redundant, both within 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 decision whether to code a picture area using inter-picture or intra-picture prediction is made at the encoder level.
- The first version of HEVC was completed in early 2015

Intra-frame coding

- Similarly to H264 if a block or macroblock is encoded in intra mode, a prediction block is formed based on previously encoded and reconstructed blocks. This prediction block PB is subtracted from the current block prior to encoding.
- More Intra Luma and Chroma prediction modes
  - Luma: 35 modes: Planar + DC + 33 angular prediction modes for all block sizes
  - Chroma: 5 modes: Derived mode (DM) + Planar + DC+ Horizontal + Vertical

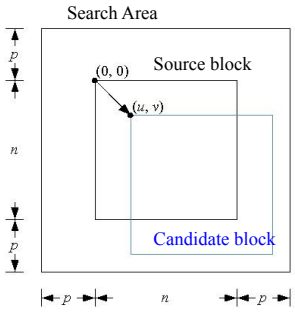






Inter-frame coding

- As H264 and other standards the inter-frame coding include motion compensation process to remove temporal redundancy. Each current frame is divided into equal-size blocks, called source blocks. Each source block is associated with a search region in the reference frame



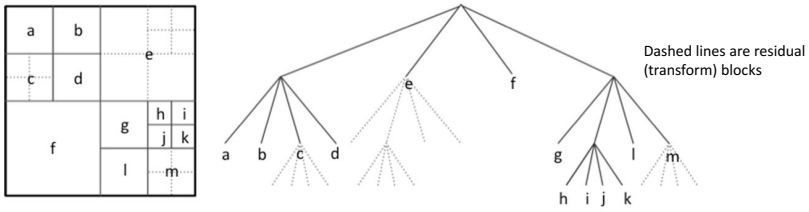
- H.264 used fixed size 16x16 macroblocks
  - Intra blocks: 16 4x4 sub-blocks or 1 16x16 block
  - Inter blocks: 16x16, 16x8, 8x16, 8x8 (and subpartitions)

- HEVC uses Coding Tree Blocks (CTBs) with size ranging from 16x16, 32x32 or 64x64 luma samples
  - Quadtree-like subpartitioning into coding blocks (CBs)
  - Minimum CB size: 8x8 (or larger if specified)

Motion vector: (u, v)
Search Area: $(n+2p)\times(n+2p)$

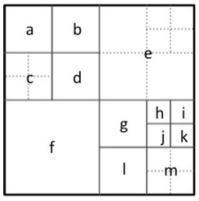
Quadtree-like subpartitioning

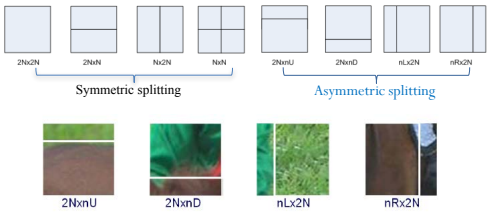
- A CTB can be recursively split into four square coding blocks (CBs) having the same size. CTBs are processed in raster scan order while CBs are processed in depth-first raster scan order. This results in the benefit that the top- and the left-neighboring coding blocks are always available
- A further quadtree structure is nested at the leaves of the coding quadtree. It indicates further subdivisions for residual coding and is referred to as residual quadtree.
- The highly variable quadtree-based structure allows a flexible adaptation to the input video signal's characteristic.



Prediction blocks

- Like H264 a number of different block sizes are used for motion prediction, useful when size of moving/stationary objects is variable
- Asymmetric splitting is possible with prediction blocks (PBs): 2Nx2N, NxN, 2NxN, Nx2N, 2NxnU, 2NxnD, nLx2N, nRx2N





H264 vs H265

- MPEG4 H264: Prediction and transform static
  - Intra/inter decision on (16x16) macroblock level
  - Prediction is coupled with block partition size
  - Transform size is always 4x4 or 8x8 in main mode
  - The prediction residual is coded using block transforms. Integer basis functions similar to those of a discrete cosine transform (DCT)
  - 1⁄4 luma pixel accuracy

- HEVC H265: Prediction and transform flexible
  - Intra/inter decision on (min. 8x8) Coding Block level
  - CB residual may be split into prediction blocks (PBs)
  - CB residual may be split into smaller Luma transform blocks (TBs)
  - TB structure may be further partitioned than the PB structure
  - The prediction residual is coded using block transforms. Integer basis functions similar to those of a discrete cosine transform (DCT) are defined for the square TB sizes 4×4, 8×8, 16×16, and 32×32.
  - 1⁄4 luma pixel accuracy

Residual coding

- The prediction residual is coded using block transforms.
- Integer basis functions similar to those of a discrete cosine transform (DCT) are defined for the square TB sizes 4×4, 8×8, 16×16, and 32×32

- The HEVC standard defines two tiers, Main and High, and thirteen levels.
  - A level is a set of constraints for a bitstream, e.g., max picture size, max sample rate.
  - The tiers were made to deal with applications that differ in terms of their maximum bit rate.

Level	Max Luma Picture Size (samples)	Max Luma Sample Rate (samples/s)	Main Tier Max Bit Rate (1000 bits/s)	High Tier Max Bit Rate (1000 bits/s)	Min Comp. Ratio
1	36 864	552 960	128	–	2
2	122 880	3 686 400	1500	–	2
2.1	245 760	7 372 800	3000	–	2
3	552 960	16 588 800	6000	–	2
3.1	983 040	33 177 600	10 000	–	2
4	2 228 224	66 846 720	12 000	30 000	4
4.1	2 228 224	133 693 440	20 000	50 000	4
5	8 912 896	267 386 880	25 000	100 000	6
5.1	8 912 896	534 773 760	40 000	160 000	8
5.2	8 912 896	1 069 547 520	60 000	240 000	8
6	35 651 584	1 069 547 520	60 000	240 000	8
6.1	35 651 584	2 139 095 040	120 000	480 000	8
6.2	35 651 584	4 278 190 080	240 000	800 000	6


H.264/AVC
HEVC
28 30
28 30
1080p@25fps at 2.7 Mbits/s