PROGETTAZIONE E PRODUZIONE MULTIMEDIALE

I2 - VIDEO

Prof. Alberto Del Bimbo

- Part I – Video
- Part II – MPEG1 encoding
- Part III – MPEG2 encoding
- Part IV – MPEG4 encoding

Part I - VIDEO

- 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

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**. Analog video, when converted into digital is coded in **YCrCb**:
  - 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.

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 can be either transferred with composite or component encoding. Composite encoding was invented to limit transmitted bandwidth at the introduction of color television and it has proven highly effective for broadcast. Component encoding was motivated by the need of compatibility with B/W TV and the importance of reducing bandwidth by spatial subsampling of color information. Quality of component is usually better than composite.
- Digital video uses component color encoding.

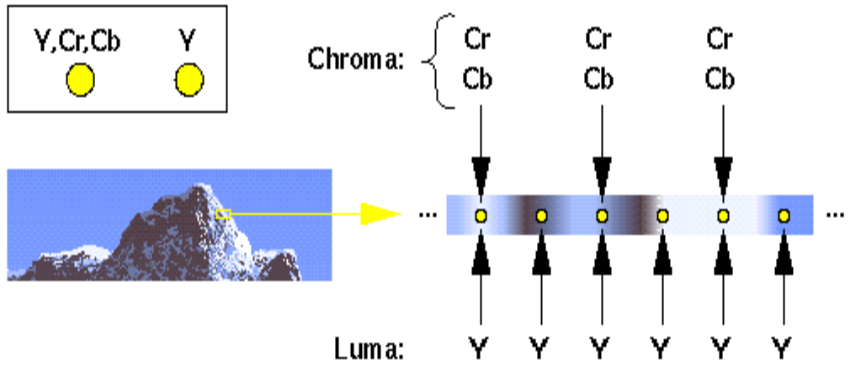
Video sampling

- Luminance and chroma information are *sampled* in each video frame.
- Sampling is expressed with three values: x,y,z
  - x = relative number of luma samples
  - y = number of chroma samples for odd lines
  - z = number of chroma samples for even lines

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

  …video compression algorithms are also available

For each line



Rec. 601 4:2:2 Sampling

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* 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

Video properties - 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 BPPs
  - with JPEG compression we reduce to 1-8 BPPs
  - with MPEG we go below 1 BPP

PAL, NTSC, SECAM, S-VIDEO….systems

- There are three main systems of anolog color video broadcast transmission (television). They are known as composite video because the brightness and color information are mixed together into a single signal:
  - 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…

- PAL (Phase Alternate Line) uses 625 horizontal lines at a field rate of 50 fields per second (or 25 frames per second). Only 576 of these lines are used for picture information with the remaining 49 lines used for sync or holding additional information such as closed captioning. (Au, NZ, UK, Europe)

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

- 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 systems transmit both chroma and luminance in the same signal. Having a composite signal is troublesome when the analog video is digitized in that it is difficult to separate the two signals.
- 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.

- 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.

- HDTV (High-definition television) has been finalized in the 90's has:
  - Higher resolution than NTSC and PAL: 1125 lines vs 525 and 625 lines: two million pixels per frame, roughly five times that of SD
  - Different aspect ratio: 16:9 (1.78 : 1) instead of 4:3 (1.33 : 1)
  - Digital broadcast
  - Allows for multiple picture rates: 60 Hz, 50 Hz, 30 Hz, 25 Hz and 24 Hz.
  - Progressive and interlaced pictures (more in following slides)

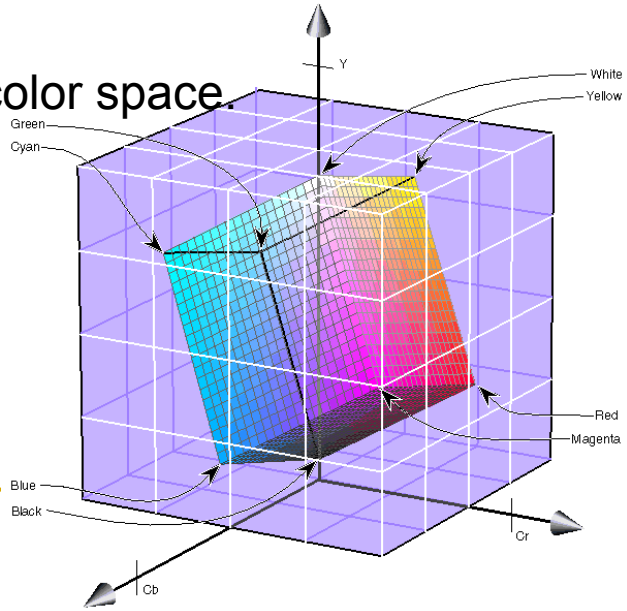
  - Rec. 709 is the specification for HD TV

  - HDMI is a common connection for HDTV devices; it's electrically compatible with DVI (common on monitors)

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 (YCrCb)
  - The pixel sampling frequency

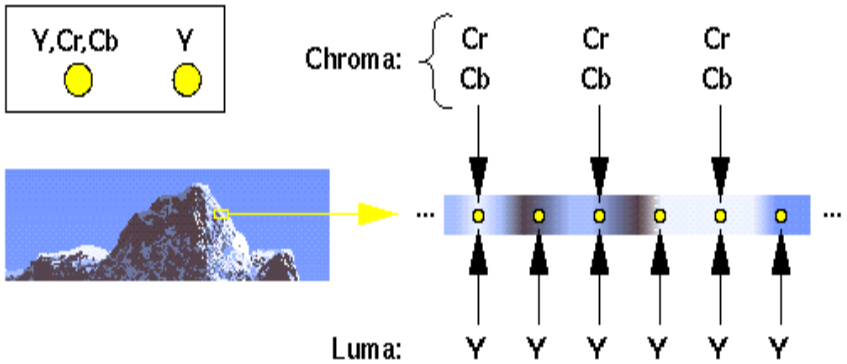
- YCrCb is the color space for digital video. YUV e YCrCb are similar but differ in the range of Y component values:
  - YUV from **0** to **255**
  - YCrCb from **16** to **235/240**
- Colors are distorted passing from RGB to YCrCb color space.



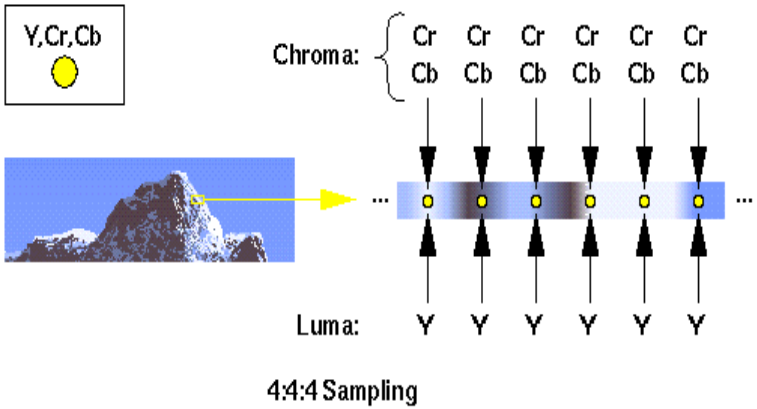
Pixel sampling frequencies are:

Digital video NTSC format : 720 x 480 pixels.
Digital video PAL format : 720 x 576 pixels.
Digital video HDTV format : 1125x 660 pixels

- CCIR 601 defines two distinct modes of color sampling:
  - 4:2:2 a pair of Cr Cb every two Y
  - 4:4:4 a pair of Cr Cb every Y



Rec. 601 4:2:2 Sampling

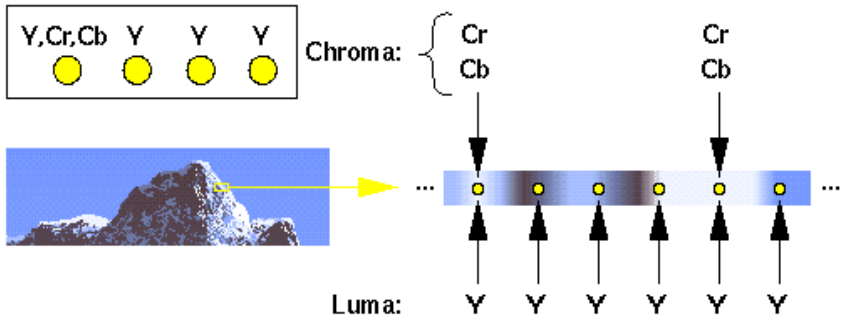


4:4:4 Sampling

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

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 and 400 for BetaSP
- Horizontal resolution for chroma is about 150 lines (about 1/4) for both DV and BetaSP

- 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 formats

- Other formats for (professional) digital video are:
  - D 1 (CCIR 601, 8bit, uncompressed)
  - D 2
  - D 3
  - D 5 (10bit, uncompressed) / D5 HD
  - D 9
  - Digital BetaCam
    - HDCAM / HDCAM SR for HD format (with 4:2:2 and 4:4:4 RGB)
- E.g.:
  - BBC uses D3
  - D2 manages 8 bit color
  - these standards have to deal with stuff like tape formats

Video field

- A video field is a set of image samples taken at the same time that is composed of alternate lines (all odd or even).

- Fields in a video sequence are sampled at different time instants according to the video *field rate*:
  - field rate for NTSC (both analog and digital) is ~60 field/sec (29.97 fps)

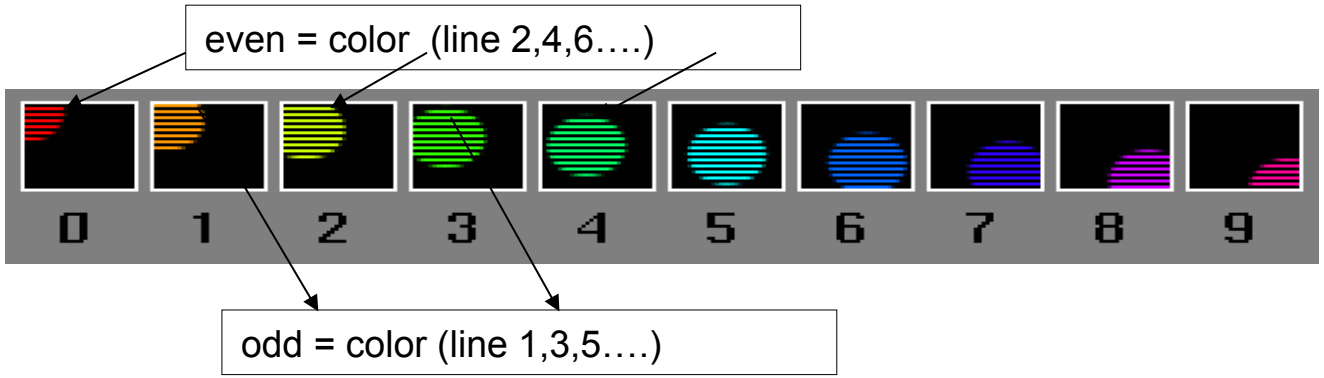
    242 lines active per field, 483 pixels per line
  - field rate for PAL (both analog and digital) is 50 field/sec (25 fps)

    290 lines active per field, 576 pixels per line

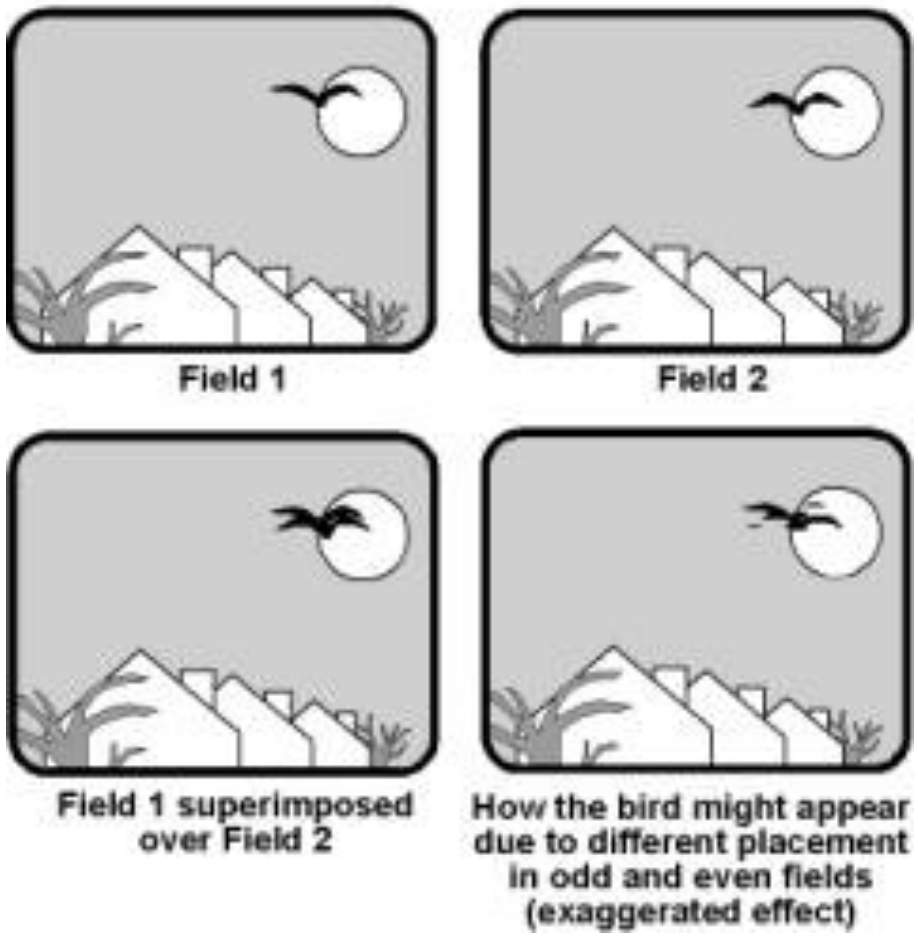
- Fields have been used historically due to the limited bandwidth of the TV signal (5,5 MHz). Computers use *frames instead of fields* (all the lines are sent together)
- Fields are displayed interlaced i.e. first the odd, then the even lines… frequency is such that two fields are perceived as a single image

- Recording a scene at 50 frames per second using a film:

- Recording a scene at 50 frames per second using a PAL camera (fields are taken at different instants):



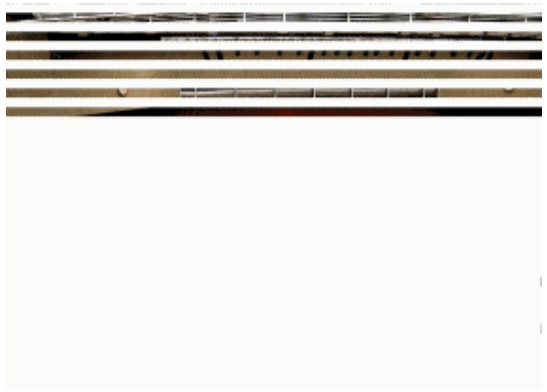
- Data in a video field are distinguished both spatially and temporally. At each time instant one half of the information is lost.



- Video formats for computer instead are not interlaced (*noninterlaced* or *progressive scan*). If we use f.e. a Matrox RT board to register VHS video in DV format we obtain interlaced video. This can create problems as in figure. Software tools are needed to reconstruct the full frame.

Interlaced vs. progressive

- EBU (European Broadcasting Union) is in favor of progressive scan



Interlaced

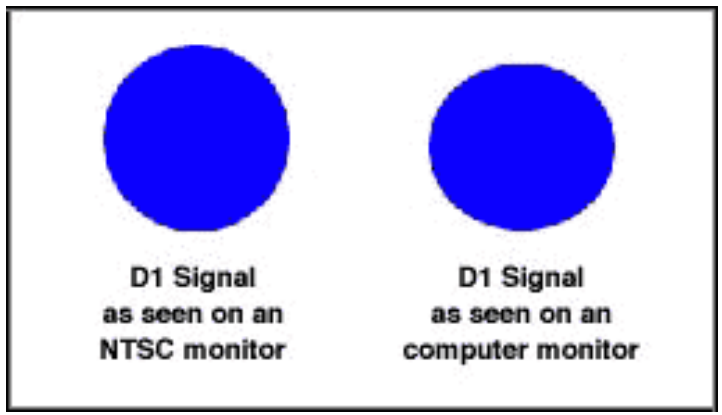
Progressive

Image 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
  - Cinemascope/Panavision: 2.35

Pixel aspect ratio

- Pixel aspect ratio = pixel height/width
- Some video formats require non square pixels (e.g. 10:11 for digital video D1 - CCIR 601)
- Computer monitors use square pixels
  - 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.



Frame resolution

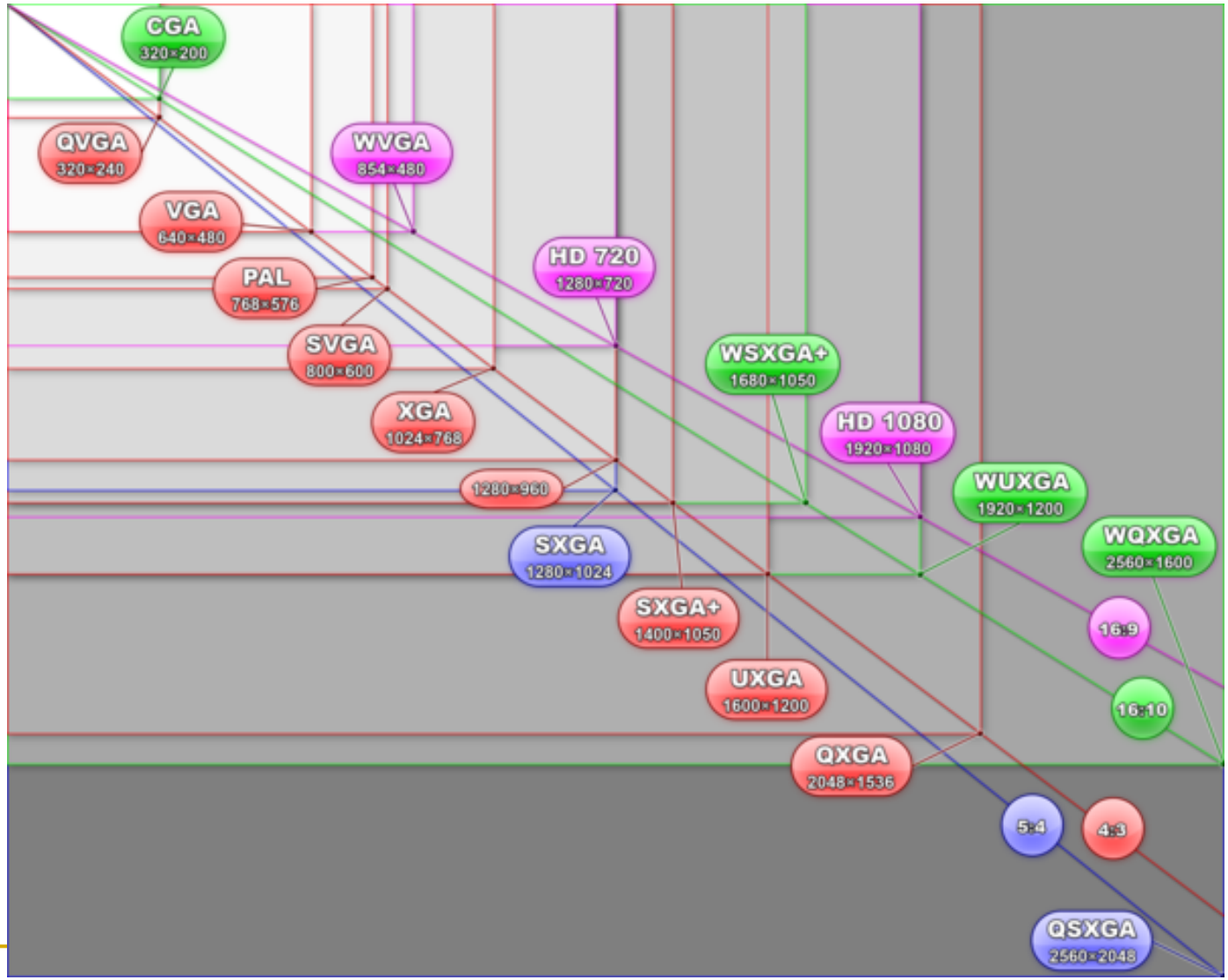
- Typical frame resolutions

QCIF
: Quarter Common Intermediate Format, 176 x 144 (25,300 pixel)

QVGA
: Quarter Video Graphics Array, 320 x 240 (76,800 pixel)

CIF
: Common Intermediate Format, 352 x 288 (101,400 pixel)

VGA
: Video Graphics Array, 640 x 480 (307,200 pixel)



Formats of video files

- 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 (.mov) - multiplatform
  - Microsoft AVI (.avi)/ Windows Media Video (.wmv)
  - MPEG (.mpg o .mpeg) – multiplatform

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

AVI

- Microsoft's AVI (Audio Video Interleave) container is a special kind of RIFF (Resource Interchange File Format – chunk based, now used by Google WebP picture format).
  - it has very simple design attributes (file size limit is 4GB – was 2GB!, no variable bit/ framerate)
  - It is able to contain a very large amount of video formats, specified using a Four Character Code (FourCC) to define which Codec was used to store the video stream.
  - Although technically superior containers like Matroska exist, AVI remains a strong choice because of large support from existing applications. It is often used as a container video format by compression codecs such as Xvid and Divx.
  - For distribution, AVI is one of the more popular choices, but has been losing ground to other containers lately.
  - The container has no native support for modern compression features like B-Frames.

ASF / WMV / WMA

- Advanced Systems Format is Microsoft's proprietary digital audio/digital video container format, especially meant for streaming media. ASF is part of the Windows Media framework.
- The format does not specify how (i.e. with which codec) the video or audio should be encoded; it just specifies the structure of the video/audio stream (similarly to the function performed by the QuickTime, AVI, or Ogg container formats).
- The most common filetypes contained within an ASF file are Windows Media Audio (.WMA) and Windows Media Video (.WMV)
- Can contain metadta (like ID3 tags in MP3)
- Supports Digital Rights Management

- Windows Media Video (WMV) is a compressed video compression format for several proprietary codecs developed by Microsoft. WMV9 supports HD and is a DVD (Blu-Ray) standard

Quicktime

- QuickTime is a file format for storing and playing back movies with sound. Developed by Apple, but it's also commonly used in Windows systems and other types of computing platforms. Typically, QuickTime files have the ".MOV" filename extension.
- The QuickTime File Format, specifies a multimedia container file that contains one or more tracks, each of which stores a particular type of data: audio, video, effects, or text (e.g. for subtitles).
- Supports progressive download and metadata, variable bitrate and variable frame rate. It can be streamed using the Darwin Streaming Server.

- The MPEG-4 file format specification was created on the basis of the QuickTime format specification published in 2001.

Matroska / WebM

- Matroska is an open-source container, and is file format that can hold an unlimited number of video, audio, picture or subtitle tracks inside a single file.
- Designed to hold any type of codec. (Audio, Video, Subtitle, etc)
- Uses Extensible Binary Meta Language (EBML), sort of binary equivalent to XML, for storing data in XML-like tags.

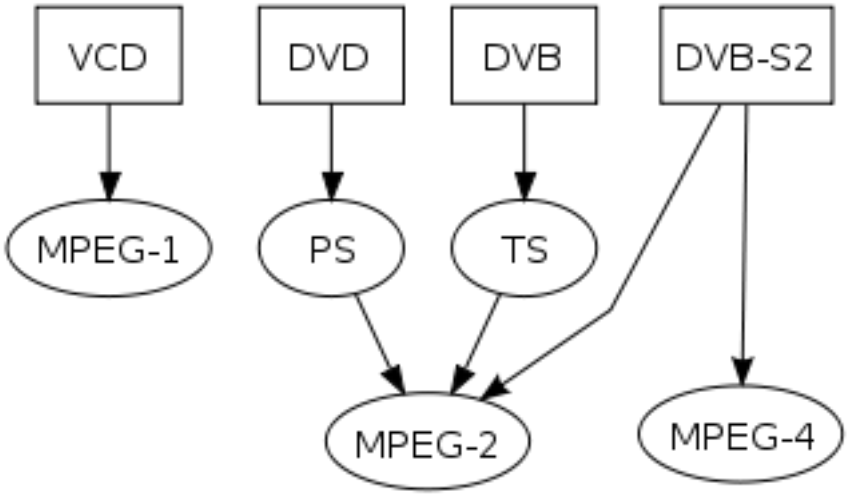
- The WebM container, proposed by Google, is a profile of Matroska that is forced to use a specific codec for audio and video.

Ogg

- Ogg is a free, open standard container format and is designed to provide for both efficient streaming and manipulation of high quality digital multimedia.
- The Ogg container format can multiplex a number of independent streams for audio, video, text (such as subtitles), and metadata.
- It's associated to two audio (Vorbis) and video (Theora) open source codecs. It's playable with VLC and out-of-the-box on Linux distros. Playable on Win/Mac using plugins.
- Firefox 3.5, Chrome 4, and Opera 10.5 support — natively, without platform-specific plugins — the Ogg container format
  - was proposed as default/standard for HTML5 video, but proposal has been retired

MPEG

- MPEG is both a video file format and a compression method defined according to ISO standard. It distinguishes:
  - MPEG 1
  - MPEG 2
  - MPEG 4



- DVDs use a special MPEG2 container called VOB that supports additional features like many audio and subtitle streams.

MPEG formats

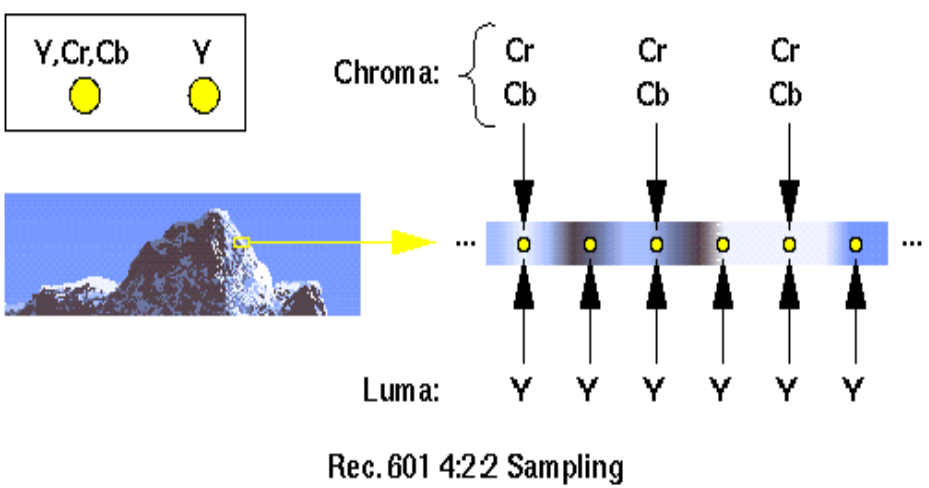
- MPEG1 and MPEG2 have defined the Program stream (PS)
- MPEG-PS is a container format for multiplexing digital audio, video. It was designed for reliable media, such as disks (like DVDs).

- 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.
- Transport Stream transmissions may carry multiple Program Streams.

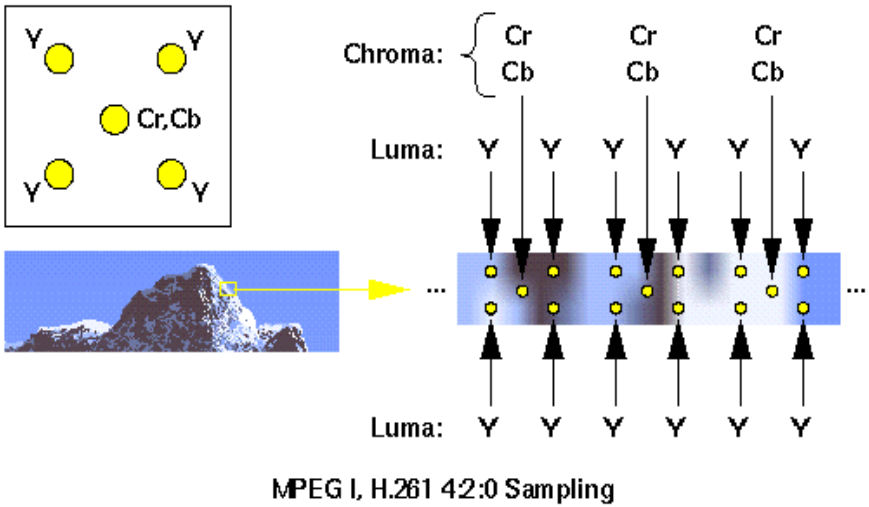
MPEG 1

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

- CCIR 601 NTSC: 2:1 in horizontal luminance; 2:1 in time; 2:1 in vertical chrominance. Lines are dropped so to make data divided by 8 and 16
  - Frame size: 720x243
  - ~60 fields per second
  - 4:2:2 sampling



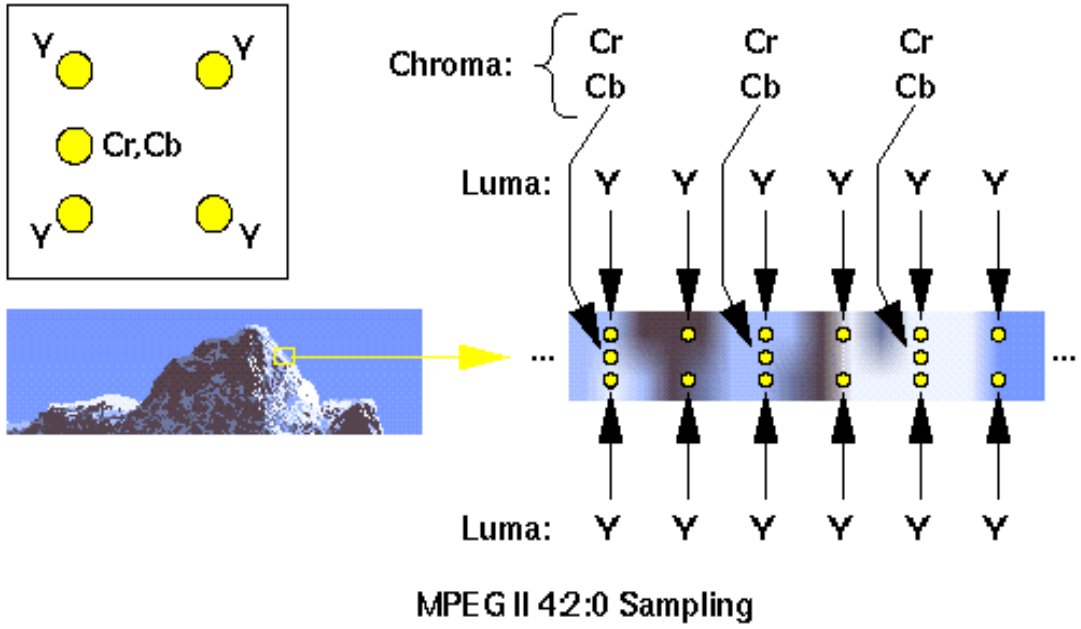
Rec. 601 4:2:2 Sampling



MPEG I, H.261 4:2:0 Sampling

MPEG 2

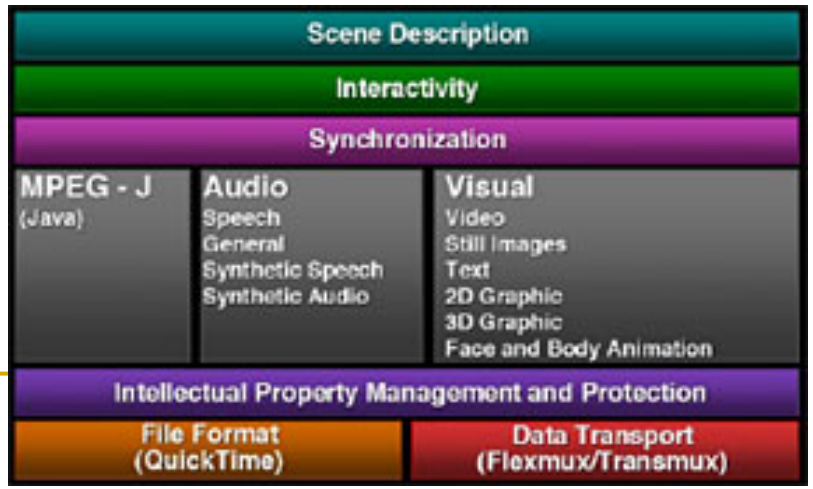
- MPEG2 bitrate 4 Mbit/s. MPEG2 was defined to provide a better resolution than MPEG1 and manage interlaced data. Based on fields instead of frames. Used for DVD and HDTV
- Frame sixe: 720x480
- 4:2:0 sampling



MPEG II 42:0 Sampling

MPEG 4

- MPEG4 format was inspired by the QuickTime format, and may contain different streams and media. Can contain metadata
- Audio-only MPEG-4 files generally have a .m4a extension.
- 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.



Flash Video

- Usually with an .flv extension, until Flash Player 9 Update 3, it was the only container format that Flash supported. New (open) format is .f4v based on the ISO base media file format (generalised from MP4 format)
- It's the format of choice for embedded video on the web. Notable users of the Flash Video format include YouTube, Hulu, Google Video, Yahoo! Video, metacafe, Reuters.com.
- More recent versions of Flash also support the MPEG 4 compression, that Adobe claims will be default.
- It has several transport modes:
  - Streaming based on RTMP (proprietary protocol); requires server like AFMS or open sourced Red5.
  - Progressive download based on HTTP
  - Standalone FLV file
  - Inclusion in SWF files

- FLV format and H.264 codec have issues with streaming. Adobe says that F4V should be used as container for H.264.

- Supported media types in FLV file format:
- Video: **On2 VP6**, Sorenson Spark (Sorenson H.263), Screen video, H.264
- Audio: MP3, ADPCM, Linear PCM, Nellymoser, Speex, AAC, G.711 (reserved for internal use)

- Supported media types in F4V file format:
- Video: H.264
- Images (still frame of video data): GIF, PNG, JPEG
- Audio: AAC, HE-AAC, MP3

Video compression

- Video compression algorithms can be lossy and lossless
  - but typically are lossy, starting with color subsampling
  - modern lossy algorithms aim at reducing **perceptual** quality loss
- Algorithms can be symmetric or not symmetric, in terms of (de)compression time/complexity
  - video compression for video conference needs to be symmetric
  - 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)

Some codecs: Vorbis

- Vorbis is an "open-source" digital audio compression format. Because Vorbis is most often used in conjunction with a Ogg digital A/V container format, it's usually referred to as "Ogg Vorbis."

- Vorbis, like MP3, is a lossy compression system, removing frequencies deemed inaudible.
- Both formats offer variable-bitrate encoding options, for better efficiency. But the algorithms Vorbis uses to decide which information to discard differ from those used by MP3. Proponents claim that the Vorbis format outperforms MP3, producing files that are significantly smaller than MP3s of similar perceptual sound quality.

- It’s the audio codec used in WebM format.

Some codecs: Theora

- Theora is a free lossy video compression format. Developed by the Xiph.Org Foundation and distributed without licensing fees.
- Theora is derived from the proprietary VP3 codec, released into the public domain by On2 Technologies. It is broadly comparable in design and bitrate efficiency to MPEG-4 Part 2, early versions of Windows Media Video.

- Theora is a variable-bitrate, DCT-based video compression scheme. Theora also uses chroma subsampling, block-based motion compensation and an 8-by-8 DCT block. We'll see these techniques when analyzing MPEG.
- Theora supports intra-coded frames and forward-predictive frames, but not bi-predictive frames which are found in H.264.
- Theora also does not support interlacing, or bit-depths larger than 8 bits per component.

- Version 1.2 optimizes for perceptual quality measures rather than signal reconstruction measures

Some codecs: Vp6

- On2 TrueMotion VP6 is a proprietary lossy video compression format and video codec. This codec is commonly used by Adobe Flash, Flash Video, and JavaFX media files.
- Multipass encoding, supports HD. It uses motion compensation to exploit temporal redundancy, a DCT transform to exploit spatial redundancy, a loop filter to deal with block transform artifacts, and entropy encoding to exploit statistical correlation.

- Popularized by YouTube: de fact internet video standard.
- Decoder available in FFMpeg, but encoder is not due to IPR problems.

One of the problems with algorithms that use frequency based block transforms is that the reconstructed video sometimes contains visually disturbing discontinuities along block boundaries.
A solution is to apply a filter within the reconstruction loop of both the encoder and decoder. Such "loop filters" smooth block discontinuities in the reconstructed frame buffers that will be used to predict subsequent frames.

VP 6

Windows Media 9

Some codecs: Vp8 / WebM

- VP8 is an open video compression format released by Google, originally created by On2 Technologies.
- Designed to handle multicore processors (reducing data dependancies between pixel blocks that ghave to be encoded)
- Reduced complexity of decompression with SIMD instructions, 8 bit math, optimization for ARM processors (used in mobile devices)
- Tries to compete with H.264
- No interlacing: simpler code.
- Intra-frame coding is the base for Google's WebP image coding

- Main criticisms:
  - very similar to H.264 compression scheme
  - almost no specs: specs made with code comments
  - bad quality of code: FFMpeg's ffvp8 library is faster than Google's library

Some codecs: Huffyuv

- Huffyuv (or HuffYUV) is a lossless video codec which is meant to replace uncompressed YCbCr as a video capture format.

- Based on Huffman coding (used in programs like ZIP, RAR, etc).
  The HuffYUV codec compresses an image by predicting the value of a pixel from its neighbors, and computes an error value (delta) by subtracting it from the effective pixel value.
  The result of this operation is then compressed with a Huffman algorithm, using a different table for each channel.

- Extremely fast but not suitable for playback.It’s useful for editing.

Some codecs: M-JPEG

- Motion JPEG (M-JPEG) is an informal name for a class of video formats where each video frame or interlaced field of a digital video sequence is separately compressed as a JPEG image.
- Extremely fast, used in many video capture cards and recent SLR cameras that capture also video.
- No temporal compression: ideal for editing

Some codecs: Dirac/Schrödinger

- Dirac is an open and royalty-free video compression format, specification and system developed by BBC Research
- Designed for a wide range of uses, from delivering low-resolution web content to broadcasting HD and beyond, to near-lossless studio editing (using no temporal compression).
- Based on wavelet instead of DCT: should preserve fine details better than block based transforms.
  - No blocky artifacts.

Video and web

- HTML5 includes a <video> element for embedding video into a web page. There are no restrictions on the video codec, audio codec, or container format you can use for your video. One <video> element can link to multiple video files, and the browser will choose the first video file it can actually play. It is up to you to know which browsers support which containers and codecs.

CODECS/CONTAINER	IE	FIREFOX	SAFARI	CHROME	OPERA	IPHONE	ANDROID
Theora+Vorbis / Ogg	·	**3.5+**	†	**5.0+**	**10.5+**	·	·
H.264+AAC / MP4	**9.0+**	·	**3.0+**	**5.0+**	·	**3.0+**	**2.0+**
VP8+Vorbis / WebM	**9.0+***	**4.0+**	†	**6.0+**	**10.6+**	·	‡

† **if plugin installed**
‡ **no deadline yet**

- There is no single combination of containers and codecs that works in all HTML5 browsers.

Part II - MPEG 1

- MPEG1 is an ISO standard (ISO/IEC 11172) developed to support VHS quality video at bitrate of ~1.5 Mbps

- MPEG1 was developed for progressive video (non interlaced)
  - MPEG1 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.

- MPEG (Moving Picture Expert Group) 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
  - 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*.
- MPEG1 defines the syntax of encoding a stream video and the method for decoding. However the encoder can be implemented in different ways (f.e. there is no standard for motion estimation)

CPB

- MPEG1 permits resolution up to 4095x4095 at 60 fps (100 Mbit/s).

  However MPEG1 video is usually seen using SIF resolution : 352x240, 352x240, 320x240 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.

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)

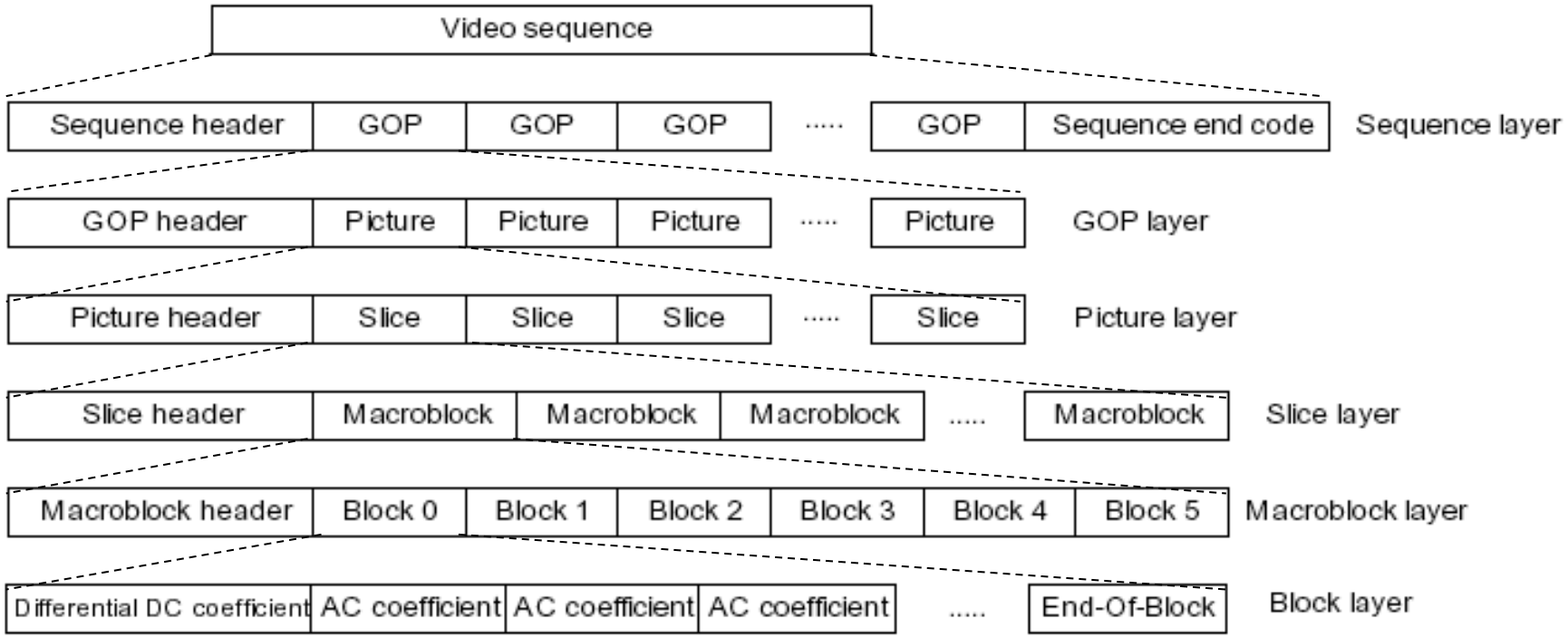
- MPEG1 can provide compressed video at broadcast quality with a bandwidth of 4 Mbps - 6 Mbps. Similar quality is obtained in MPEG-2 with 4 Mbps bandwidth, thanks to fields

- Possible resolutions in CPB are defined according to the Standard Image Format, being 352x240 o 352x288.
- When coding, in order to adapt the MPEG stream to the size of a TV screen it is scaled to 704x480 or 704x576. Typically field 2 is ignored and field 1 is scaled horizontally.
  - The number of samples is made divisible by 16 to make it simple adaptation to macroblocks
    (704/2) / 16 = 22…

Resolution	Frames per Second
352 × 240	29.97
352 × 240	23.976
352 × 288	25

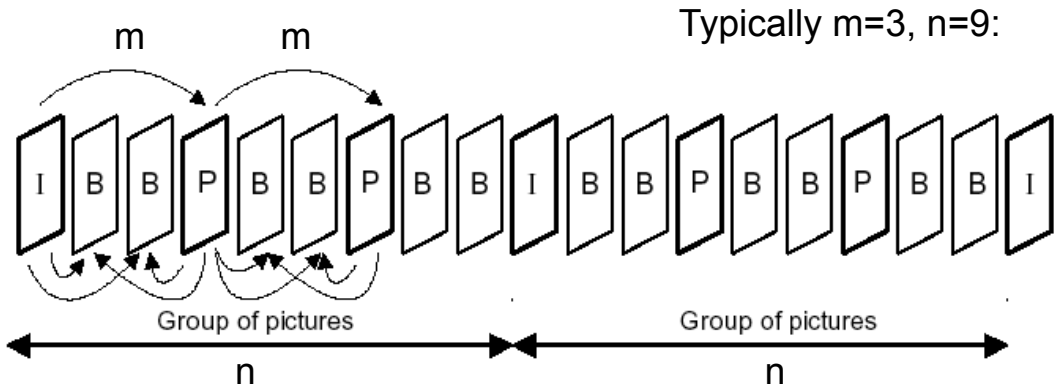
MPEG: 6 layers

- 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


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

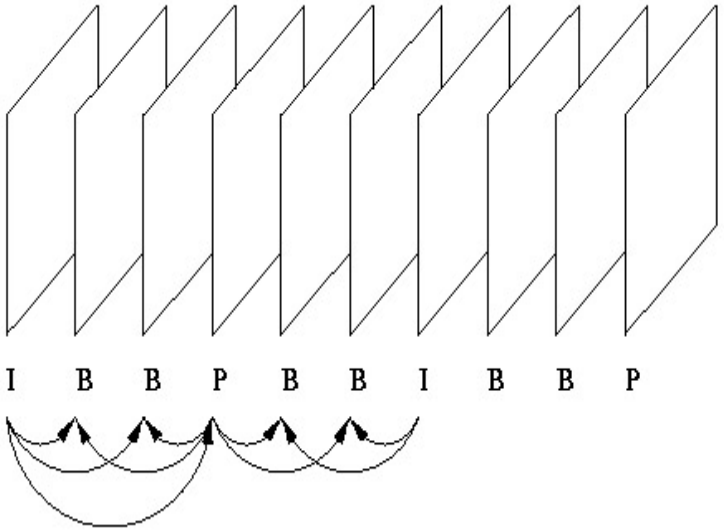
GOP

- A video sequence is decomposed in Groups of Pictures (GOPs).
  - I, P, B, D 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 (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).
- Typical GOP lenght are 14-17



Frames

- Frame types: I, P, B. They occur in repetitive patterns within a GOP

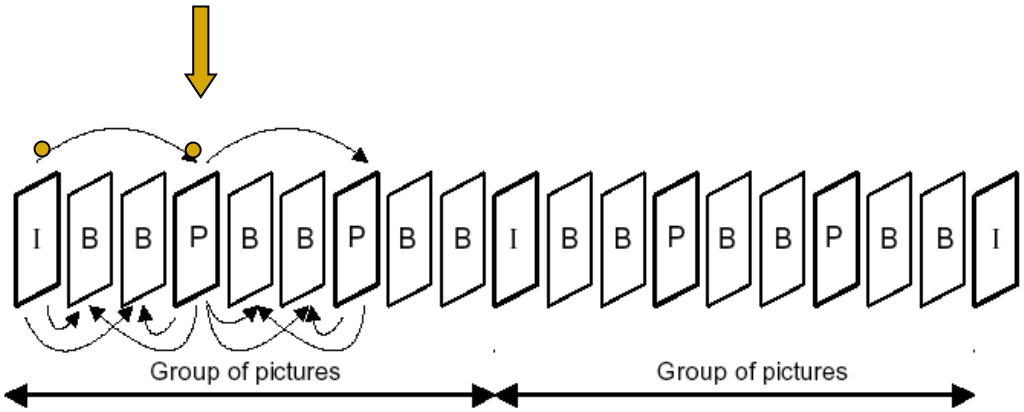


Predictive relationships between I, P and B frames

I-frames

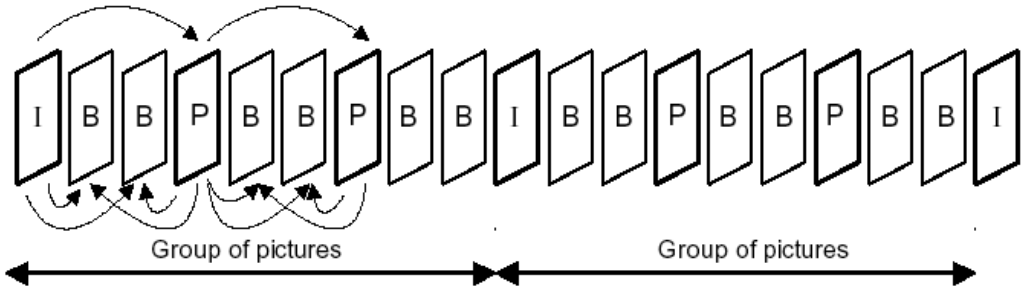
- Intra – coded frames are so called because they are decoded independently from any other frames. They are identical to JPEG frames.

- 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)



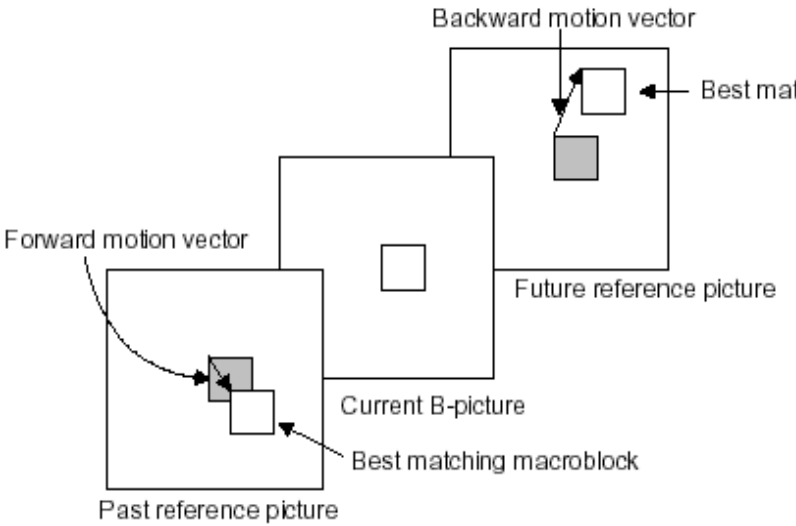
P-frames

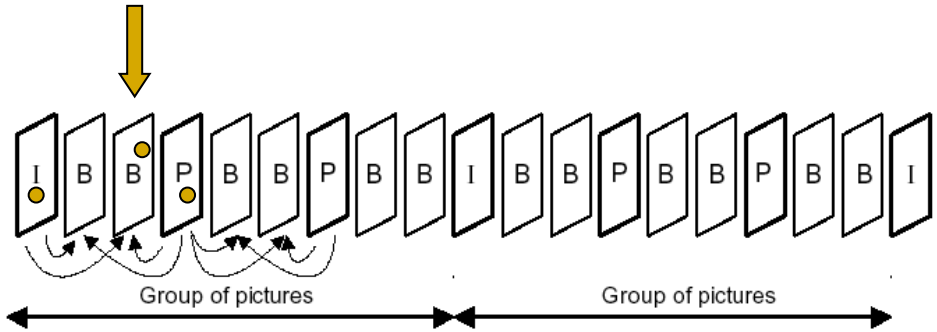
- Predictive-Coded frame are coded with forward motion prediction from preceding I o P frame.
- Improve compression by exploiting thetemporal redundancy. They store the difference in image from the frame immeditely preceding it. The difference is calculated using motion vectors.



B- frames

- Bi-directional-Coded frame are coded with bidirectional (past and future) motion compensation using I e P frame (no B frame). Motion is inferred by averaging past and future predictions. Harder to encode introduce 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 Bframes computationally complex and requires large data buffers.

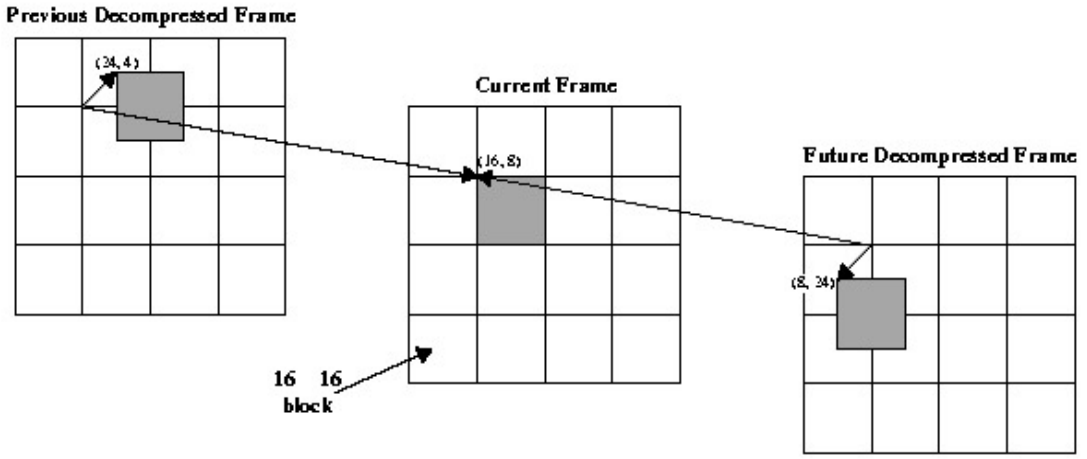




Frames and Macroblocks

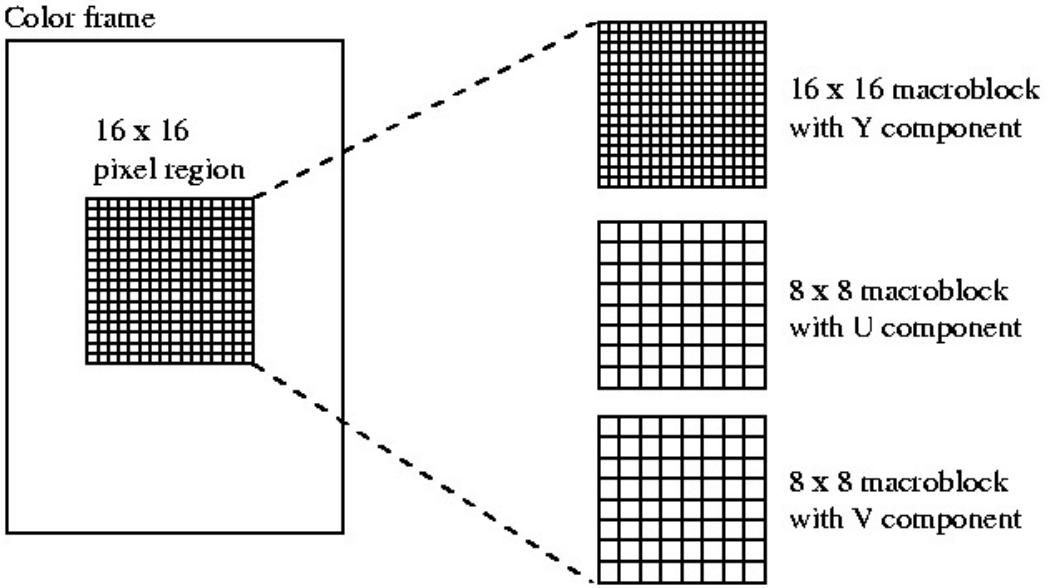
- Each video frame contains macroblocks that is the smallest independent unit of video considered by MPEG. M*acroblocks* are set of (16x16 pixel) are necessary for purposes of the calculation of motion vectors.

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



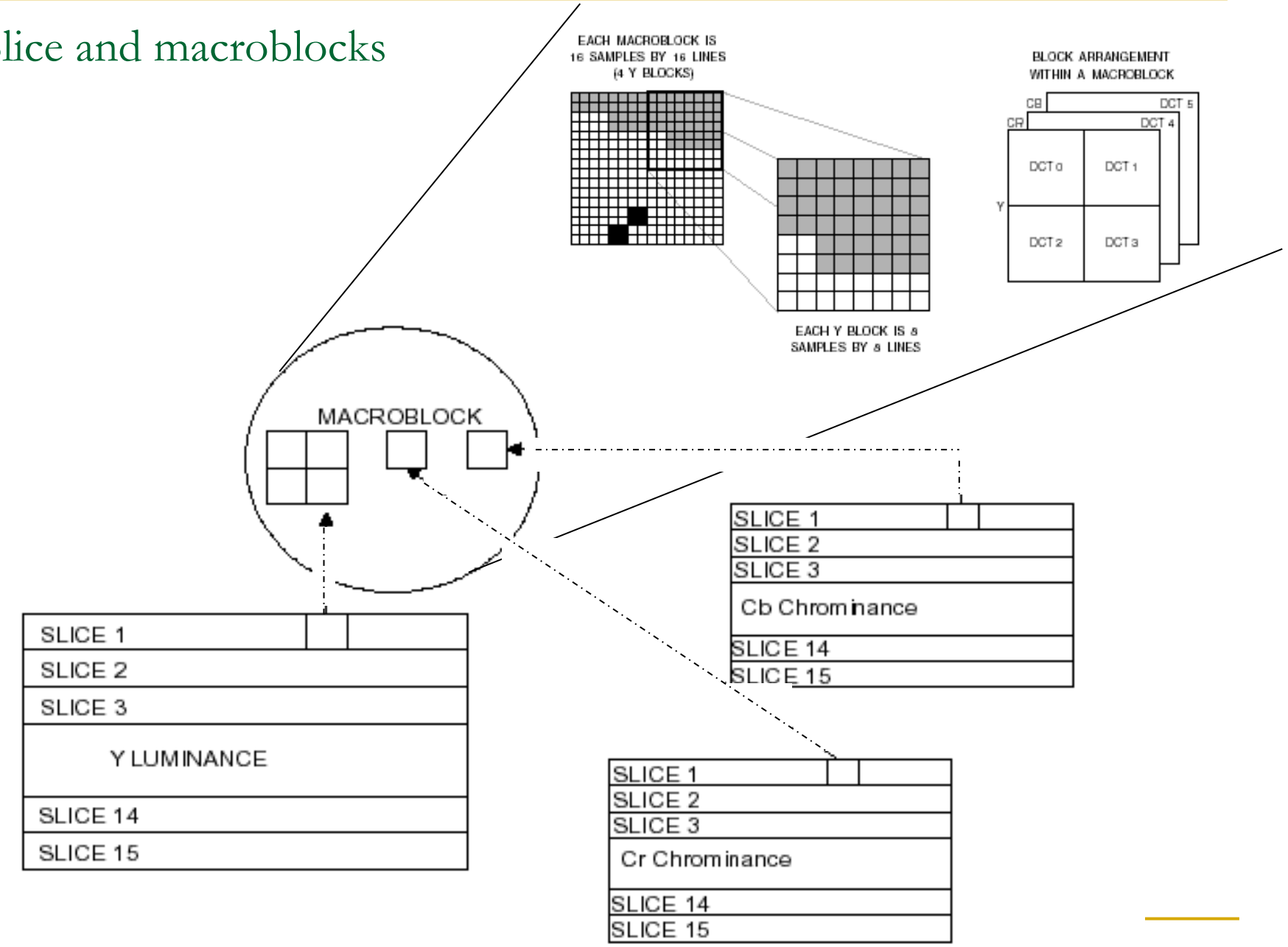
Block motion compensation

- Each macroblock is encoded separately.

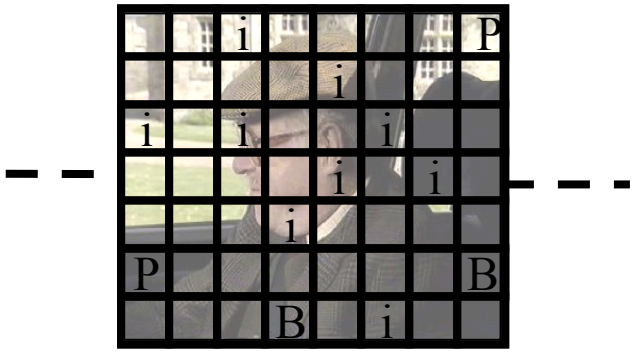


The component of a macroblock for motion compensation Y is luminance component. U and V are chrominance components.

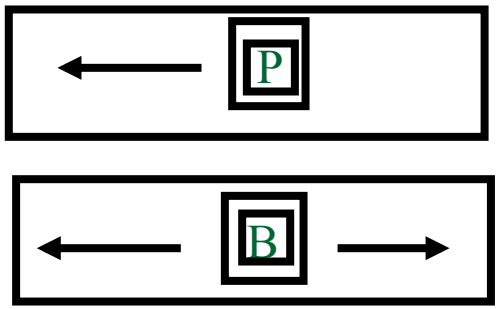
Slice and macroblocks



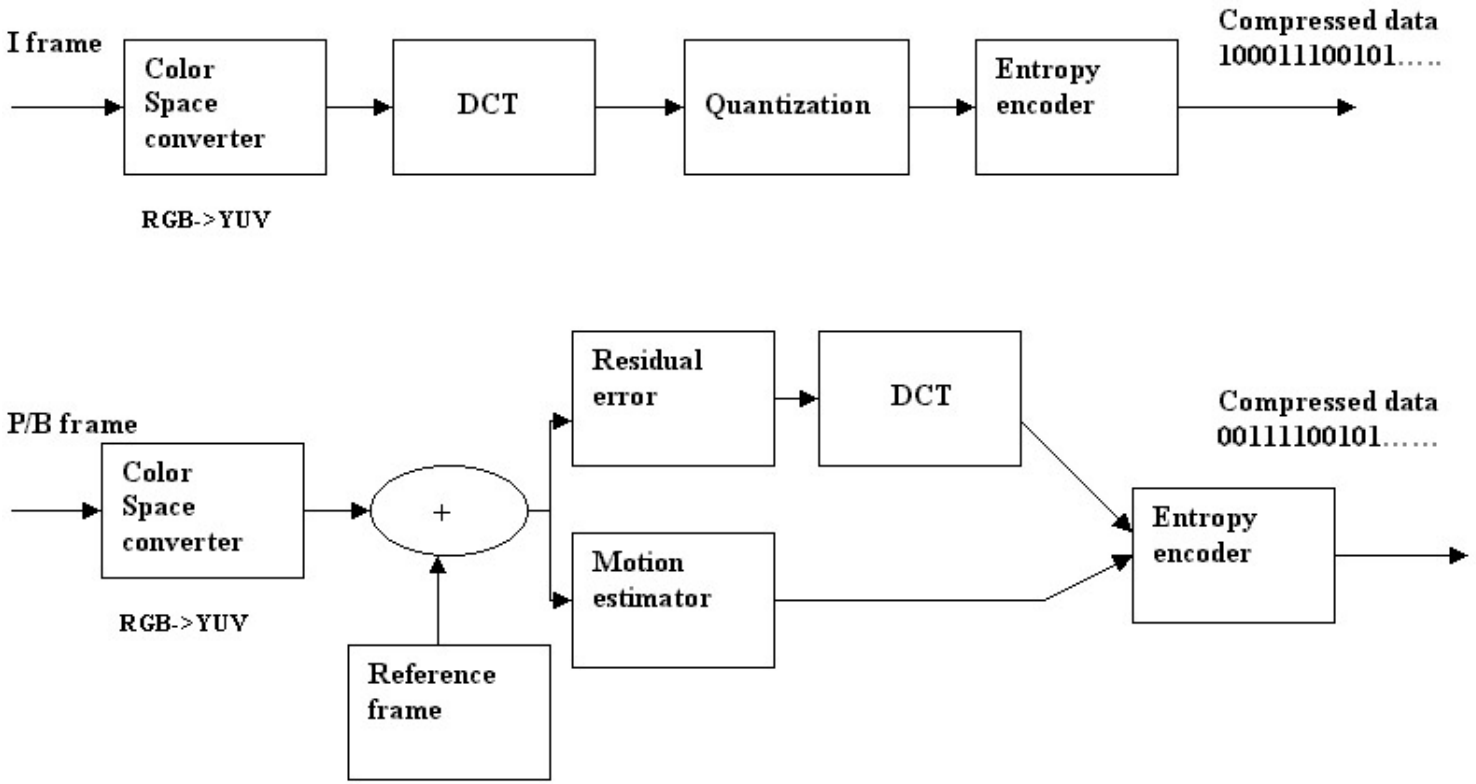
Frame with macroblocks



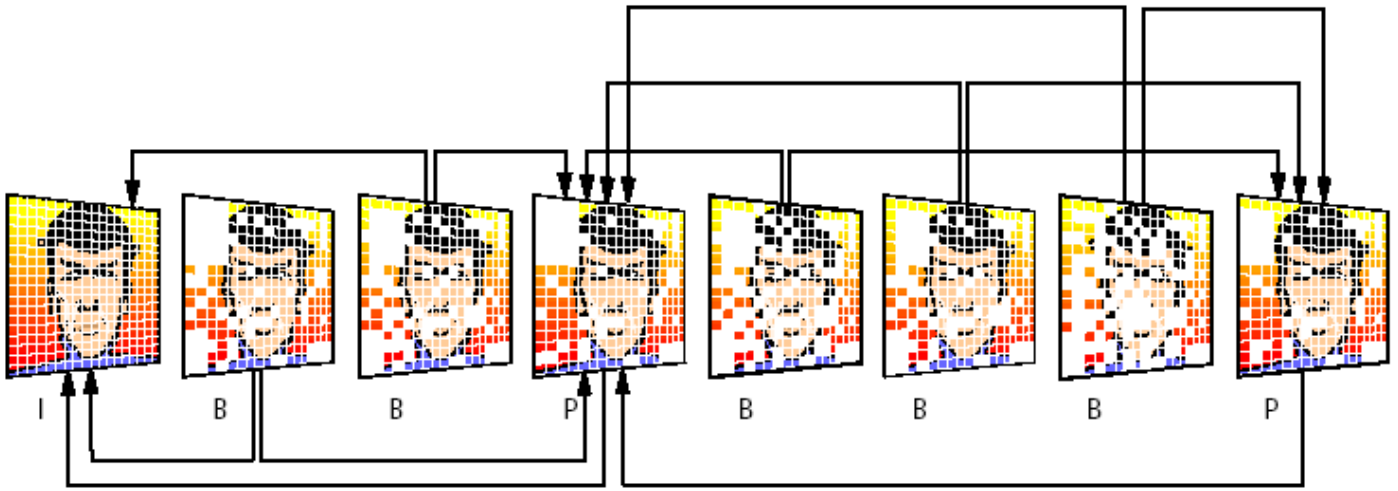
- Three types of macroblocks
  - I encoded independently of other macroblocks
  - P encode not the region but the motion vector and error block of the previous frame
  - B same as above except that the motion vector and error block are encoded from the previous or next frame



- I frames contain Intra-coded MBs (I) with direct encoding from the image samples by 2D Discrete Cosine Transform (DCT)
- P and B frames contain encoding of residual error after prediction
  - P frames contain Intra-coded MBs (I) or Forward-predicted MBs (P)
  - B frames contain Intra-coded (I) Forward or/and Backward-predicted MBs or skipped (P or B)

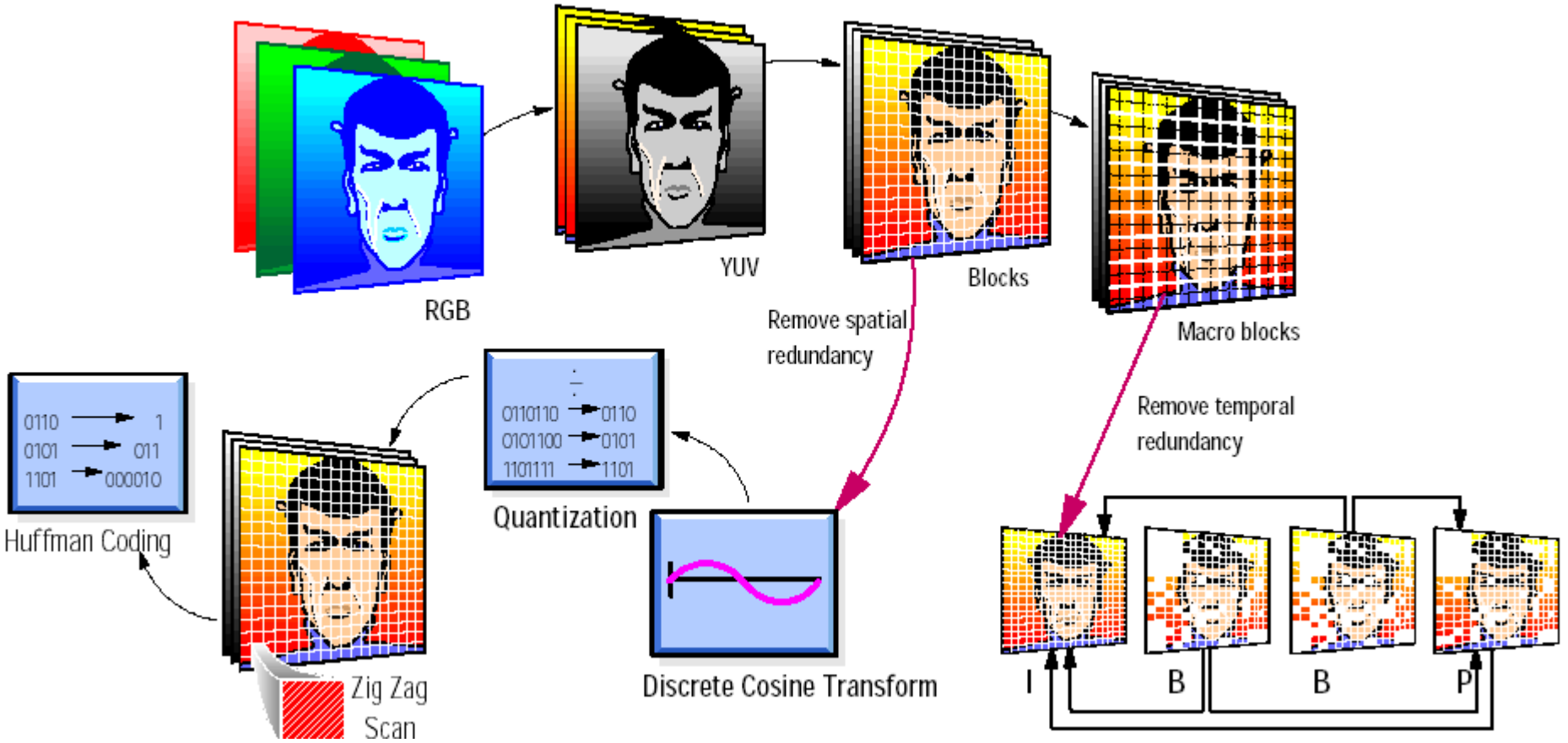


The block diagram of the MPEG encoder



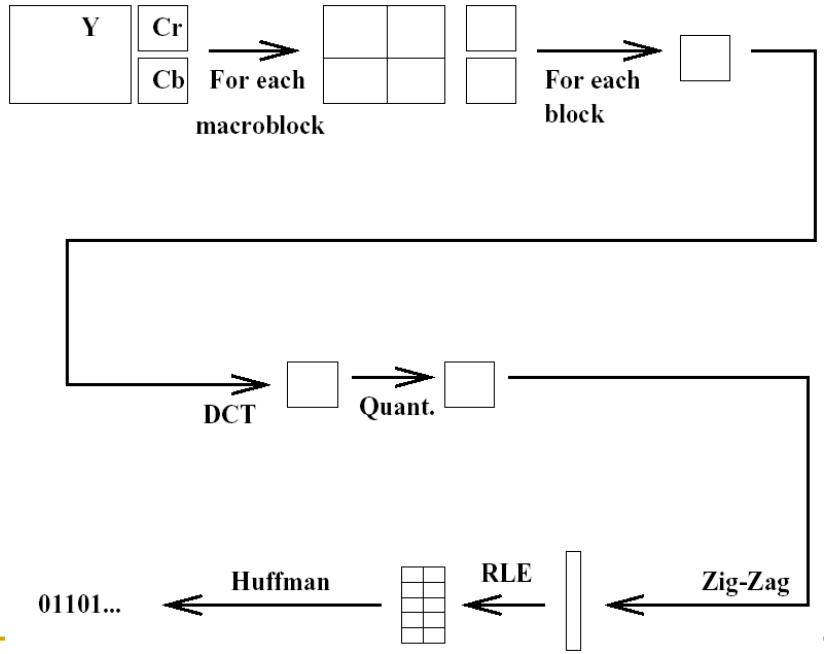
- I-frame: contains the full image
- P-frame is based on preceding I o P-frame
- B-frame uses past or future I o P frames

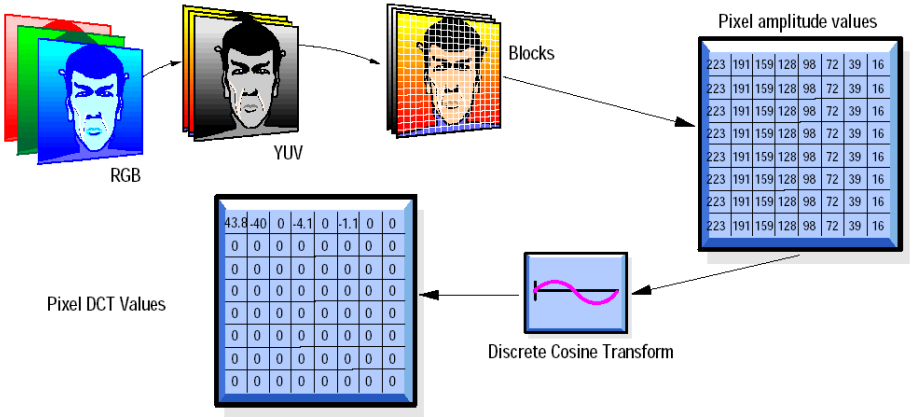
MPEG Coding

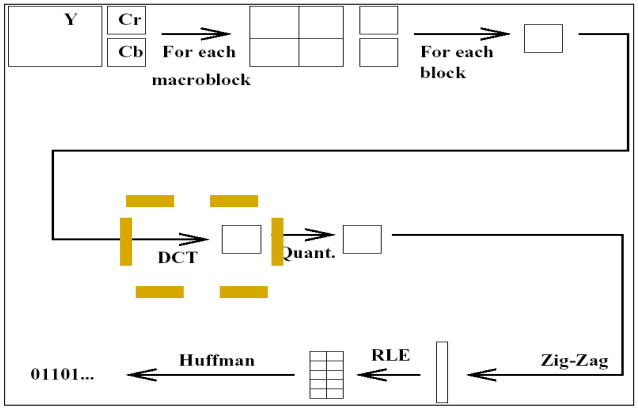

RGB
YUV
Blocks
Macro blocks
Remove spatial redundancy
Remove temporal redundancy
Quantization
Discrete Cosine Transform
Zig Zag Scan
Huffman Coding
0110 → 1
0101 → 011
1101 → 000010
0110110 → 0110
0101100 → 0101
1101111 → 1101
I
B
B
P

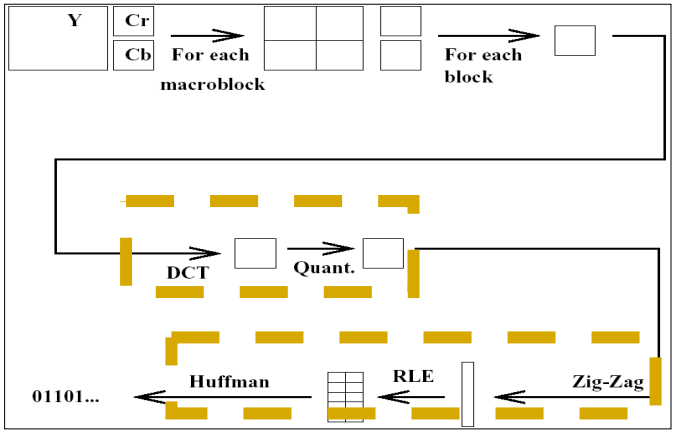
I-frame coding

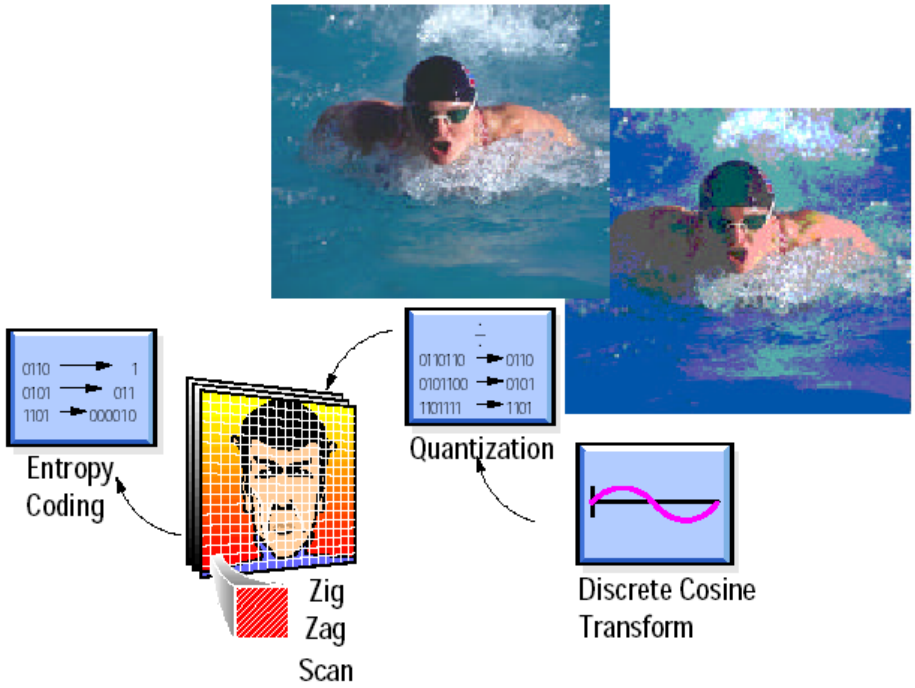
- Intra blocks are processed through DCT 8x8 (lossless)
- DCT coefficient quantization (lossy)
- zig-zag scanning
- DC (RLE) and AC (DPCM) coding
- Entropy coding (Huffman)

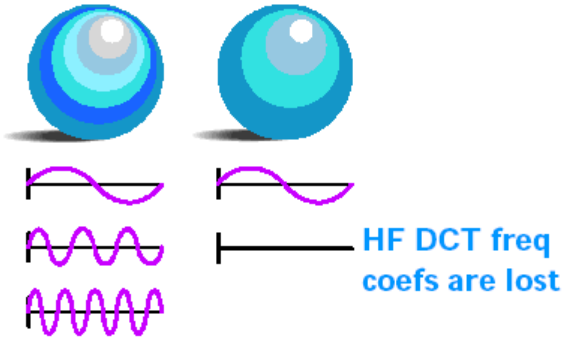



RGB
YUV
Blocks
Pixel amplitude values
223 191 159 128 98 72 39 16
Discrete Cosine Transform
Pixel DCT Values
43.8 -40 0 -4.1 0 -1.1 0 0


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


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

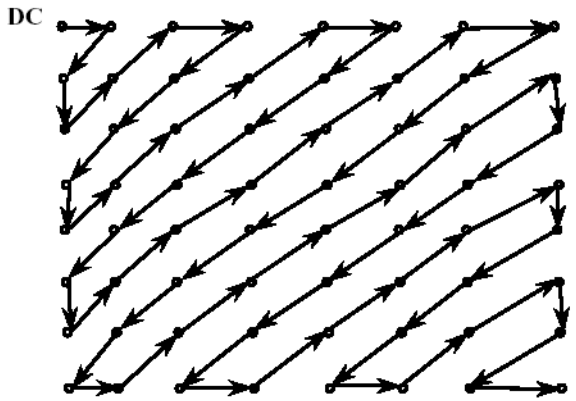

0110 → 1
0101 → 011
1101 → 000010
Entropy
Coding
Zig
Zag
Scan
0110110 → 0110
0101100 → 0101
1101111 → 1101
Quantization
Discrete Cosine
Transform


HF DCT freq
coefs are lost

- Quantization is allowed to differ between slices and macroblocks. Allows to define a scaling factor. The default quantization matrix can be changed for each sequence.

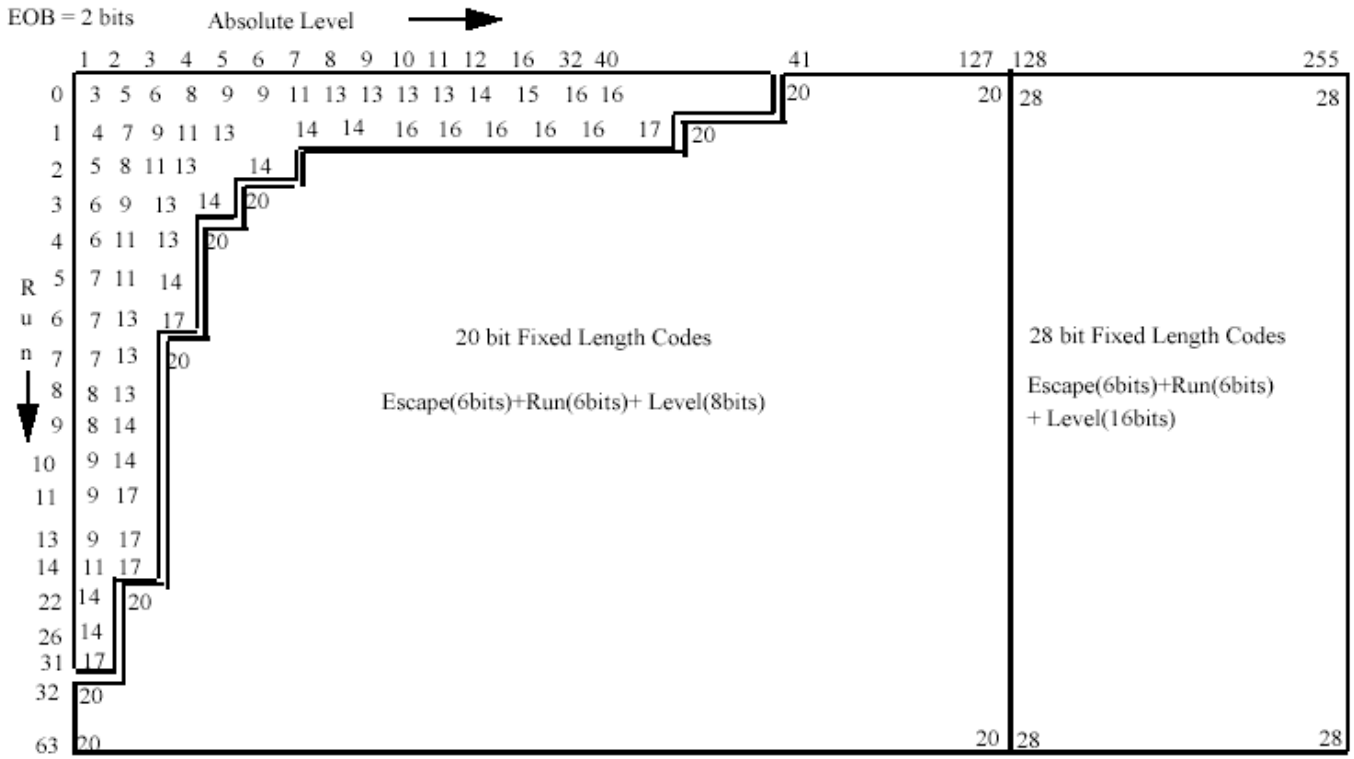
8	16	19	22	26	27	29	34
16	16	22	24	27	29	34	37
19	22	26	27	29	34	34	38
22	22	26	27	29	34	37	40
22	26	27	29	32	35	40	48
26	27	29	32	35	40	48	58
26	27	29	34	38	46	56	69
27	29	35	38	46	56	69	83

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

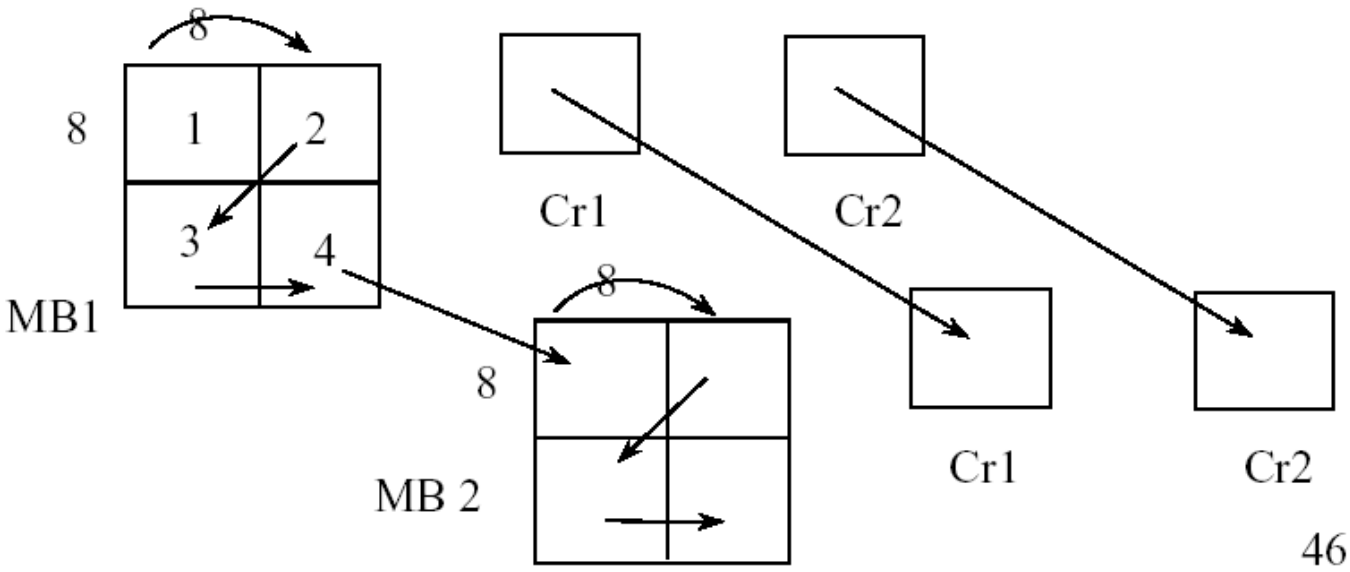


- AC coefficients are encoded losslessly according to *run level encoding* and Huffman coding (VLC: variable length coding)

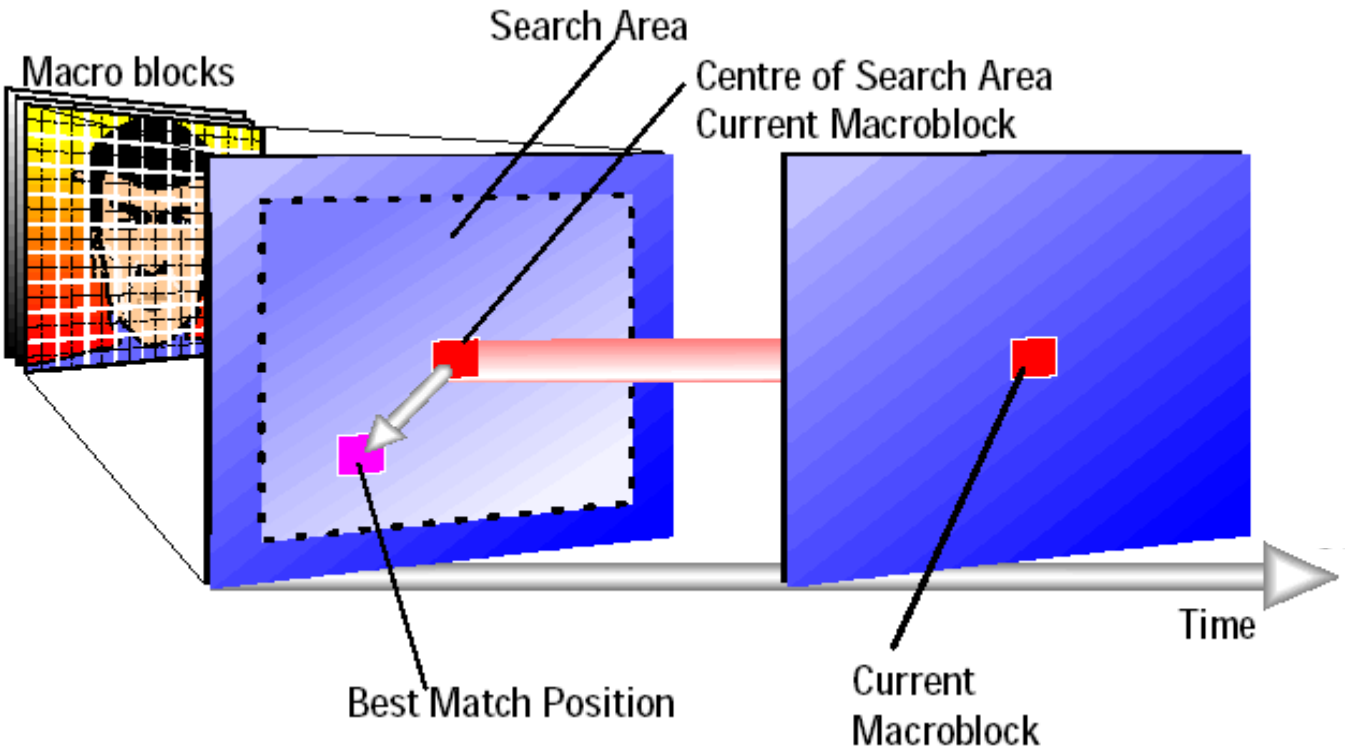
- *run length* and *level tables* are formed on a statistical *basis*



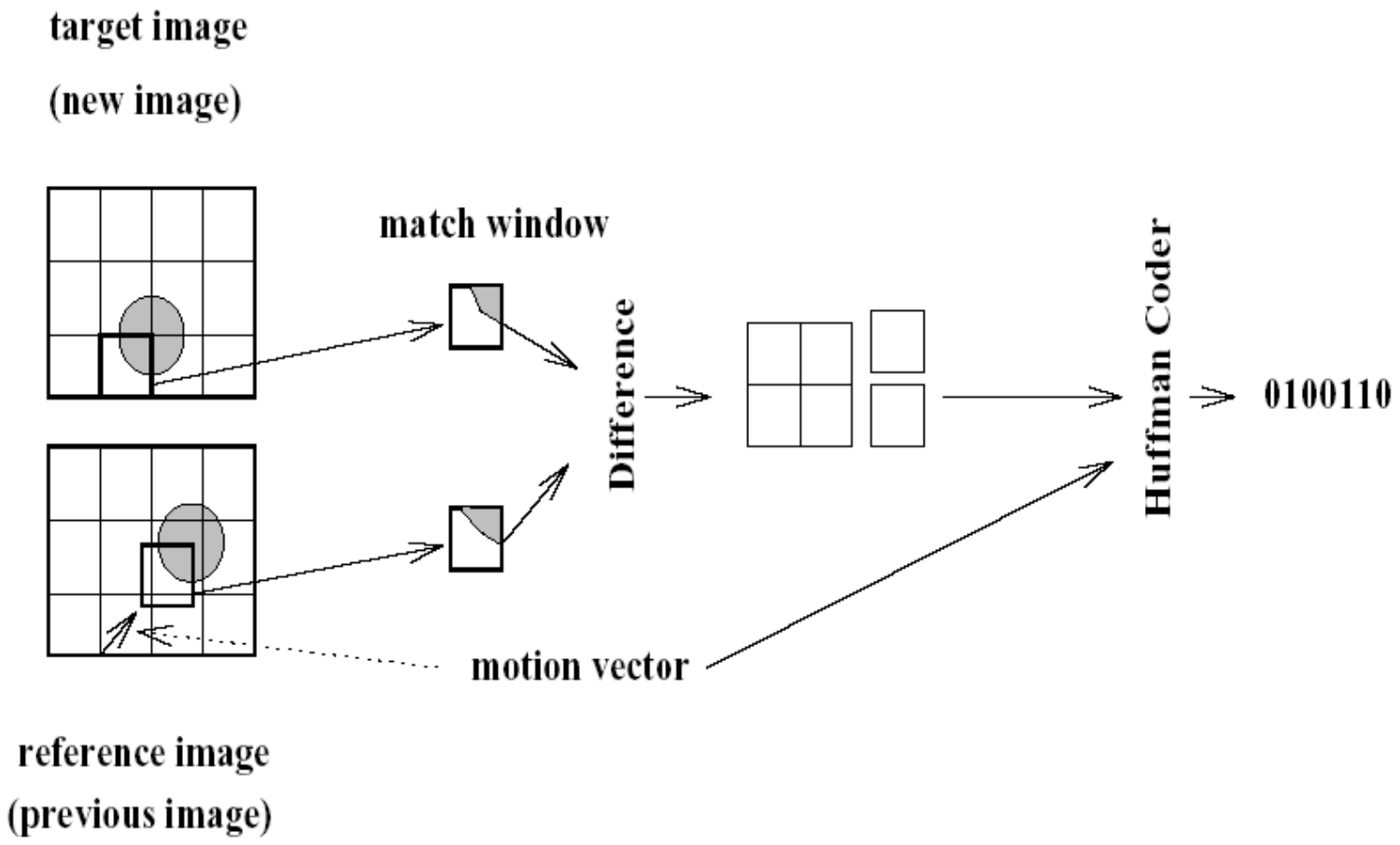
- DC si coefficients encode differences between blocks of the macroblock
  - The initial block value is 1024
  - Different tables for Y and CbCr



- Slightly better than JPEG. However it motion estimation that produces effective compression.

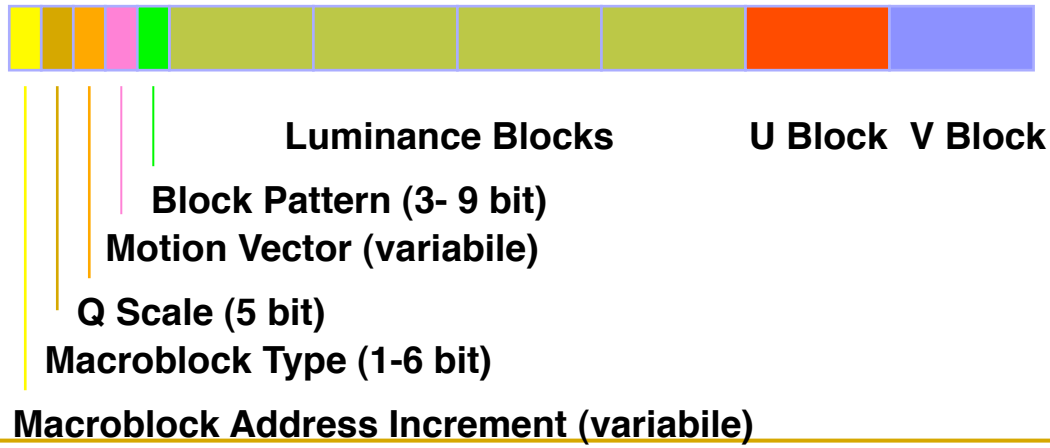


P-frame coding

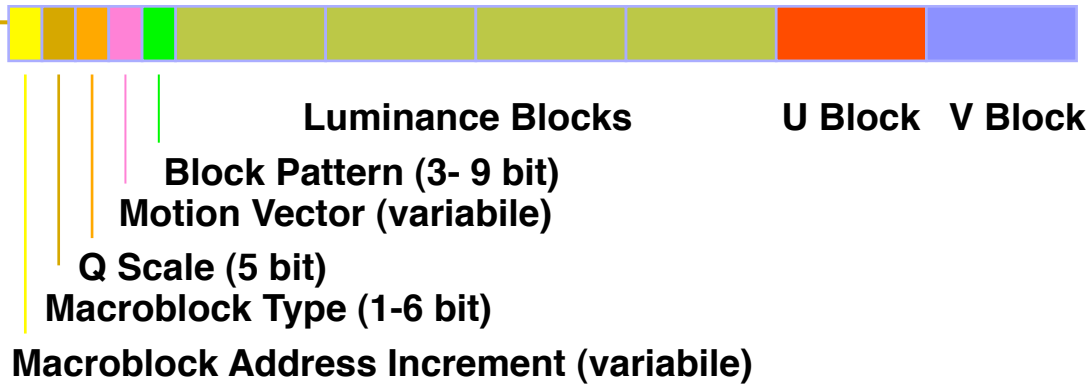


- Per le P-frame è compito dell'encoder scegliere se codificare un macroblocco come intra o come predetto. Un possibile meccanismo compara la varianza della componente luminanza del macroblocco originale con il macroblocco errore. Se la varianza dell'errore predetto è maggiore il macroblocco è codificato intra.

- L'informazione a livello di macroblocco è codificata in una stringa:



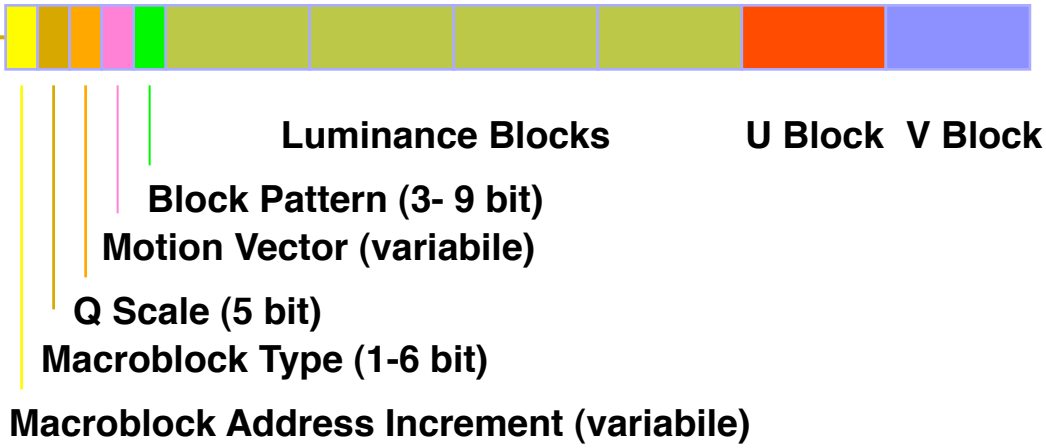
Address Increment



- Ogni macroblocco ha un indirizzo.
  - MB_ADDR = MB_ROW * MB_WIDTH + MB_COL
    - MB_WIDTH = luminance width / 16
    - MB_ROW = # riga pixel alto a sx/ 16
    - MB_COL = # col. pixel alto a sx / 16
- Il decoder mantiene l'indirizzo del macroblocco precedente PREV_MBADDR.
  - Impostato a **-1** all'inizio del frame.
  - Impostato a **(SLICE_ROW * MB_WIDTH-1)** all'inizio dell'header dello slice.
- L'indirizzo dell'incremento del macroblocco è sommato a PREV_MBADDR per dare l'indirizzo del macroblocco corrente.

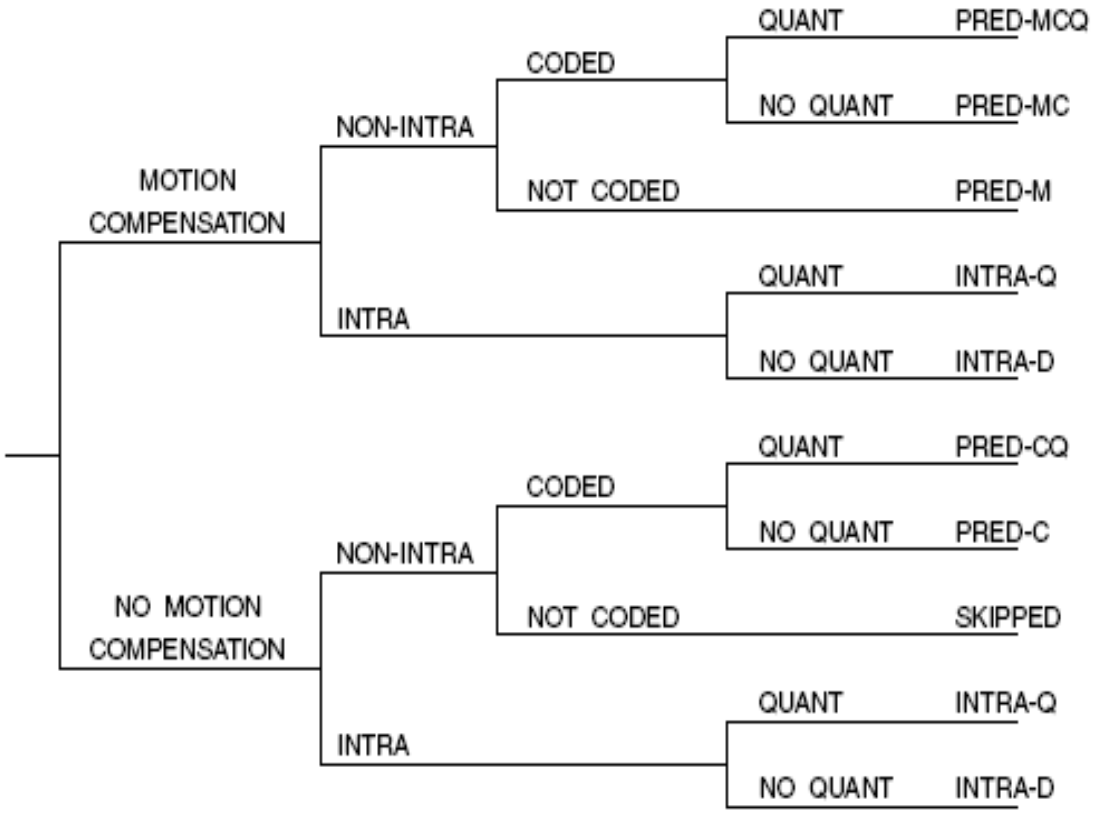
- L’Address Increment è codificato con Huffman, secondo una tabella predefinita, la stessa usata per le I frame:
  - 33 codici (1-33).
    - 1 il più piccolo (1-bit)
    - 33 il più grande (11-bit)
  - 1 codice di ESCAPE
    - ESCAPE significa: aggiungi 33 al codice di incremento indirizzo che segue.
    - Si possono usare più ESCAPE in sequenza per codificare distanze grandi.

Macroblock Type



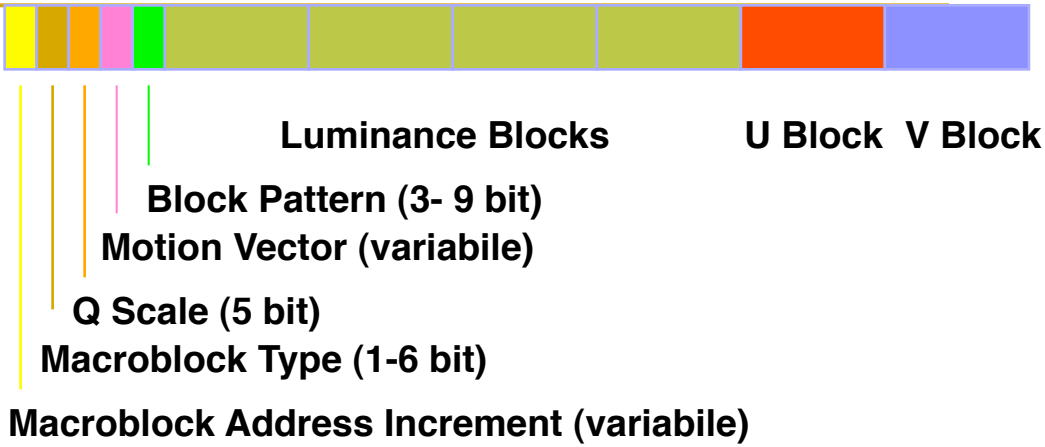
- Il Macroblock Type determina se il macroblocco è di tipo Intra o no, se esistono o no per il macroblocco Q Scale, Motion Vector, e Block Pattern. E' codificato con Huffman.

- Non tutte le combinazioni sono possibili. Ci sono 8 possibili macroblock type (1 - 6 bit).



Albero di scelta dell'encoder per selezionare il tipo di macroblocco

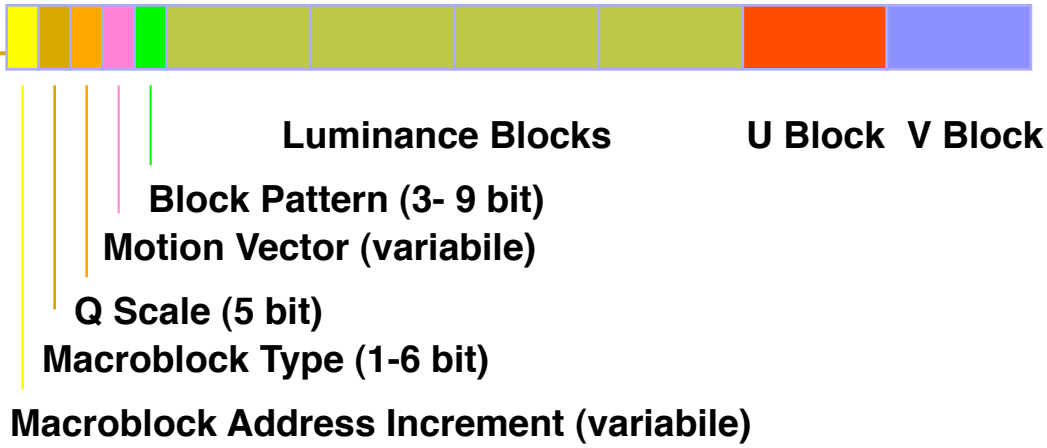
Quantization Scale



- Codifica valori tra 1 e 31 che sono interpretati come valori tra 2 e 62 per il fattore Q Scale (non si codificano valori dispari). Il valore Zero è illegale. Impiega 5 bit. .

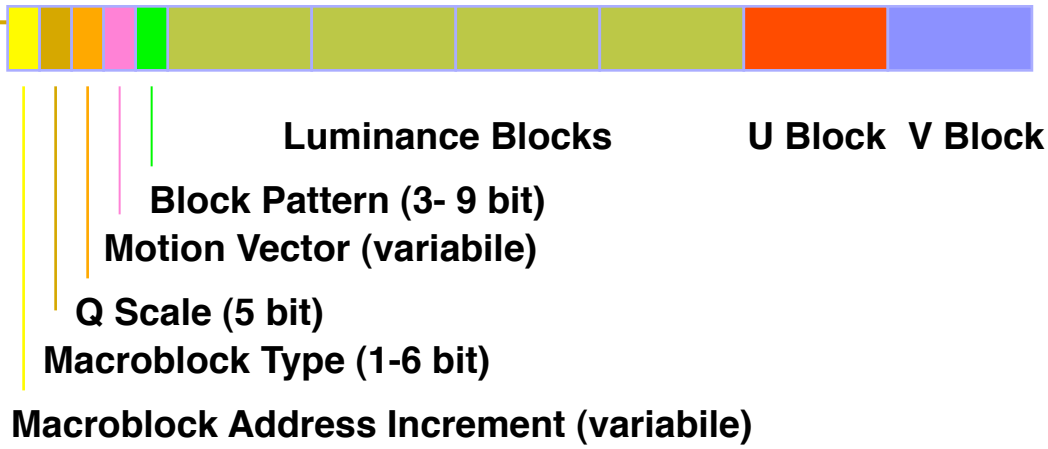
- Il decoder mantiene il q-scale corrente se non è specificato un nuovo q-scale. Altrimenti rimpiazza il q-scale con il nuovo valore indicato.

Motion Vector



- Il Motion Vector è usato per definire una base predittiva per il macroblock corrente partendo dall'immagine di riferimento. La predizione si usa per la determinazione dei vettori di moto. La differenza tra predittore e valore del vettore è codificata con Huffman.
- E' specificato con due componenti (offset orizzontali e verticali). Prima si indica la componente orizzontale e poi la verticale. L'assenza di motion vector è specificata con il valore (0,0).
  - L'offset è calcolato a partire dal pixel in alto a sx. del MB:
    - Valori positivi indicano in alto e a dx.
    - Valori negativi indicano in basso e a sx.
  - E' impostato o 0,0 all'inizio di frame o di slice o all'inizio di MB tipo I.
- I Macroblock di tipo P hanno sempre una base predittiva scelta secondo il vettore di moto. Se il vettore di moto è (0,0) la base predittiva è lo stesso macroblocco nel fotogramma di riferimento.

Block Pattern

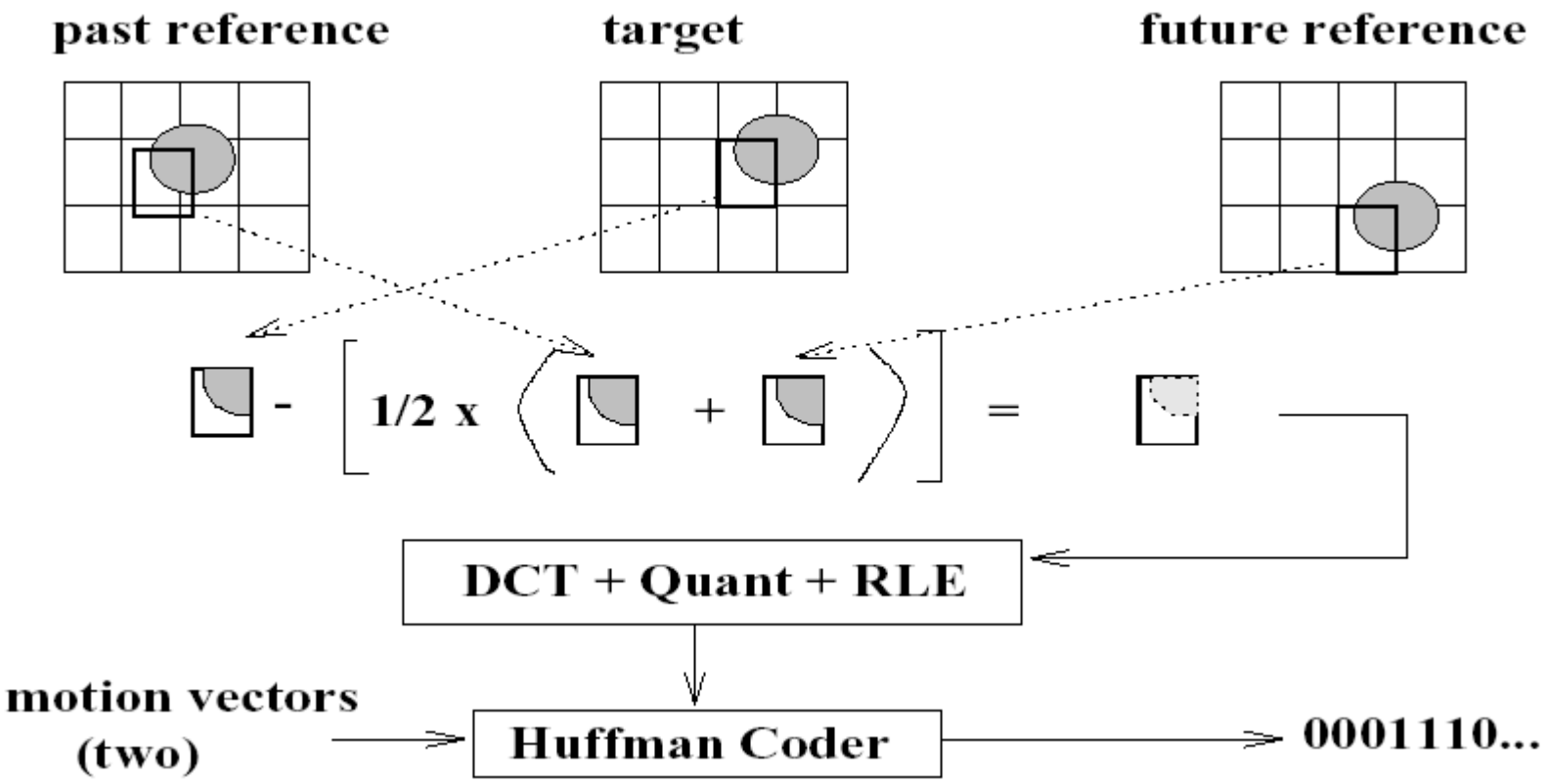


- Il Block Pattern indica quali blocchi hanno un errore abbastanza grande rispetto al blocco di riferimento da dover essere codificato (compensazione del moto). Lo scopo della compensazione di moto è quello di trovare una base predittiva che assomigli il più possibile al macroblocco in analisi.

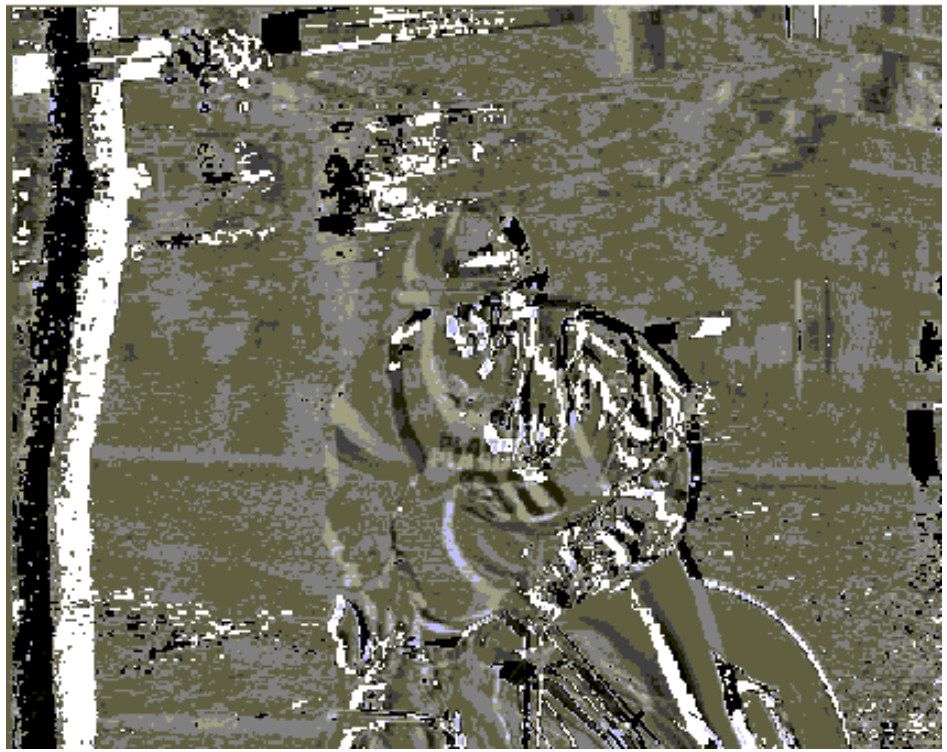
- Se non c'è il block pattern allora significa che il match tra il blocco corrente e il corrispondente di riferimento è particolarmente buono e non c'è bisogno di codificare qualcosa. In un macroblocco P uno o anche tutti I blocchi possono essere assenti.

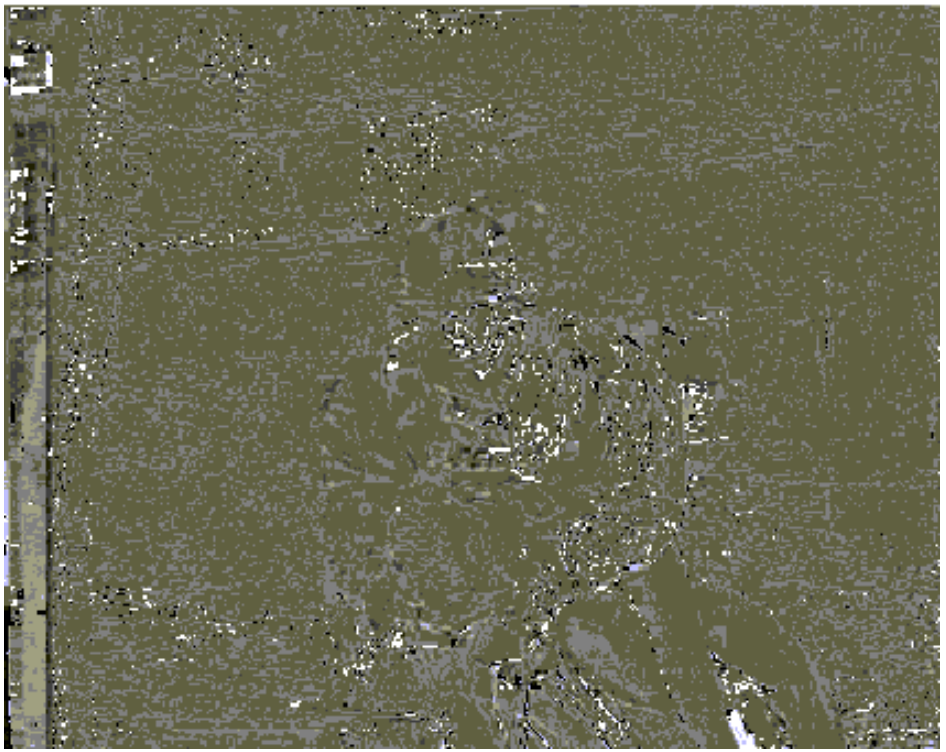
- La differenza tra blocco di riferimento e blocco codificato è trattata come un blocco normale, spesso definito come error block
- Si usa una matrice di quantizzazione diversa rispetto agli I-block:
  - Ha il valore “16” in tutte le posizioni.poichè gli error block hanno molta informazione in alta frequenza.
  - Non c’è una buona correlazione percettiva tra le frequenze dell’error coding ed eventuali artefatti di compressione.
- I termini sono codificati RLE dopo scansione zig-zag e quindi codificati con Huffman. La componente DC è trattata come le AC:
  - Non si usa differential encoding rispetto ad un predittore.
- I predittori DC sono resettati quando è incontrato un MB tipo P o skipped.
- I predittori dei motion vector sono resettati quandi si incontra un MB di tipo I

B-frame coding



- Frame N da codificare.
- Frame all’istante N-1 usato per la predizione del contenuto del frame N.





- Immagini di errore di predizione senza motion compensation.
- Immagini di errore di predizione da codificare con motion compensation.

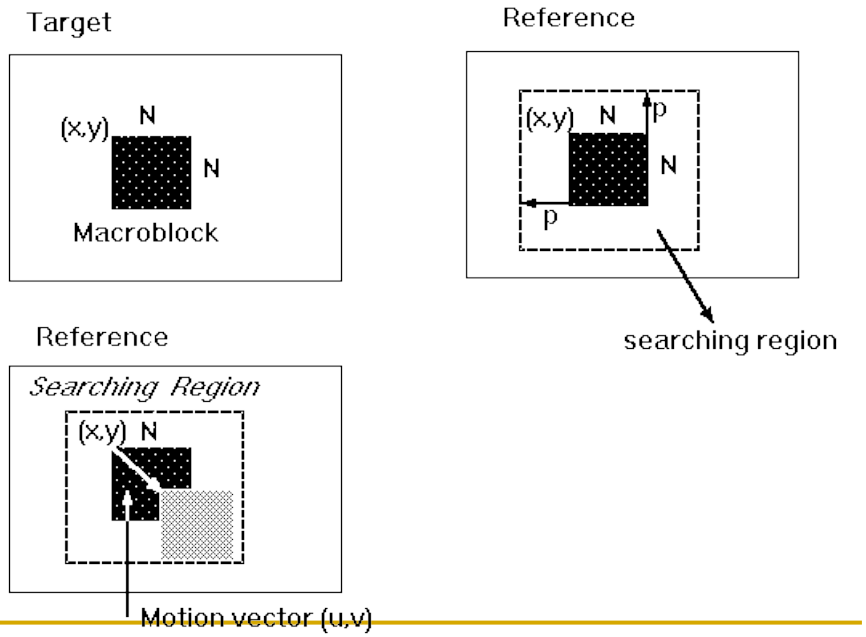
Esercizio consigliato

1. Prendere VCDemo: http://ict.ewi.tudelft.nl/index.php?Itemid=124
2. Aprire un file MPEG (es. bike.mpg)
3. Impostare l'output su Coded difference e aggiungere il display dei vettori di moto
4. Visualizzare il filmato
5. Impostare l'output su Frame prediction e togliere i vettori di moto
6. Visualizzare il filmato facendo particolare attenzione ai bordi delframe e della moto

Block matching

- Esistono diverse tecniche per il block matching. Possiamo scegliere tra:
  - Tecniche per la determinazione del match tra blocchi
  - Strategie di ricerca dei blocchi
  - Scelte dimensione dei blocchi

- Spesso si limita l'area in cui cercare il match

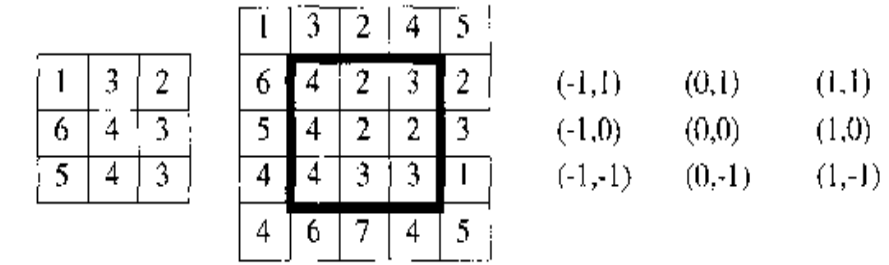


- Mean Squared Error (MSE) (per un blocco N x N):

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

dove C_{ij} è il campione del blocco corrente e R_{ij} il campione del blocco di riferimento

- Es:



- L' MSE tra il blocco corrente e la stessa posizione (posizione *(0,* 0)) sul riferimento è dato da:

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

- Tutti i valori sono:

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

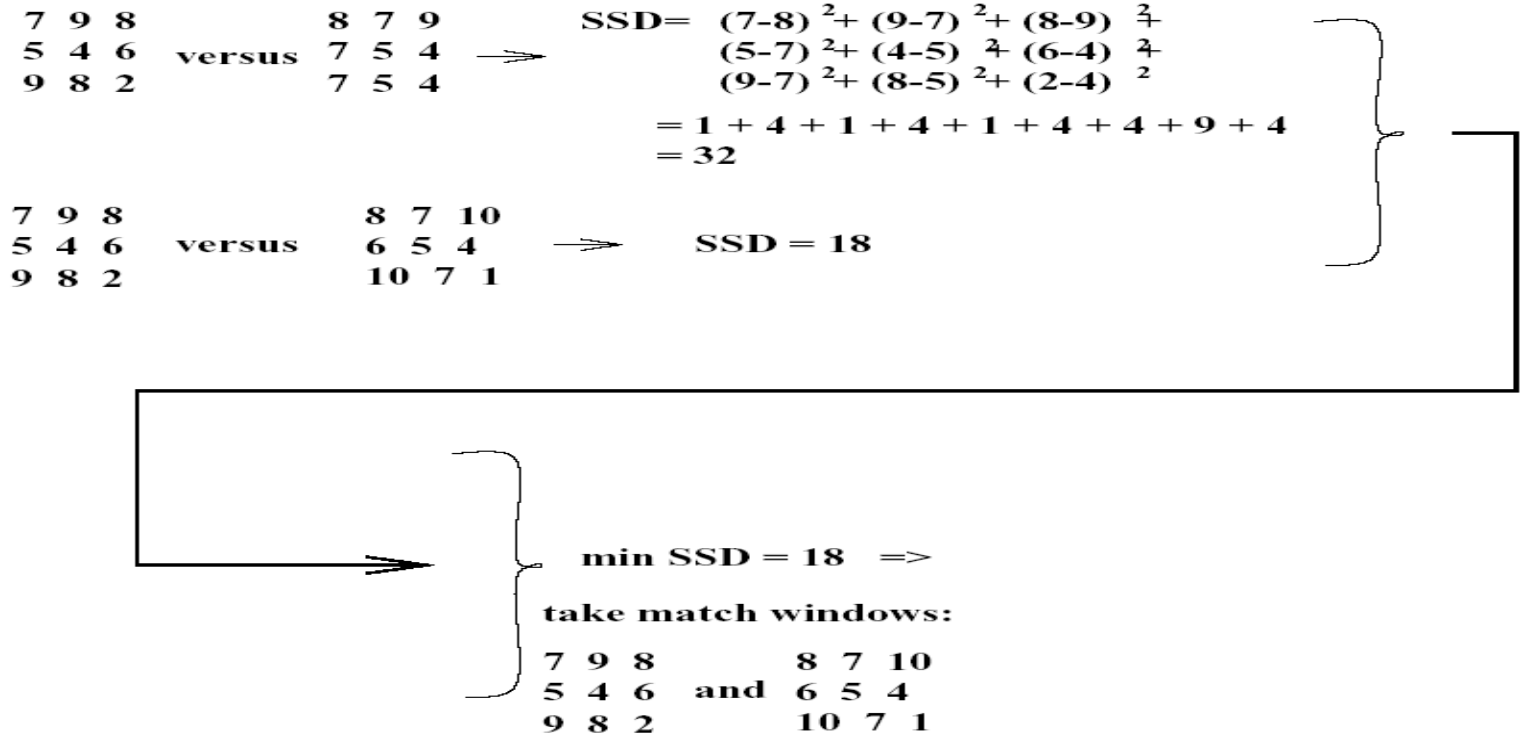
- Mean absolute error/difference (MAE/MAD)
  - È più facile/veloce da calcolare rispetto a MSE ed è ancora un'approssimazione ragionevole

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

- Matching pel count (MPC)
  - Si contano il numero di pixel simili in due blocchi
  - Devono essere scelte una metrica ed una soglia per valutare la distanza tra due pixel

- Sum of Squared Differences (SSD)

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



- Sum of absolute errors (SAE) o sum of absolute differences (SAD)

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

rispetto a SSD è meno sensibile ad eventuali outlier: un solo punto molto diverso rende il valore di SSD molto grande

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:

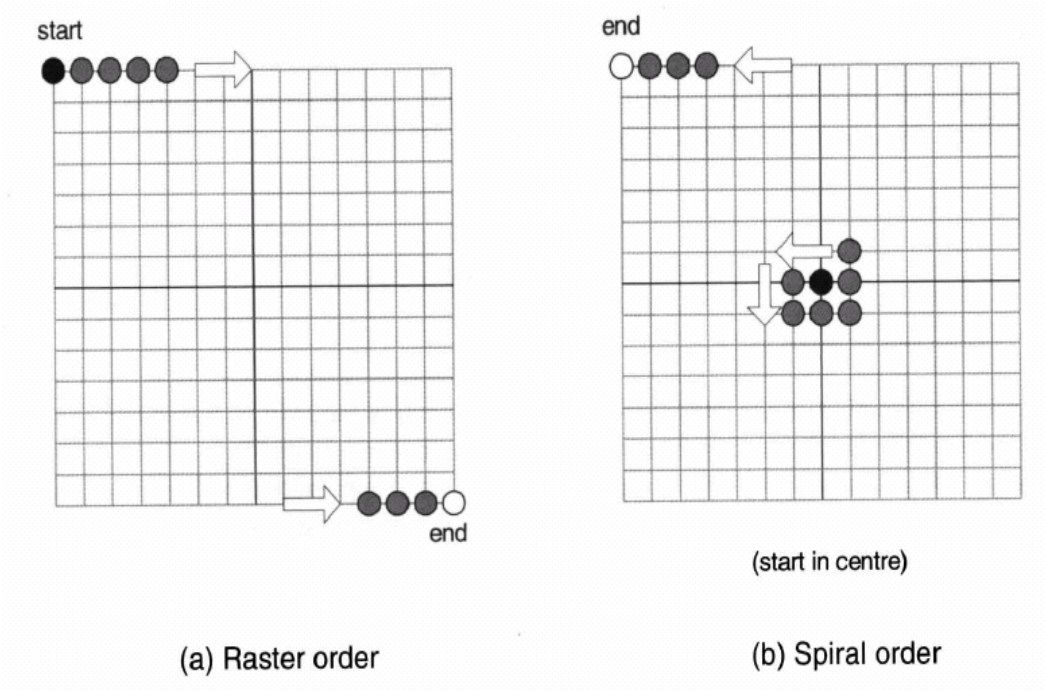
7 9 8		8 7 10	
5 4 6	versus	6 5 4	-> SAD = 211
9 8 2		10 7 202	

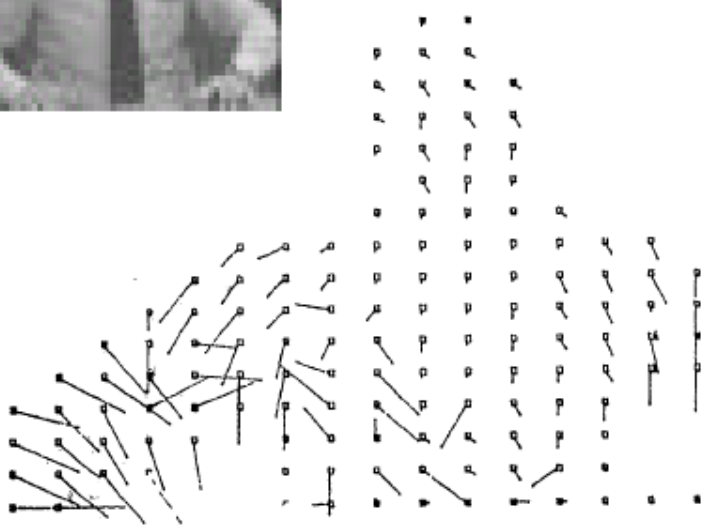
Algoritmi di ricerca

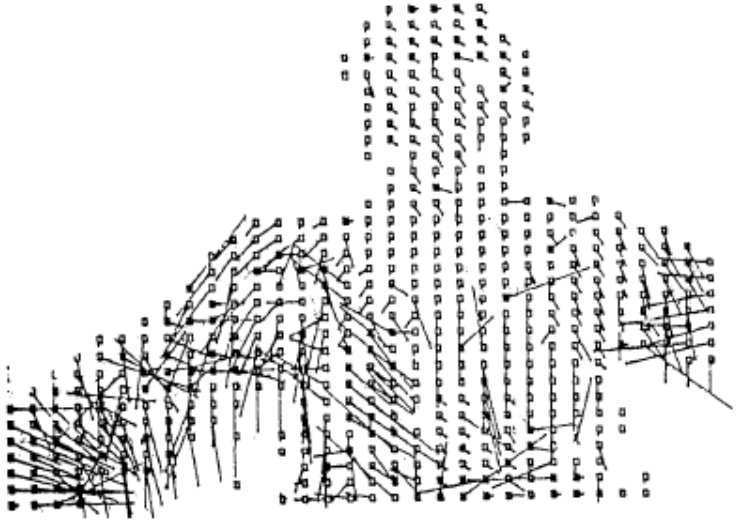
- La dimensione della finestra di ricerca dipende da diversi fattori:
  - Risoluzione dei frame
  - Potenza di calcolo disponibile
  - Tipo di scena

Full search

- Usando un criterio di comparazione tra blocchi (es. SAE/SAD) cerco in tutte le possibili posizioni della finestra
  - È computazionalmente costosa, adatta per implementazioni h/w







Blocchi grandi

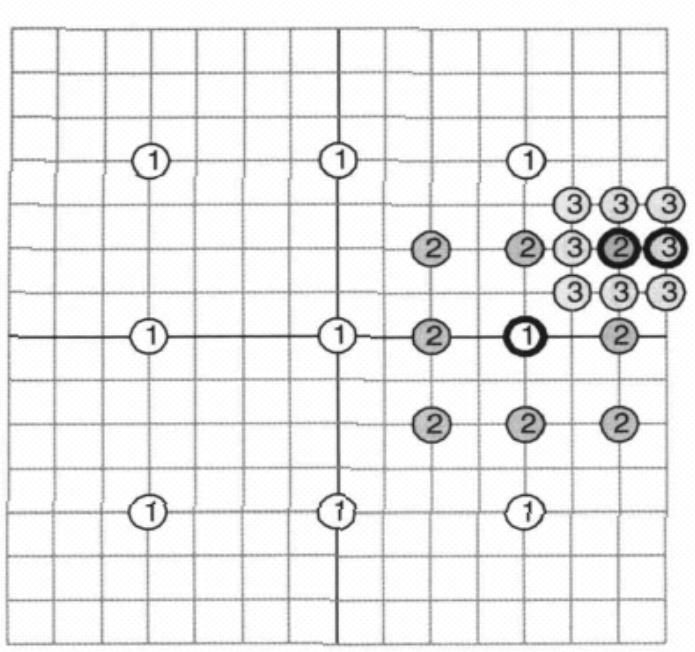
Blocchi piccoli

Ricerca veloce

- Cerco di ridurre il numero di comparazioni rispetto a quelle necessarie per full search
  - Con full search trovo il minimo globale di SAE
  - Con fast search rischio di finire in minimi locali

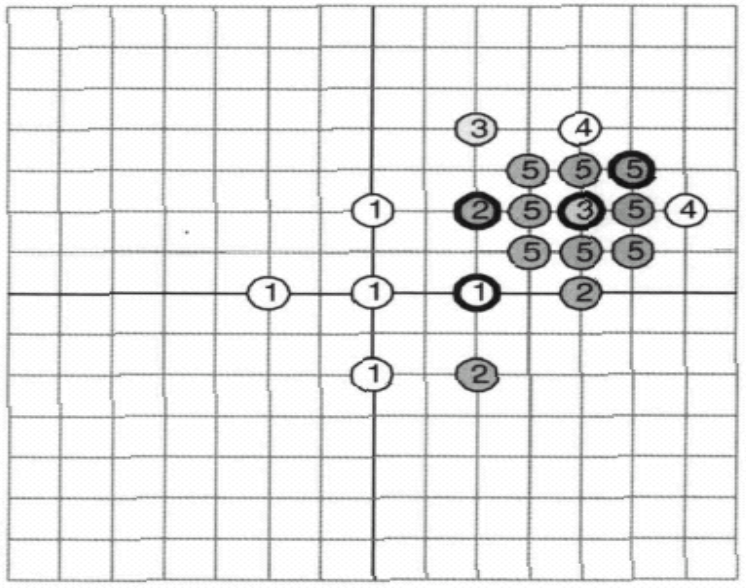
Three step search (TSS)

1. Comincio a cercare da (0, 0).
2. Pongo S = 2^{N-1} (dimensione passo).
3. Cerco nelle 8 locazioni a +/-S pixel di distanza attorno a (0, 0).
4. Tra le 9 locazioni analizzate prendo quella con SAE minore e la faccio diventare il nuovo centro di ricerca.
5. Pongo *S* = S/2.
6. Ripeto I passi da 3 a 5 finché *S* = 1.



Logarithmic Search

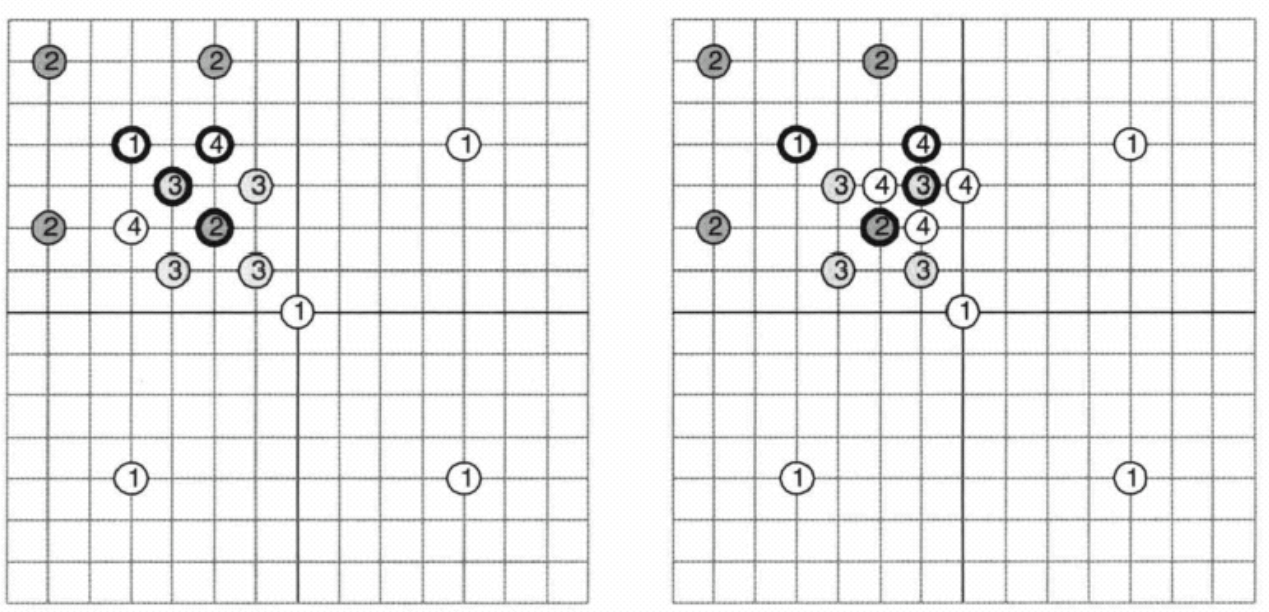
1. Comincio a cercare da (0, 0).
2. Cerco nelle 4 posizioni adiacenti in orizzontale e in verticale, a S pixel di distanza dall'origine (con S passo di ricerca iniziale). Le 5 posizioni modellano un ' + '.
3. Impostare la nuova origine in corrispondenza del match migliore. Se il match migliore è la posizione centrale del '+' allora *S* = S/2, altrimenti *S* rimane uguale.
4. Se *S* = 1 vai al punto *5* , altrimenti vai al punto 2.
5. Cerca le 8 posizioni attorno al best match. Il risultato finale è il best match tra queste 8 posizioni e la posizione centrale.



Cross-Search

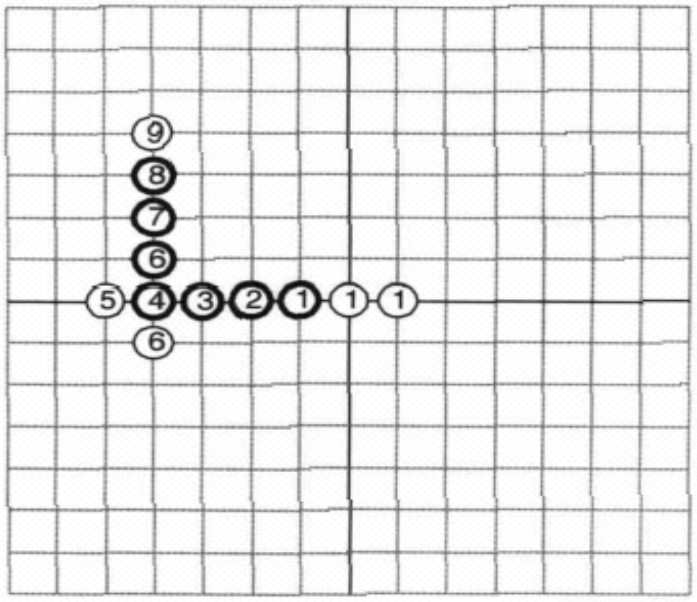
- E’ simile a TSS tranne che ad ogni passo si comparano 5 punti (che formano una **X)** invece di 9 punti

1. Comincio a cercare da (0, 0).
2. Cerco nelle 4 posizioni a +/-S pixel di distanza, che formano una ‘X’ (con S = 2^N*-1* come in TSS).
3. Imposto la nuova origine in corrispondenza al best match tra I 5 punti.
4. Se S > 1 allora S = S/2 e vado al passo 2; altrimenti vado al passo 5.
5. Se il best match è in alto a sx. o in basso a dx della ‘X’, valuto altri 4 punti che formano una ‘X’ ad una distanza di +/-1; altrimenti (best match in alto a dx o basso a sx) valuto altri 4 punti che formano un ‘ + ’ ad una distanza di +/-1.



One-at-a-Time Search

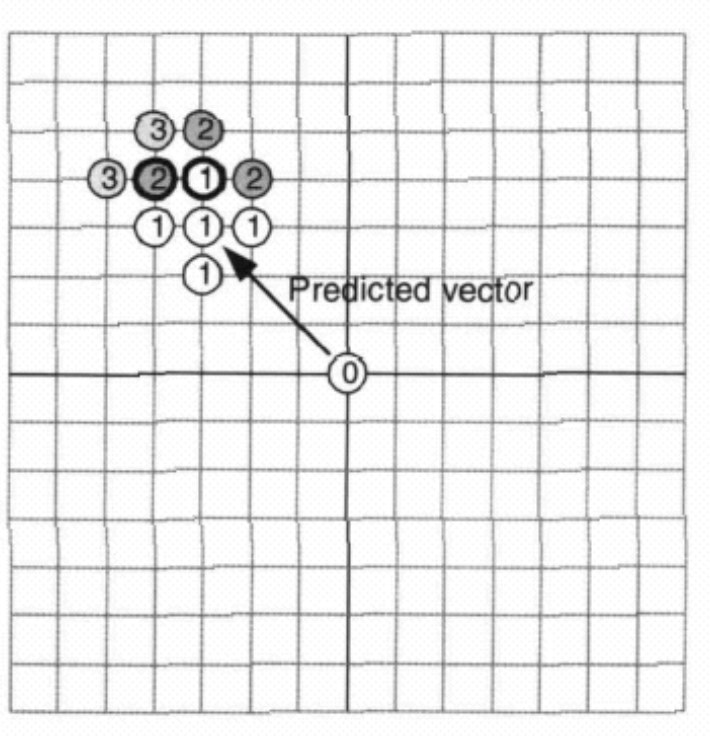
1. Comincio a cercare da (0, 0).
2. Cerco nell'origine e nei due vicini orizzontali
3. Se l'origine ha il SAD più basso allora vado al passo, altrimenti. . . .
4. Imposto l'origine nel punto orizzontale con il SAD più piccolo e cerco nel vicino in cui ancora non avevo cercato. Vado al passo 3.
5. Ripeto i passi da 2 a 4 nella direzione verticale.



Nearest Neighbours Search

- Algoritmo proposto per H.263 e MPEG-4: ogni vettore di moto viene predetto dai vettori vicini (già codificati). Tiene conto del fatto che:
  - I macroblocchi vicini hanno spesso vettori di moto simili (un predittore basato su mediana è vicino al 'vero' best match)
  - Un vettore vicino alla mediana verrà codificato con un VLC piccolo.

1. Comincio a cercare da (0, 0).
2. Pongo l'origine della ricerca nella posizione indicata dal vettore predetto e cerco a partire da questa nuova posizione.
3. Cerco nei 4 vicini in forma di '+ '.
4. Se l'origine della ricerca (o la posizione 0, 0 della prima iterazione) fornisce il risultato migliore prendo il risultato; altrimenti imposto la nuova origine nel best match e vado al passo 3. L'algoritmo si ferma quando il best match è al centro di un ' + ' (o quando si è raggiunto il bordo della finestra di ricerca).

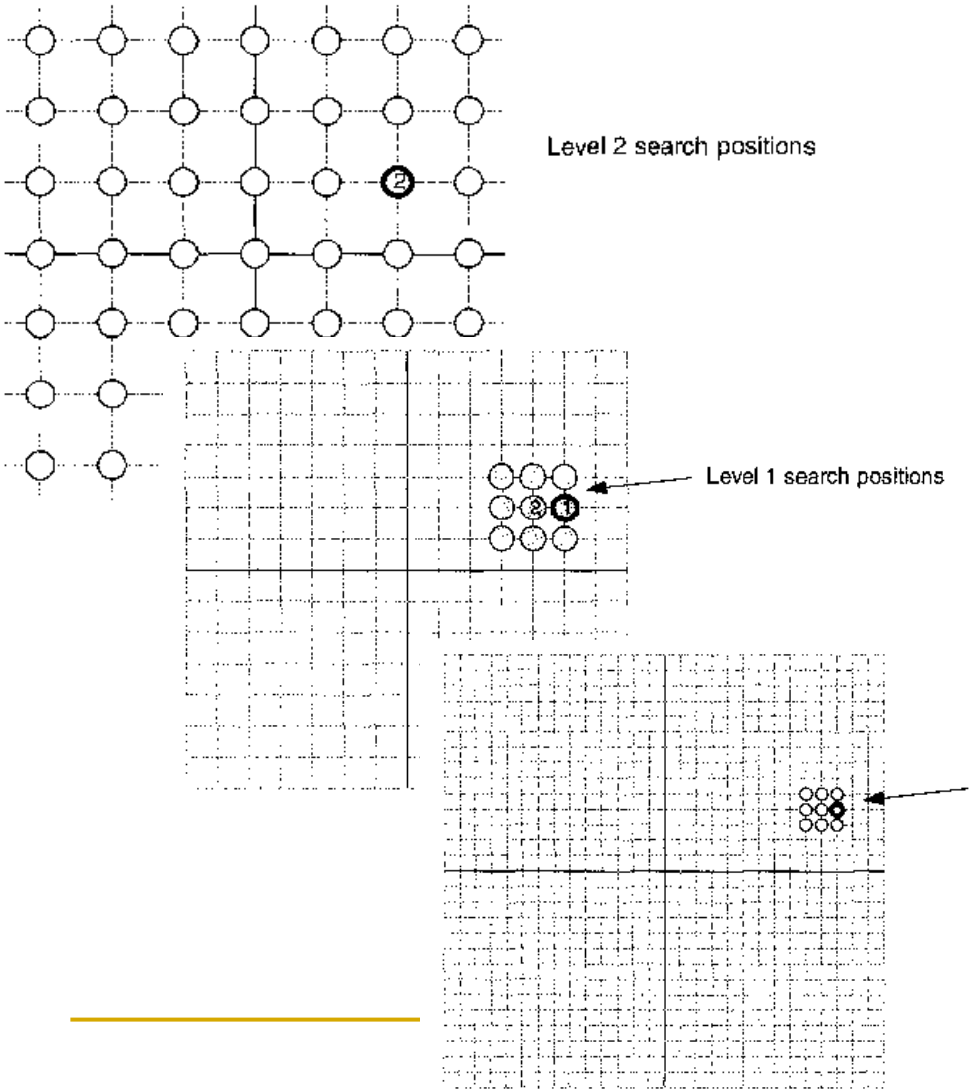

3
2
3
2
1
2
1
1
1
1
Predicted vector
0

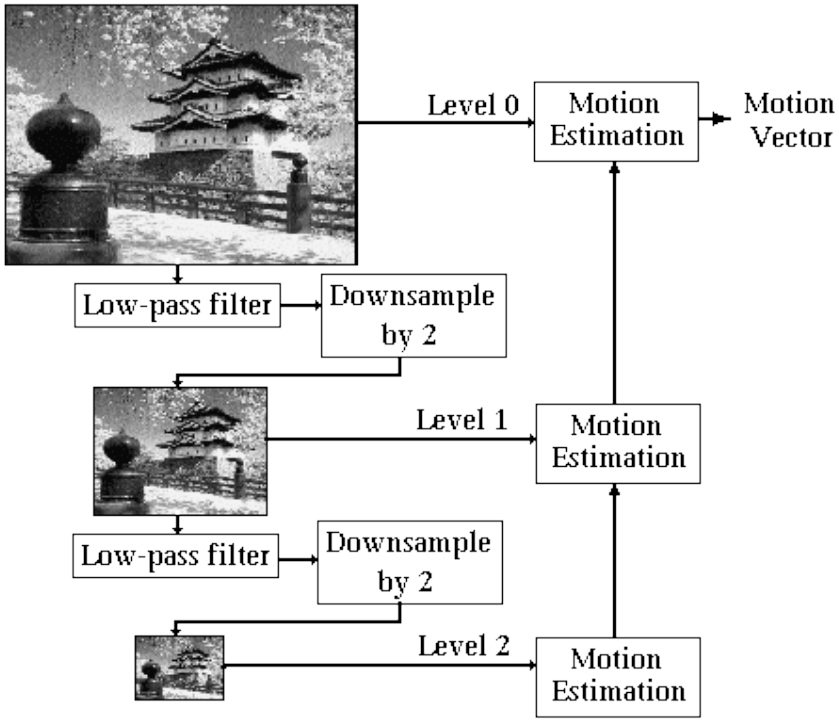
- L’algoritmo funziona bene quando i vettori di moto sono ragionevolmente omogenei
  - Non ci devono essere troppi cambiamenti grossi nel campo dei vettori di moto.
- Vengono comunque seguiti due accorgimenti per migliorare il risultato.
  - Se il predittore di mediana non può essere molto accurato (es. perché troppo macroblocchi vicini sono intra-coded e quindi senza MV), si usa un algoritmo alternativo come TSS.
  - Si usa una funzione di costo che stima se è ragionevole il costo computazionale di un altro set di ricerche.

Hierarchical Search

- Effettua la ricerca in una versione sottocampionata dell'immagine, la ricerca viene quindi raffinata usando versioni dell'immagine a risoluzione maggiore, finché non si arriva alla risoluzione originale

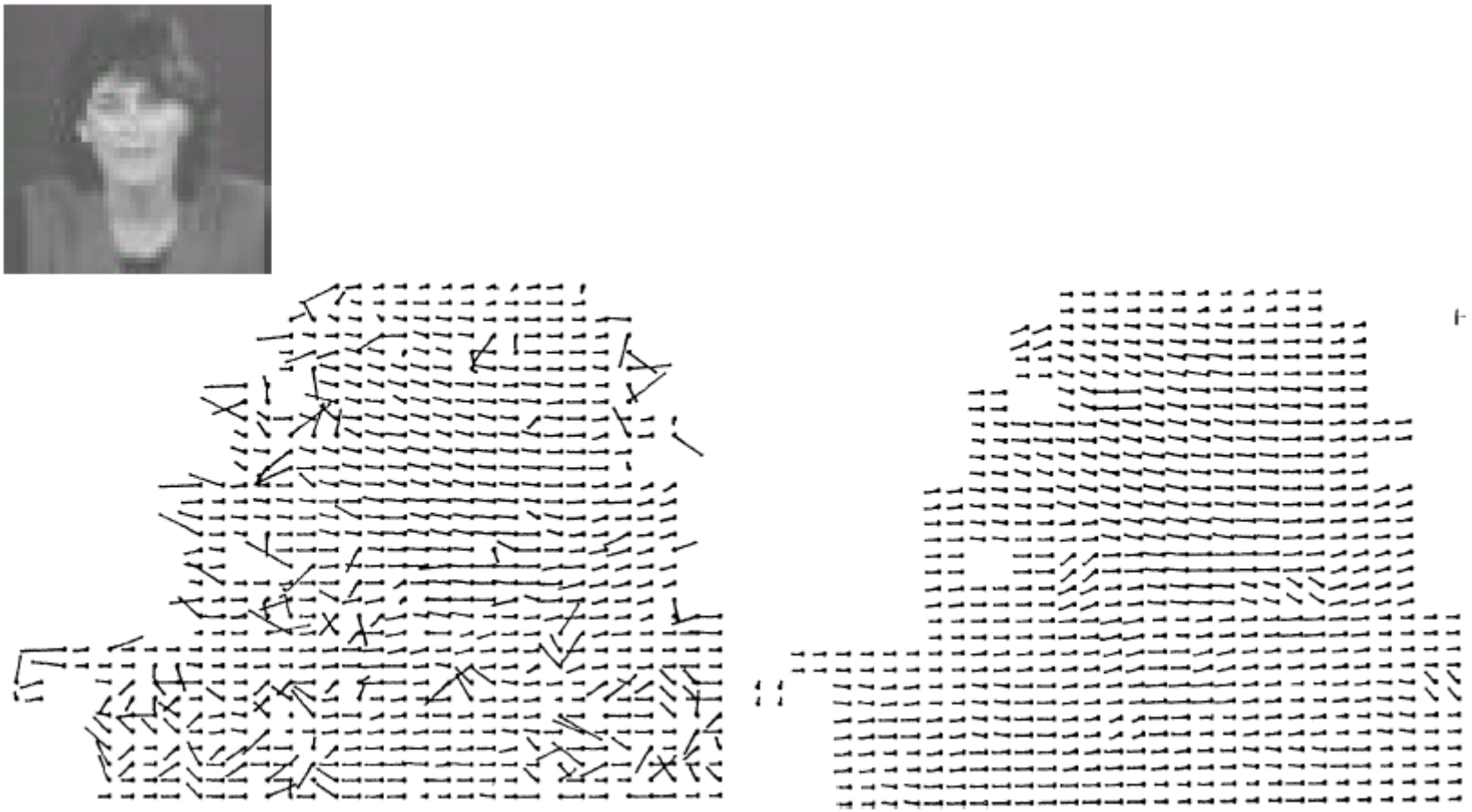
1. Il livello 0 consiste nell'immagine corrente ed in quella di riferimento a piena risoluzione. Si sottocampiona il livello 0 di un fattore 2 sia in orizzontale che verticale, producendo il livello 1.
2. Si ripete il subsampling del livello 1 per produrre il livello 2, e così via finché si raggiunge il numero di livelli necessari (tipicamente 3 o 4 livelli bastano).
3. Si inizia a cercare nel livello più alto (risoluzione più bassa) per cercare il best match: questo è il motion vector più 'grezzo'.
4. Si cerca nel livello immediatamente più basso (maggiore risoluzione) intorno alla posizione del vettore di moto 'grezzo' e si cerca il best match.
5. Si ripete il passo 4 finché non si trova il best match al livello 0.





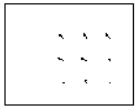
Dirac uses this motion estimation

- Implicitamente si ha uno smoothing dei vettori di moto



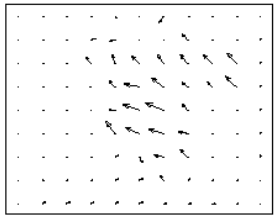
Full search block matcher

Hierarchical block matcher



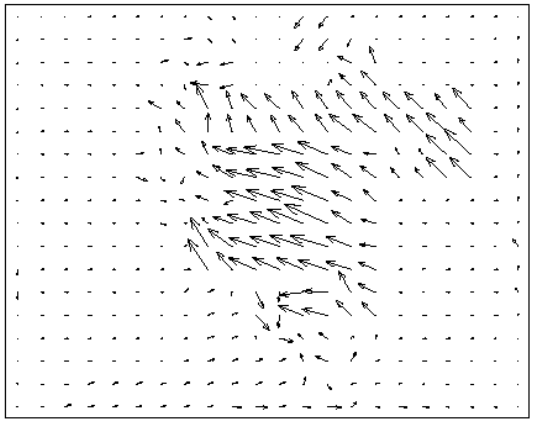
(a)

(b)



(c)

(d)



(e)

(f)

Criteri per uso di algoritmi block matching

1. Matching performance:
   quanto è efficace l'algoritmo nel mimizzare il blocco residuo ?
3. Rate-distortion performance:
   come si comporta complessivamente l'algoritmo a vari bitrate ?
4. Complessità:
   quante operazioni sono necessarie per il block matching ?
5. Scalabilità:
   l'algoritmo funziona bene sia con finestre di ricerca grandi che piccole ?
6. Implementazione:
   l'algoritmo va bene sia che sia implementato in s/w che in h/w?

Comparazione algoritmi block matching

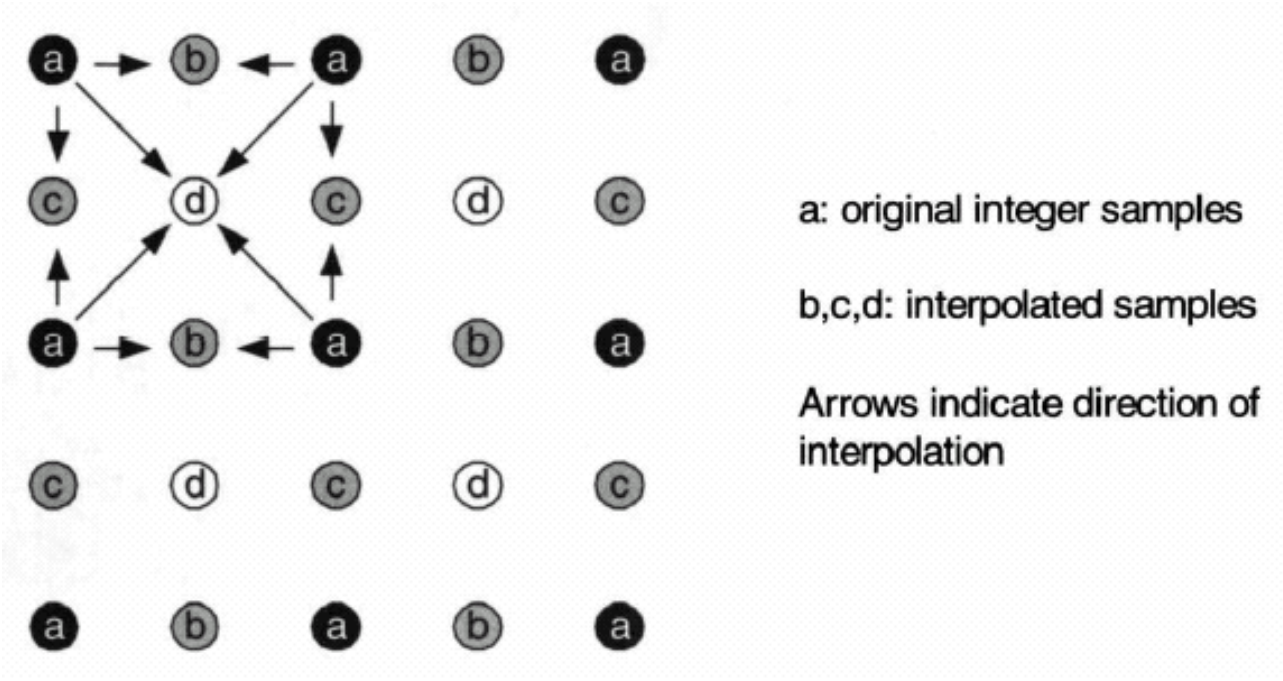
- Logarithmic search, cross-search e one-at-a-time hanno bassa complessità computazionale, al costo di spese una matching performance relativamente bassa.

- Hierarchical search è un buon compromesso tra performance e complessità ed è adatto per implementazioni hardware.

- Nearest-neighbours search, con la sua tendenza strutturale verso la predizione basata su mediana dei vettori di moto sembra funzionare bene quasi come una full search, ma con complessità molto più ridotta.
  - La buona performance è dovuta al 'bias' della mediana, che tende a produrre piccole differenze nei vettori di moto, e quindi una loro codifica efficiente.

Sub pixel motion estimation

- Per molti blocchi si ottiene un miglior match cercando in una regione interpolata, con accuratezza sub-pixel. Il generico algoritmo di ricerca viene esteso come segue:
    1. Si interpolano i campioni nella search area dell'immagine di riferimento per creare una regione interpolata a più alta risoluzione.
    2. Si effettua la ricerca in locazioni full-pixel e sub-pixel nella regione interpolata e si cerca il best match.
    3. Si sottraggono i campioni della regione su cui si ha il best match (full- o sub-pixel) dai campioni del blocco corrente per formare il blocco differenza (error block).

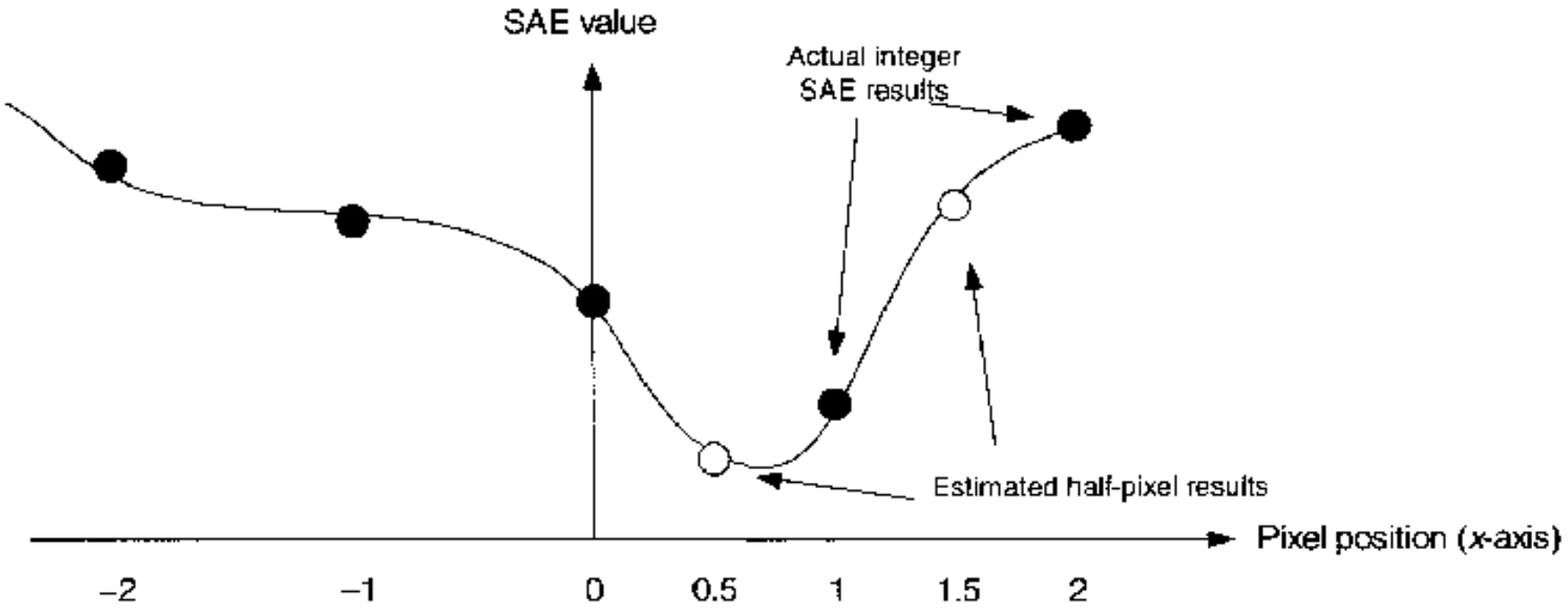


Half pixel interpolation

- Motion compensation con accuratezza half-pixel è supportata da H. 263, MPEG-1 e MPEG-2 standard

- Livelli di interpolazione più elevati (1/4 pixel o più) sono proposti per gli standard emergenti H.26L/H.264. ¼ pixel è usato in MPEG-4

- Aumentare la 'profondità' di interpolazione porta ad avere una migliore block matching performance al costo di un aumento di complessità computazionale.
  - Per limitare l'incremento di complessità computazionale di solito si cerca il best match su posizioni intere e poi si raffina con ricerca sub-pixel attorno alla posizione iniziale, oppure stimo a partire da valori basati su full-pixel


SAE value
Actual integer
SAE results
Estimated half-pixel results
Pixel position (x-axis)
−2
−1
0
0.5
1
1.5
2

Esercizi

- Usare VCDemo per provare il cambiamento di velocità nella ME con hierarchical matching quando si cambiano i livelli
  - Aprire sequenza video YUV
  - Aprire finestra ME
  - Impostare livelli diversi
  - Esaminare anche l'immagine errore
- Sempre con VCDemo, mantenendo lo stesso livello, cambiare dimensioni blocco di match e range di ricerca: esaminare i residui

Ottimizzazioni

- Early termination
  - Ogni volta che viene calcolato il best match, nel ciclo di calcolo della misura (es. SAE) si controlla se si è passato il minimo ottenuto fino a quell'istante:

  - Es.:
```
if (SAE_attuale > SAE_minimo)
                    break;
```

- Per SAD ed SSE valgono le equazioni:

$$\sum_{\text{block}} |S_k - S_{k-1}| \geq \left| \sum_{\text{block}} S_k - S_{k-1} \right| = \left| \sum_{\text{block}} S_k - \sum_{\text{block}} S_{k-1} \right|$$

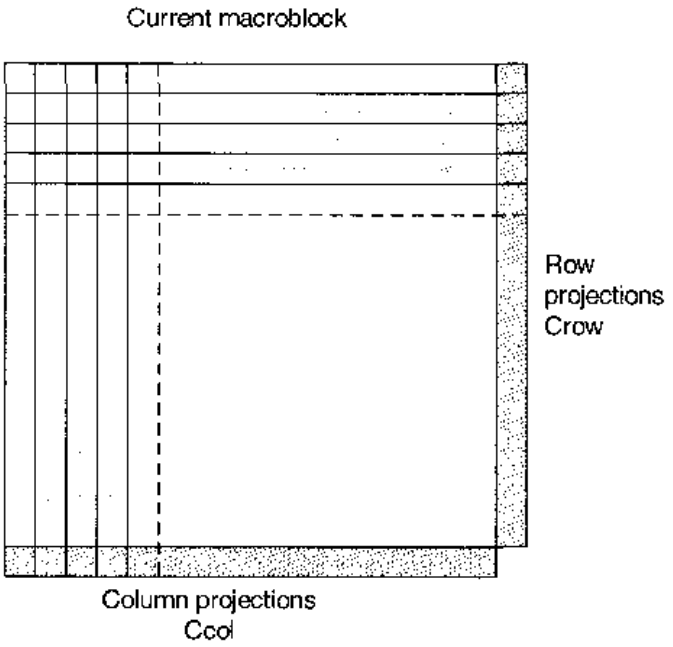
$$\sum_{\text{block}} |S_k - S_{k-1}|^2 \geq \frac{1}{N} \left| \sum_{\text{block}} S_k - S_{k-1} \right|^2 = \frac{1}{N} \left| \sum_{\text{block}} S_k - \sum_{\text{block}} S_{k-1} \right|^2$$

number of terms in sum

- La strategia è:
  1. Calcolare le somme parziali per blocco corrente e riferimento
  2. Comparare i blocchi usando le somme parziali
  3. Non calcolare la comparazione standard se le somme parziali mostrano che l'errore è già maggiroe del miglior risultato precedente

- Row / column projections: si usano i valori delle proiezioni per approssimare il calcolo di SAE
  - Una proiezione è calcolata sommando I valori di luminanza sulle colonne e le righe

$$\mathrm{SAE}_{\mathrm{approx}} = \sum_{i=0}^{N-1} Ccol_i - Rcol_i + \sum_{j=0}^{N-1} Crow_j - Rrow_j$$



Gestione bitrate: VBR vs. CBR

- Ci sono due modi per gestire il bitrate:
  - Variable Bit Rate (VBR)
    - Il bitrate può variare
  - Constant Bit Rate (CBR)
    - Il bitrate è costante all'interno di una qualche finestra temporale.

- Nel sequence header è specificato se CBR o VBR.
  - Nel sequence header anche informazioni su come calcolare il buffer minimo per decomprimere i frame.

VBR Q-scale

- In generale: VBR usato per mantenere la qualità del video.
- Il Q scale è aggiornato per mantenere la massima compressione sulla base di una qualità minima richiesta.

- Dobbiamo specificare una metrica per definire la qualità. Soluzione comune: Q scale impostato staticamente per I, P, e B frame.
  - Si varia all'interno dei macroblocchi

CBR Q-scale

- Per mantenere il CBR si usa Q scale per controllare il bitrate.
  - Maggiore è il valore di Q scale maggiore è la compressione (qualità peggiore).
  - Con bassi Q scale si privilegia la qualità a spese della compressione.

- Soluzione comune: si imposta una dimensione obiettivo per I, P, e B frame; quindi si aggiusta il Q scale dei macroblocchi man mano che si codificano, per raggiungere il target.

Decompressione

- È veloce: non richiede né ricerca nè matching. Opera come segue:

1. Se i Motion Vector sono presenti
   - Usa le tavole standard per decodifca Huffman dei vettori di moto
   - Ricrea da codifica differenziale I vettori di moto
   - Legge i blocchi di riferimento dal buffer

2. Se il Quantizer è presente
   - scala la matrice di quantizzazione con q-scale

3. Se Block Code è presente
   - Usa la tavole dello standard per la decodifica Huffman dei coefficienti
   - Decomprime RLE
   - Effettua Zigzag al contrario
   - Effettua la quantizzazione al contrario
   - Effettua la DCT inversa

4. Combina blocchi differenza e riferimento

5. Combina 6 blocchi in un macroblocco (con un-subsampling)

6. Combina macroblocchi in un’immagine

7. Converte YCbCr in RGB per fare il display sullo schermo

Part II - MPEG 2

- MPEG2 è stato definito per ottenere video in qualità broadcast a 4-9 Mbps, in alternativa a MPEG1 che mirava a qualità VHS a 1.5 Mbps. MPEG2 arriva a supportare HDTV e bitrate fino a 60 Mbps

- Similmente a MPEG1 la definizione del bitstream definisce implicitamente gli algoritmi di decompressione. Gli algoritmi di compressione sono invece non definiti e lasciati agli implementatori

- Per raggiungere qualità broadcast è stato aggiunto il supporto dei field

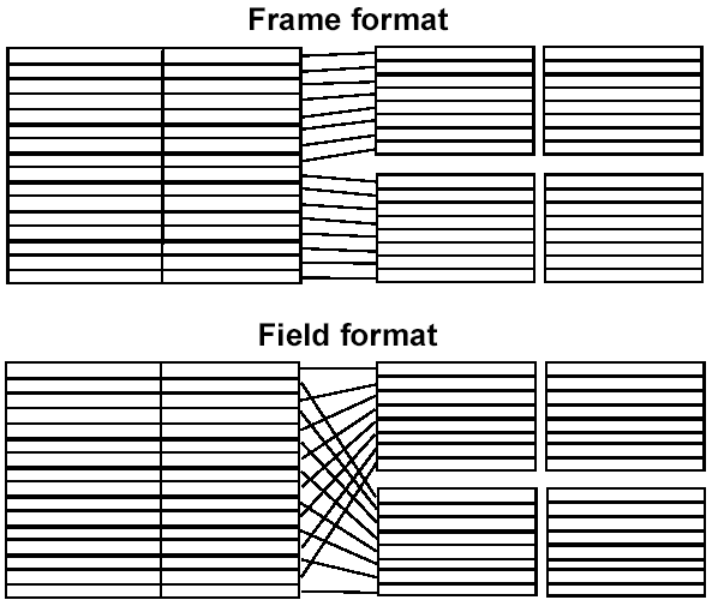
- MPEG2 introduce in aggiunta alla frame-based prediction di MPEG1:
  - Field-based prediction
  - Field-based DCT

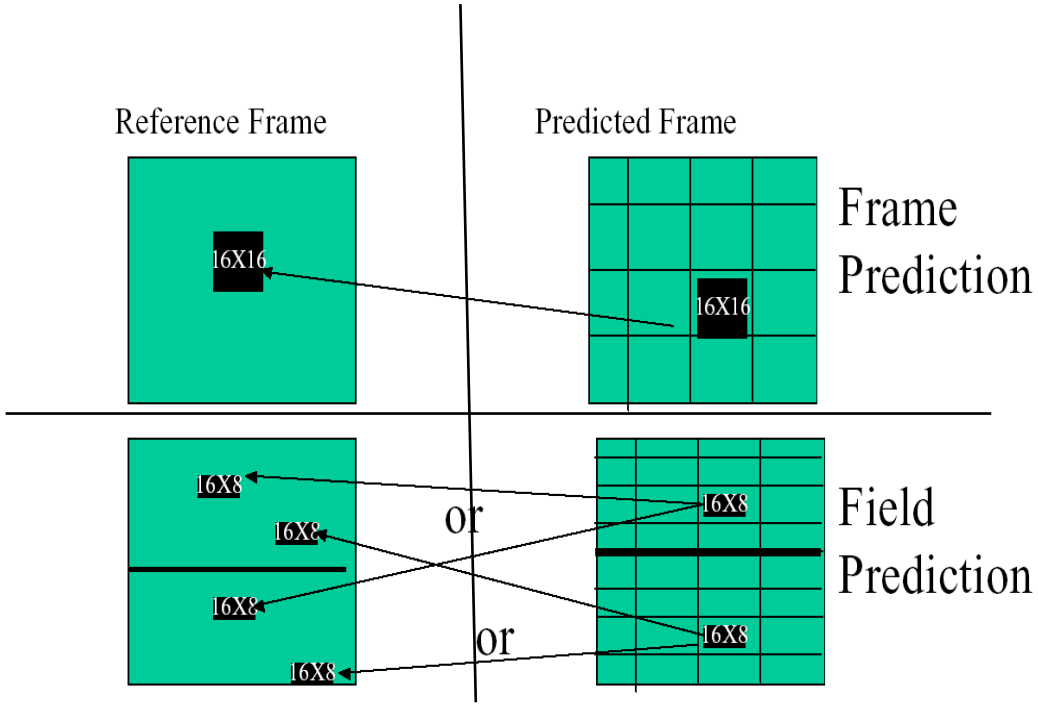
  In fase di encoding posso scegliere se produrre frame o field

- MPEG2 è stato disegnato per essere un superset di MPEG1, e per essere compatibile all'indietro. Non è però necessario implementare tutte le nuove funzionalità. Gestisce:
  - Progressive e interlaced video
  - 4:2:0, 4:2:2, 4:4:4 color sampling
  - Profili e livelli diversi
  - Scalabilità
  - Dimensioni delle immagini fino a 16K x 16K;
  - PAL e NTSC
  - Frame rate: 23.98, 24, 25, 29.97, 30, 50, 59.94, 60

Frame picture: frame /field - based prediction

- Se in fase di encoding si sceglie di produrre frame è possibile impiegare frame-based o field-based prediction. Sono consentite frame-based e field-based DCT
- La frame-based prediction usa un solo vettore di movimento (forward o backward) per descrivere il movimento rispetto al frame di riferimento
- La field-based prediction usa due vettori di movimento: uno per ogni field
  - Nel caso di B frame usa fino a 4 vettori: 2 per field per forward e backward
  - Motion compensation è fatto per blocchi di 16x8 pixel
  - I vettori di moto sono calcolati su base half-pixel: più precisi rispetto a MPEG1 e con miglior compressione

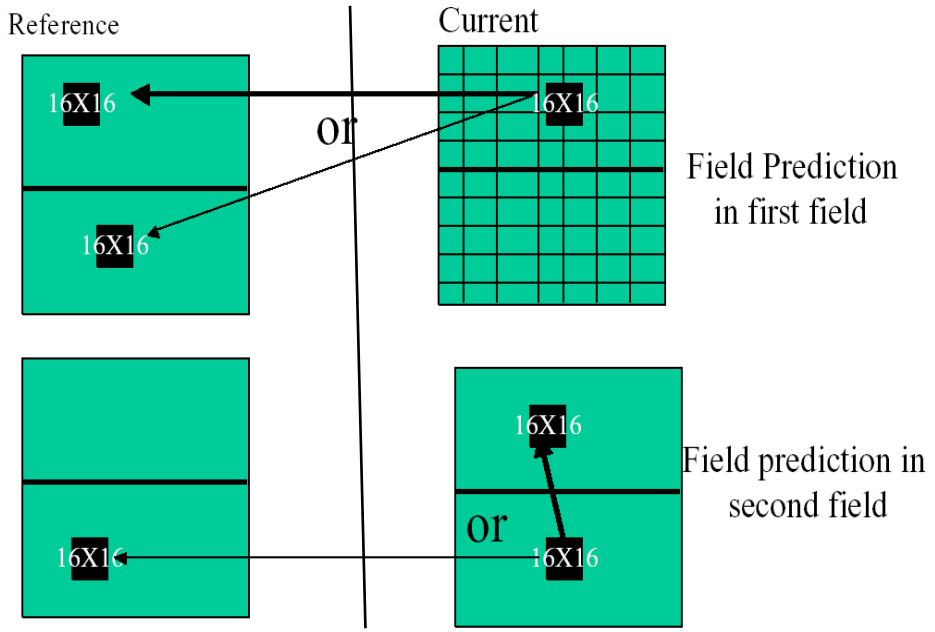

Frame format
Field format


Reference Frame
Predicted Frame
16X16
16X16
Frame
Prediction
16X8
16X8
16X8
16X8
or
or
16X8
16X8
Field
Prediction

Field picture: field – based prediction

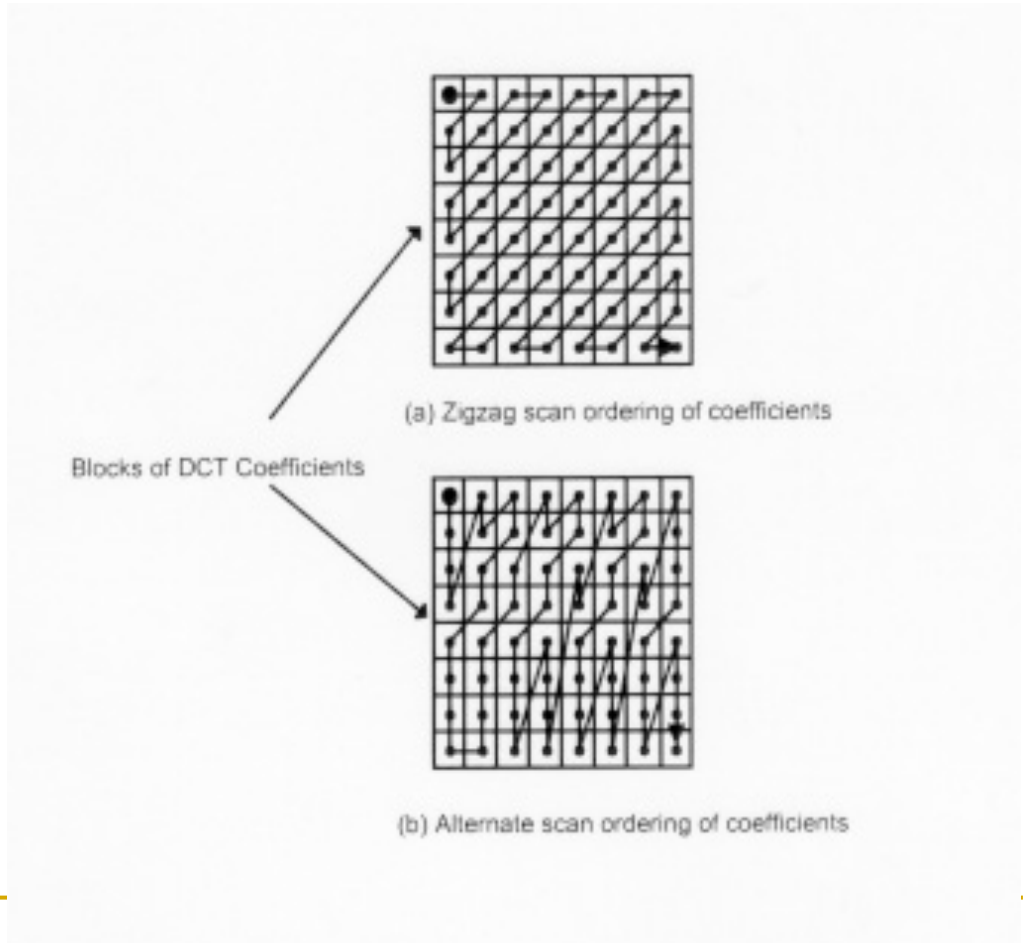
- Se in fase di encoding si sceglie di produrre field è possibile impiegare solo field-based prediction.

- La DCT è Field-based: opera su linee alternate, su blocchi di 8x8 ottenuti raggruppando linee dello stesso field



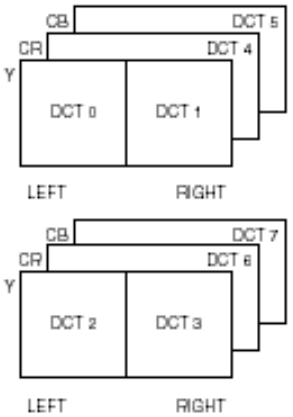
Scansione Zig-Zag

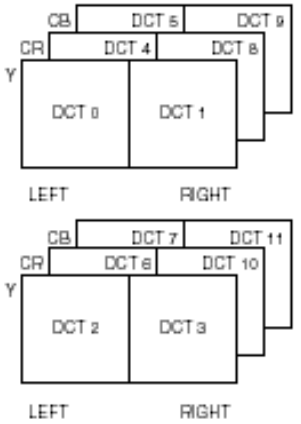
- Oltre alla scansione zig-zag classica del JPEG e dell'MPEG1 in MPEG2 c'e' una scansione diversa adatta a funzionare per frame interlacciati



Campionamento colore

- Sono stati aggiunti il 4:2:2 e 4:4:4
  - Consentono qualità professionale
  - Cambiano i macroblocchi
- Quando si usano i campionamenti 4:2:2 e 4:4:4 si possono usare matrici di quantizzazione diverse per Y e CbCr
- In generale si può cambiare matrice di quantizzazione nei picture layer
  - In MPEG-1 è possibile solo nel program layer
  - Sono definiti nuovi VLC per i coefficienti DCT





Q scale

- In MPEG2 è consentito un Qscale non lineare, in aggiunta a quello ammesso in MPEG 1

.5	1.0	1.5	2.0	2.5
2.5	3.0	3.5	4.0	5.0
6.0	7.0	8.0	9.0	10.0
11.0	12.0	14.0	16.0	18.0
20.0	24.0	26.0	28.0	32.0
36.0	40.0	44.0	48.0	52.0
56.0				

Profiles e Levels

- In MPEG2 profiles e levels definiscono le capacità minime richieste ad un decoder
  - Profiles: specificano la sintassi, es. algoritmi
  - Levels: specificano i parametri, es. risoluzione, frame rate, etc.

  - Si indica profile@level

Profiles

- Simple Profile (4:2:0)
  - Adatto per videoconferenza
  - Corrisponde al Main profile senza B frame
- Main profile (4:2:0)
  - Tipico per videoprofessionale SDTV (bitrate a 50 Mbps)
  - Di applicazione generale, è il profilo più importante
- Multiview profile
  - Adatto per riprese fatte con doppie telecamere che riprendono la stessa scena
- 4:2:2 profile
  - Adatto a video professionale SDTV, e per HDTV ((bitrate a 50 Mbps)
- SNR e Spatial Scalable profile (4:2:0)
  - Aggiungono il supporto per scalabilità SNR e/o spaziale: gestiscono diversi gradi di qualità
- High 4:2:0 profile
  - Adatto a HDTV

Levels

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

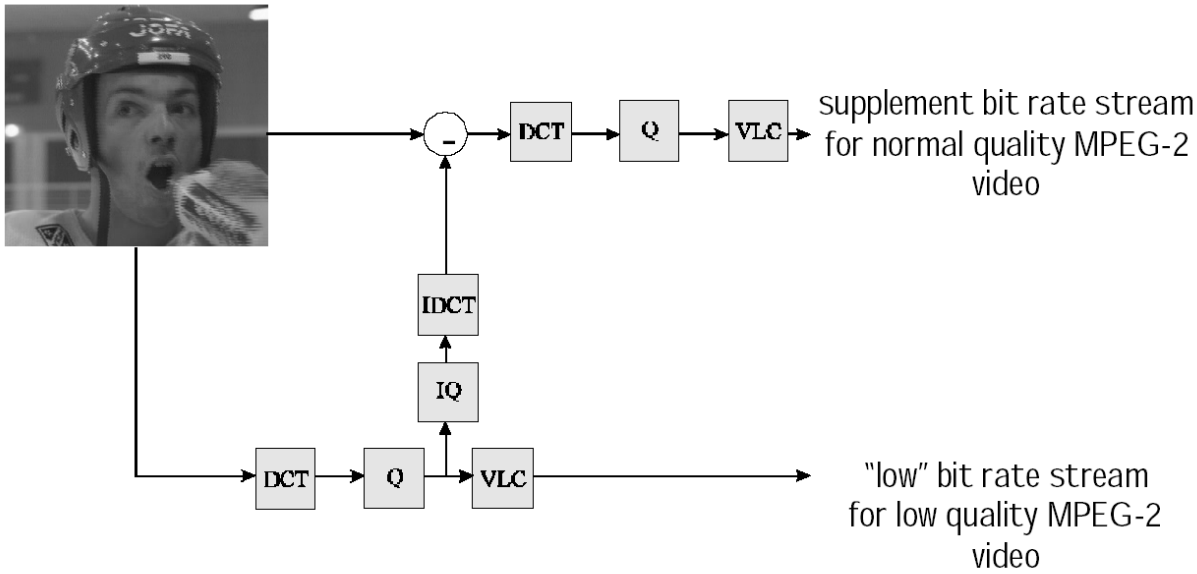
Profili e livelli

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		

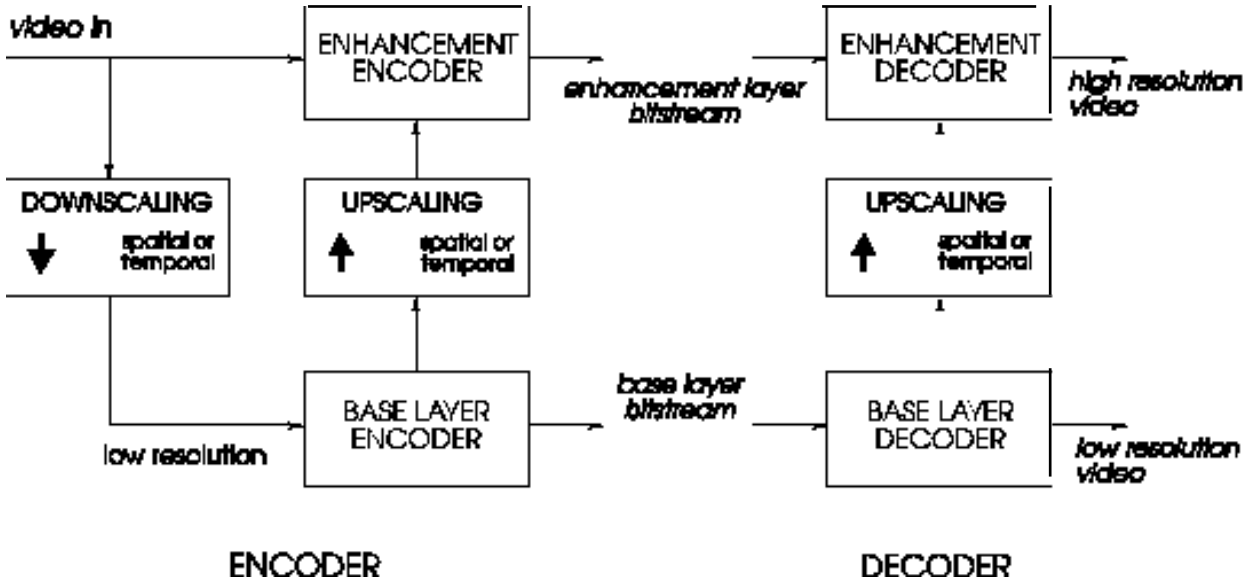
Scalabilità

- SNR, Spatial e High profile supportano 4 modi di operazione scalabili. I modi dividono i video MPEG2 in layer per gestire le priorità dei dati video.
- I modi di scalabilità forniscono interoperabilità tra sistemi diversi, es. uno stream per HDTV visibile anche su SDTV
- Un sistema che non vuole ricostruire il video a risoluzione spaziale o temporale più alta ignora il raffinamento dei dati e si limita a prendere la versione base

- SNR scalability (2 layer)
  - Per applicazioni che richiedono più livelli di qualità
  - Tutti i layer hanno la stessa risoluzione spaziale. Il layer di base fornisce la qualità, quello di enhancement la migliora fornendo dati più raffinati per i coefficienti DCT del layer di base
  - Consente il “graceful degradation”



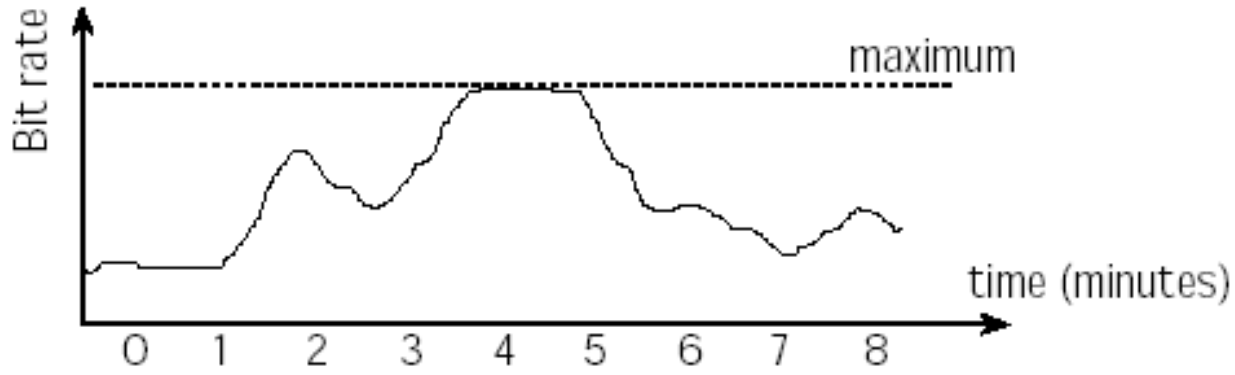
- Spatial scalability (2 layer)
  - Il layer di base fornisce la risoluzione spaziale e temporale di base
  - Il layer di enhancement usa il layer di base interpolato spazialmente per aumentare la risoluzione spaziale
  - Uso l'upscaling per predire la codifica della versione ad alta risoluzione. L'errore di predizione è codificato nel layer di enhancement



- Temporal scalability
  - Simile alla spatial scalability, solo che qui si raffina nel tempo
- Data partitioning
  - Fornisce resistenza agli errori di trasmissione
  - I coefficienti a bassa frequenza della DCT e altri dati vengono mandati su un canale ad alta priorità, gli altri coeff. DCT su un canale a priorità più bassa (es. su ATM)

CBR vs. VBR

- Anche in MPEG-2 si può scegliere tra
  - CBR: es. per digital broadcast
  - VBR: es. per DVD, la qualità degrada solo se passiamo il massimo consentito



Part IV - MPEG 4

Applicazioni di MPEG4

- Mpeg-4 è uno standard progettato per quattro diversi contesti applicativi:
  - real-time communication (videoconferenza);
  - televisione digitale;
  - applicazioni grafiche interattive (DVD, ITV);
  - World Wide Web.

- Fornisce un insieme di soluzioni per soddisfare le necessità di:
  - autori;
  - service providers;
  - utenti finali.

- Per tutte le parti coinvolte, MPEG vuole evitare la proliferazione di una moltitudine di formati proprietari e formati non di rete.

Caratteristiche

- Per *autori*, MPEG-4 permette la produzione di contenuto che ha maggiore riusabilità e flessibilità di quella oggi possibile con singole tecnologie come televisione digitale, grafica animata, pagine dal World Wide Web e loro estensioni. Rende anche possibile una migliore gestione e protezione dei diritti di autore sul contenuto.

- Per *fornitori di servizi di rete,* MPEG-4 fornisce informazione trasparente che può essere interpretata e tradotta negli appropriati segnali nativi di ciascuna rete. Comunque, esclude considerazioni su Quality of Service, per cui MPEG-4 fornisce un generico insieme di parametri QoS per differenti media MPEG-4. Il mapping esatto per queste traduzioni è al di fuori dello scopo di MPEG-4 ed è lasciato alla definizione dei network provider.

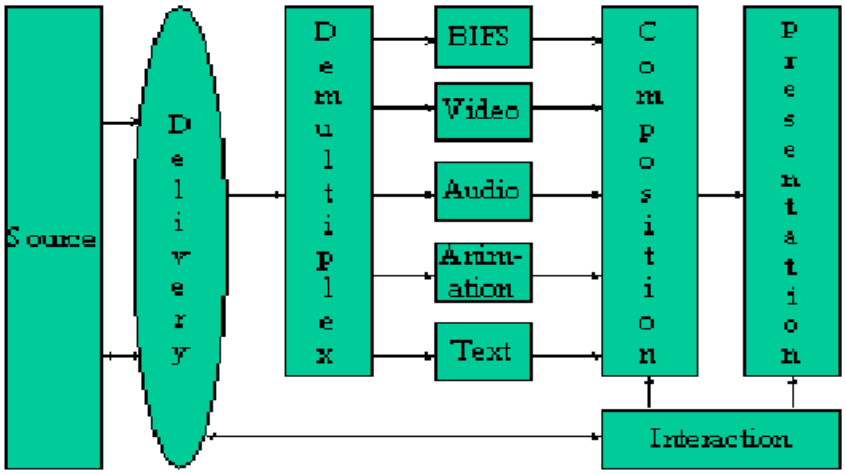
Caratteristiche

- Per *fornitori di servizi di rete,* MPEG-4 fornisce informazione trasparente che può essere interpretata e tradotta negli appropriati segnali nativi di ciascuna rete. Comunque, esclude considerazioni su Quality of Service, per cui MPEG-4 fornisce un generico insieme di parametri QoS per differenti media MPEG-4. Il mapping esatto per queste traduzioni è al di fuori dello scopo di MPEG-4 ed è lasciato alla definizione dei network provider.

- Per *utenti finali*, MPEG-4 rende disponibili molte funzionalità a cui potenzialmente si potrebbe avere accesso da un singolo terminale e più alti livelli di interazione con il contenuto secondo i limiti imposti dall'autore. Tra le applicazioni, comunicazione real-time, sorveglianza e multimedia mobile.

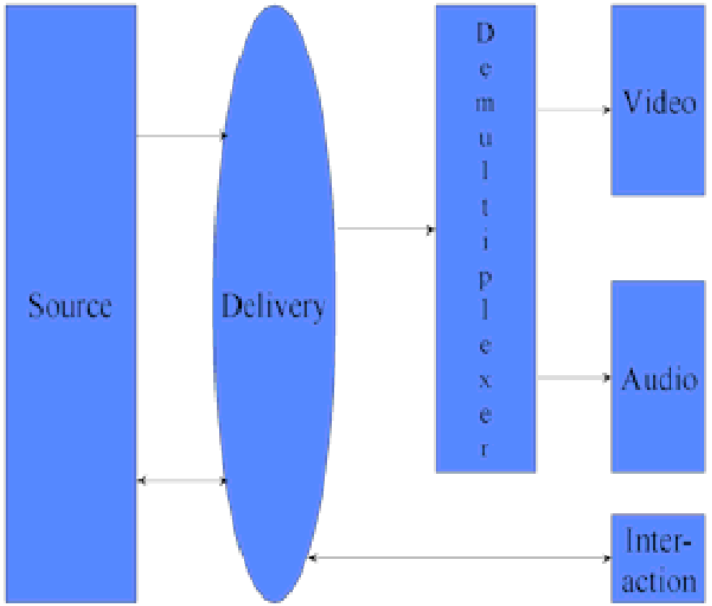
- Scopi di MPEG4
  - Indipendenza delle applicazioni dai dettagli di basso livello
  - Usabilità su un range di bitrate che vada da pochi kbit/s fino a qualche Mbit/s
  - Riusabilità di tool di codifica e dei dati: l'informazione è rappresentata a livello di singole componenti chiamate "Audio-Visual objects (AVO)"
  - Identificazione e gestione dell' IPR dei contenuti (Intellectual Property Right)
  - Interattività non solo a livello di video (video A vs. video B) ma tra diversi AVO, stile pagina Web
  - Possibilità di avere hyperlink per far interagire più sorgenti di informazione (sempre a livello di AVO), stile pagina web
  - Capacità di gestire contemporaneamente informazione naturale/ sintetica e real-time/non-real-time
  - Capacità di comporre e rappresentare informazione secondo l'interazione con l'utente (paradigma VRML e computer grafica in generale)

Da MPEG2 a MPEG4



MPEG2

MPEG4



- MPEG2 ha anche una funzione per il supporto di interazioni client server (che MPEG1 non ha)

- MPEG4 estende il modello di MPEG2 aggiungendo la capacità di modificare la visualizzazione

Lo standard MPEG4

- Lo standard MPEG4 comprende le tecnologie per supportare:
  1. rappresentazione (codificata) per unità di contenuto aurale, visuale e audiovisuale (audio-visual objects" o AVO);
  2. il modo in cui gli AVO sono composti in una scena;
  3. il multiplexing e la sincronizzazione di AVO, per trasportarli su canali con QoS adeguato alla natura degli AVO (o QoS adeguato all'utente);
  4. Un'interfaccia generica tra applicazione e trasporto
  5. Il modo in cui un utente interagisce con la scena (es. punto di vista) e gli oggetti che la compongono (es. click su di un oggetto)
  6. Proiezione delle scene AV composte secondo il punto di vista/ ascolto desiderato

- Le scene audiovisuali sono composte da diversi AVO, organizzati gerarchicamente, es.:
  - Sfondo fisso 2D
  - Immagine di una persona che parla (senza lo sfondo)
  - Voce associata alla persona
  - Oggetto sintetico (tavolo e mappamondo)
  - Suono sintetico (es. musica della sigla)
  - Etc.

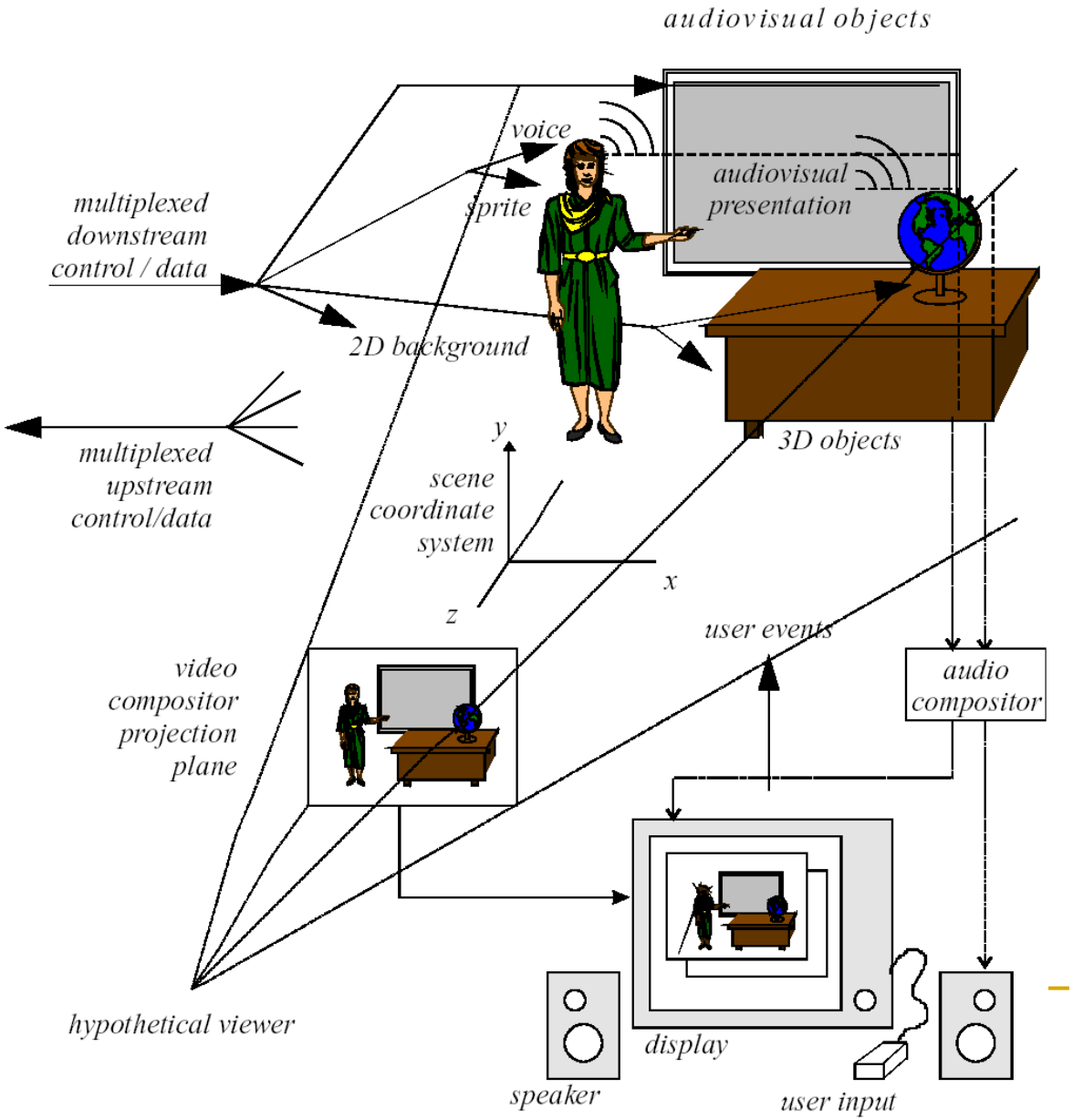
- MPEG standardizza un insieme di primitive AVO per rappresentare oggetti naturali ed artificiali, 2D e 3D

- La descrizione di una scena consiste delle seguenti informazioni:
  - Come gli oggetti sono raggruppati insieme (struttura gerarchica)
  - Come gli oggetti sono posti nello spazio e nel tempo: ogni oggetto ha coordinate locali che vengono rimappate sulle coordinate della scena, e l'apparizione degli AVO deve essere sincronizzata
  - Attributi degli AVO: alcuni elementi possono essere modificabili (es. contrasto o colore)

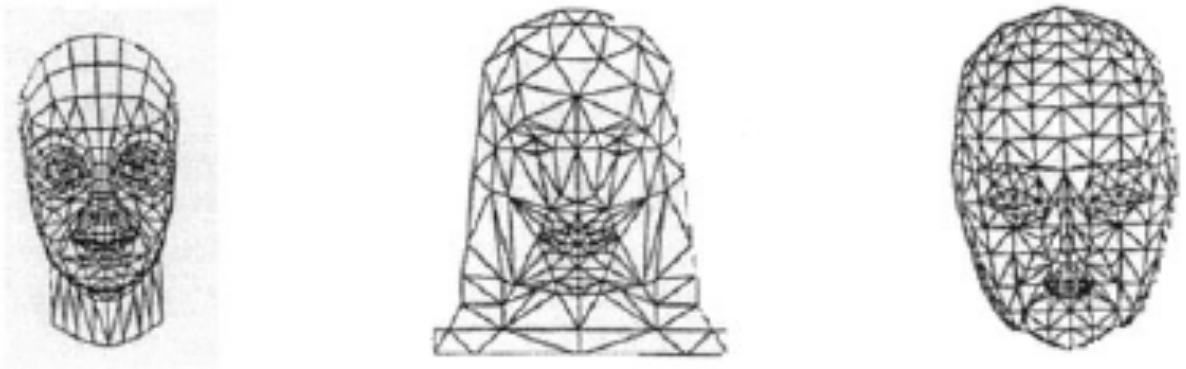
- BIFS: BInary Format for Scene description
  - Linguaggio binario (derivato da VRML) per la descrizione delle scene
  - La descrizione della scena è codificata in modo indipendente dagli stream dei media primitivi

- La scelta dei parametri di un oggetto associabili ad una scena è precisa:
  - Se il parametro può essere usato per la codifica dell'oggetto (es. parametri di moto) allora NON viene descritto nello stream BIFS

Esempio di scena MPEG4



- Oltre alle primitive viste sopra (presenti nella figura) esistono AVO per:
  - Testo e grafica
  - talking heads e testo associato: usato dal ricevente per sintetizzare il parlato ed il movimento della testa
  - Animazione del corpo umano
  - Audio sintetico (es. TTS)



Rappresentazione codificata dei media objects

- Nella loro forma codificata, questi oggetti sono rappresentati nel modo più efficiente possibile.
  - La codifica di questi oggetti è il più efficiente possibile, anche tenendo conto che si supportano diverse funzionalità come la robustezza al rumore, estrazione ed editing di un oggetto o l'avere un oggetto disponibile in forma scalabile.

Audio

- Sono definiti diversi algoritmi di compressione audio, secondo il bitrate che si vuole ottenere
  - È definito anche un framework per rendere scalabile la compressione audio
  - È importante notare che nella loro forma codificata, gli oggetti (audio o video) possono essere rappresentati indipendentemente da quelli circostanti e dalla sfondo.

AVO

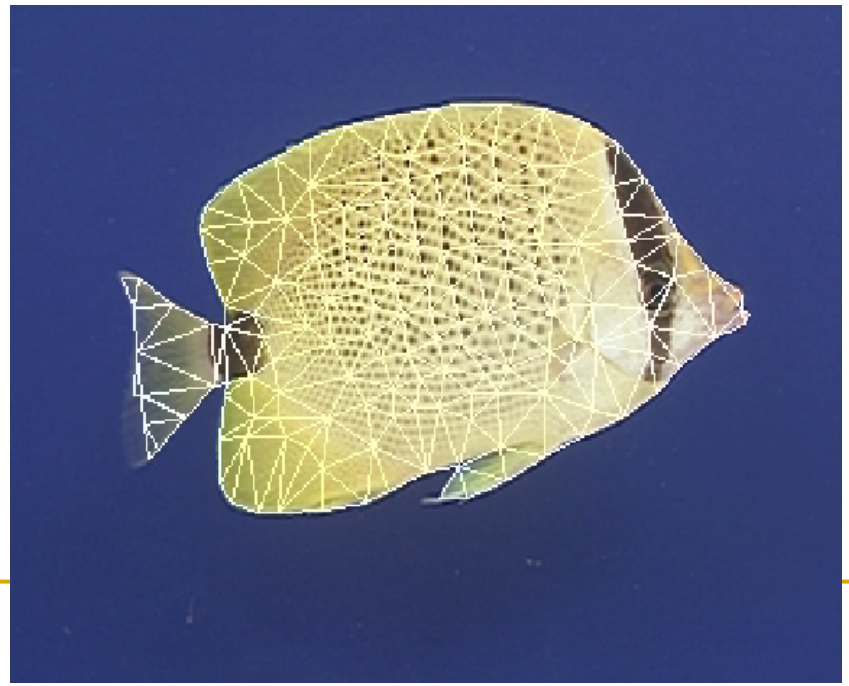
- Gli AVO possono essere naturali o artificiali
  - AVO naturali: texture, immagini e video
    - Set di tool e algoritmi per:
      - Compressione immagini e video
      - Compressione di texture per texture mapping su mesh 2D e 3D
      - Compressione di mesh 2D
      - Compressione di geometrie variabili nel tempo per l'animazione di mesh
      - Accesso diretto ai VO
      - Codifica content-based di immagini e video
      - Scalabilità content-based di texture, immagini e video
      - Scalabilità spaziale, temporale e di qualità

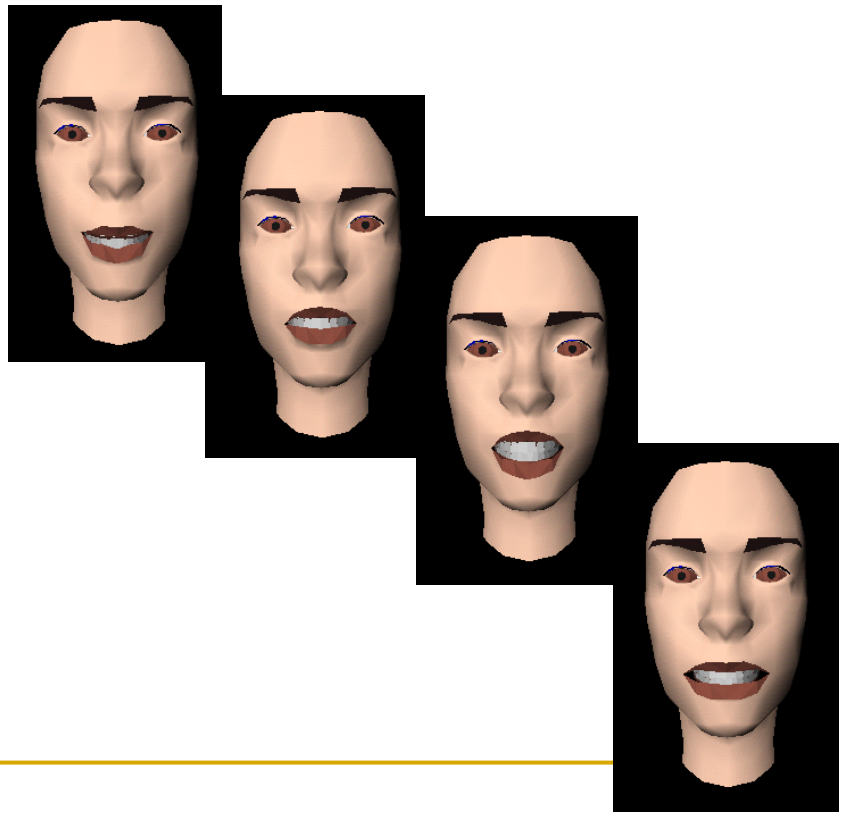
- MPEG4 consente di codificare figure di forme diverse (non solo rettangolari)
  - Il video diventa una composizione di oggetti 2D
    - Possono essere disposte in uno spazio 3D

Audio Visual Objects (AVOs)

AVOs sintetici includono:

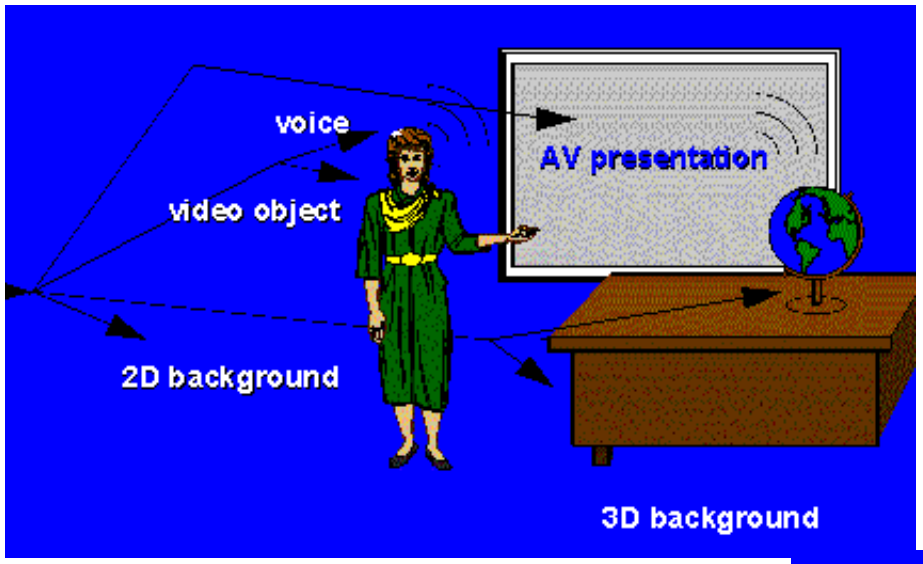
- volti animati;
- corpi animati;
- mesh 2D animate.

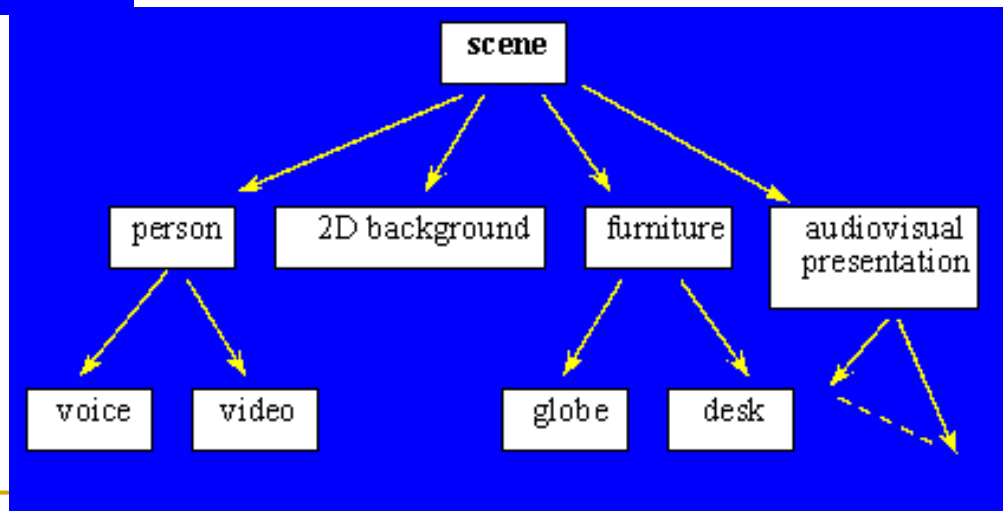




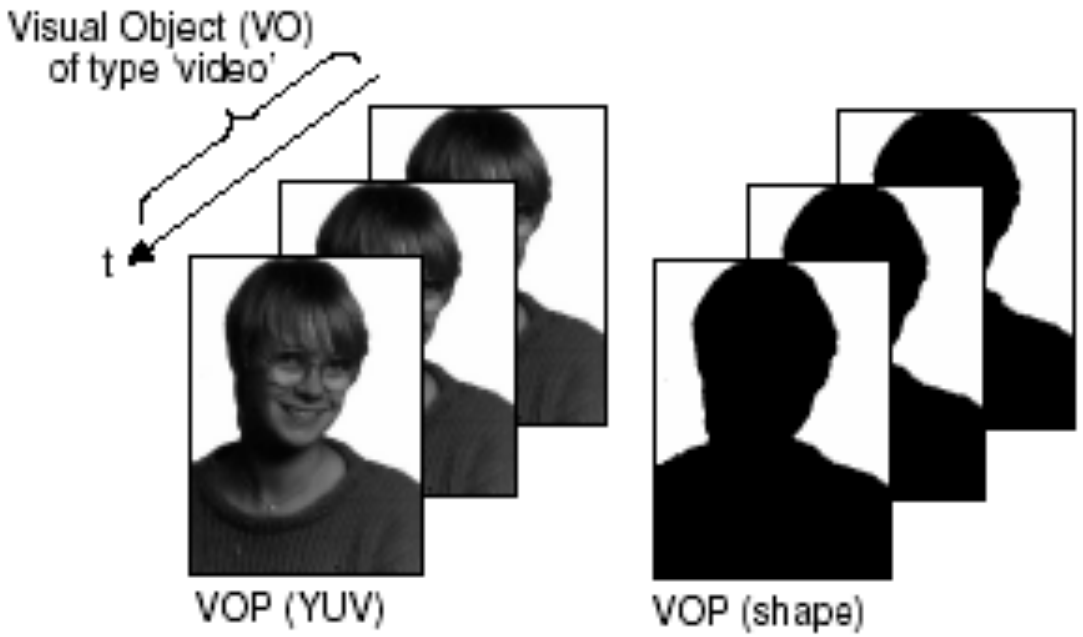
Composizione della scena

- La composizione della scena consente di:
  - eliminare, spostare e posizionare a proprio piacere gli AVOs che compongono la scena stessa;
  - raggruppare diversi AVOs in modo da formare AVOs composti, così da consentire al fruitore di manipolare insiemi consistenti di oggetti;
  - fornire dati di animazione agli AVOs in modo da personalizzarne e modificarne gli attributi (applicare una particolare tessitura ad un oggetto, mandare parametri di animazione ad un volto ecc.);
  - cambiare interattivamente il punto di vista nella scena.


voice
video object
AV presentation
2D background
3D background


scene
person
2D background
furniture
audiovisual presentation
voice
video
globe
desk

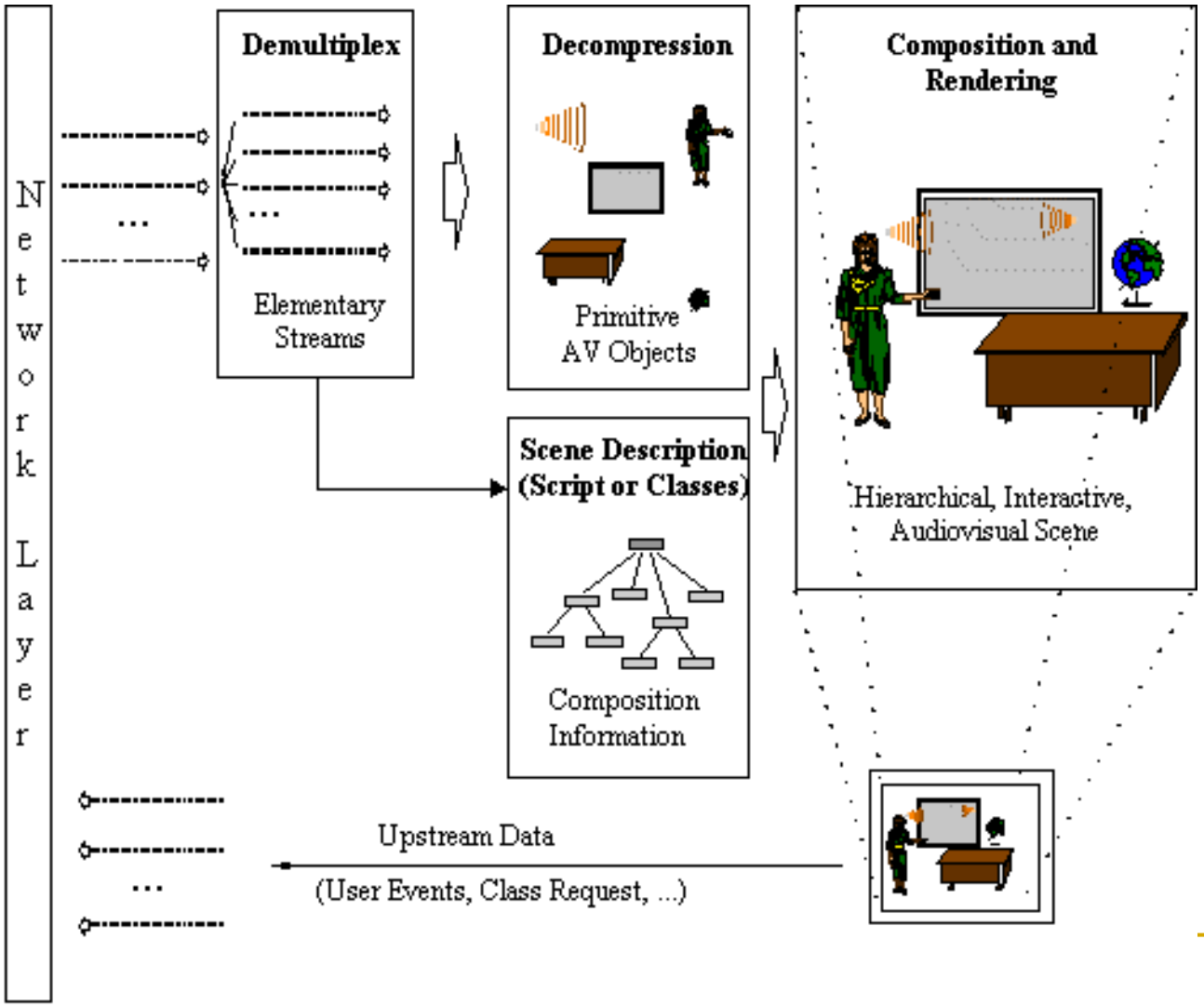
- Esempio di VO video, composto da Video Object Planes



- Il supporto di queste funzionalità è ottenuto attraverso una composizione e definizione della scena che adotta molte delle soluzioni ed impostazioni definite dallo standard VRML.
- Sono comunque eliminati molti vincoli che rendono troppo rigido lo standard VRML.
  In particolare Mpeg-4 si propone (in più rispetto al VRML) di:
  - codificare informazione audio-video ad elevata qualità;
  - fare riferimento ad una scena dinamica che può essere cambiata in ogni momento.

- È previsto un livello di interazione con l'utente finale che consente a quest'ultimo di:
  - navigare nella scena;
  - eliminare o spostare gli oggetti all'interno della scena;
  - avviare l'esecuzione di alcune azioni (ad esempio selezionare un oggetto e provocare il play di un filmato);
  - selezionare uno tra più linguaggi nel caso siano presenti diverse tracce audio.

La decodifica di MPEG4



Componenti principali

- Stream provenienti dalla rete (o da un dispositivo di memoria) sono demultiplexati per ottenere stream elementari.

- Gli stream elementari (ESs) sono esaminati e passati agli appropriati decodificatori.
  - La decodifica recupera il dato in un AVO dalla sua forma codificata ed esegue le operazioni necessarie per ricostruire l'AVO originale pronto per il rendering sul dispositivo appropriato.

- L’AVO ricostruito è reso disponibile al layer di composizione per il suo uso potenziale durante il rendering della scena.

- Gli AVOs decodificati, insieme con informazione di descrizione della scena, sono usati per comporre la scena come descritto dall’autore.

- L’utente finale può interagire con la scena, secondo quanto stabilito dall’autore, che è eventualmente renderizzata e presentata.

Codifica di oggetti visuali naturali

- Mira a fornire una tecnologia standardizzata che consenta efficiente memorizzazione, trasmissione e manipolazione di tessiture, immagini e video per ambienti multimediali.
  - Tools per la decodifica e rappresentazione di unità atomiche di contenuto immagine e video detti "*video objects*" (VOs). Un esempio di VO potrebbe essere una persona che parla (senza background) che può poi essere composta con altri AVOs per creare una scena. Immagine rettangolari convenzionali sono trattate come un caso speciale di tali oggetti.

Parte visual di MPEG-4

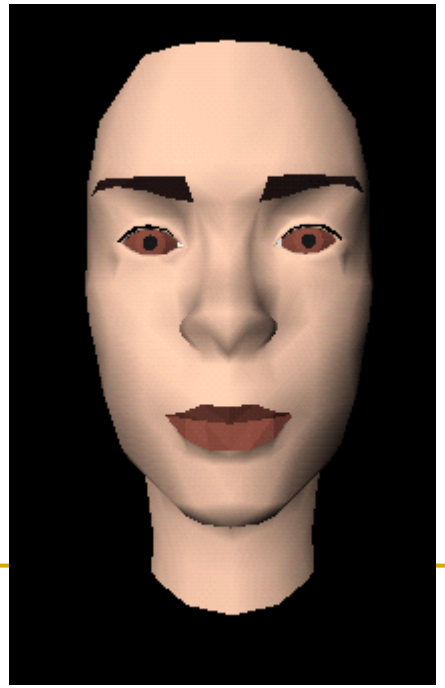
- Fornisce soluzioni nella forma di tools e algoritmi per:
  - compressione efficiente di immagini e video;
  - compressione efficiente di tessiture per il mapping di tessiture su mesh 2D e 3D;
  - compressione efficiente di stream geometrici variabili nel tempo che animano le mesh;
  - accesso casuale efficiente a tutti i tipi di oggetti visuali;
  - funzionalità di manipolazione estese per immagini e sequenze video;
  - codifica basata sul contenuto di immagini e video;
  - scalabilità basata sul contenuto di tessiture immagini e video
  - scalabilità spaziale, temporale e di qualità;
  - robustezza all'errore.

Oggetti sintetici

- Formano un sottoinsieme della classe più ampia di oggetti ottenuti con grafica computerizzata. Consideriamo i seguenti VOs sintetici:
  - rappresentazione parametrica di
    - descrizione sintetica di faccia e corpo umano
    - stream di animazione della faccia e del corpo
  - codifica di mesh statica e dinamica con mapping delle tessiture;
  - codifica della tessitura per applicazioni dipendenti dall'osservatore.

Animazione facciale

- Il volto è un AVO in grado di poter essere visualizzato ed animato.
  - Forma, tessitura ed espressione del volto sono controllate da uno stream di dati contenenti insiemi di *Facial Definition Parameters* (FDPs) e *Facial Animation Parameters* (FAPs).



  - Appena istanziato un volto rappresenta una generica faccia priva di espressioni particolari.

- Il volto può essere animato attraverso i parametri di animazione (FAP).

- I FDP possono essere utilizzati per personalizzare l'aspetto del volto:
  - è possibile passare da una generica faccia ad una con particolari caratteristiche in termini di forma e tessitura.

- Localmente è consentito all'utente di interagire con il modello del volto:
  - ad esempio è possibile amplificare i movimenti della bocca e rendere così più facile l'interpretazione labiale del parlato.

Corpi animati

- Il corpo è un AVO in grado di poter essere visualizzato ed animato.
- Forma, tessitura e postura del corpo sono controllate da uno stream di dati contenenti insiemi di *Body Definition Parameters* (BDPs) e *Body Animation Parameters* (BAPs).
- Appena istanziato un corpo si presenta in modo eretto con le braccia allineate ai fianchi.
  - Il corpo può essere animato attraverso parametri di animazione (BAP).
  - I BDP possono essere utilizzati per personalizzare le caratteristiche del corpo (altezza, larghezza spalle etc.).

Mesh 2D animate

- Una mesh 2D è una tassellazione di una regione planare in elementi poligonali.
  - I vertici dei poligoni sono chiamati *node points* della mesh.
- Lo standard Mpeg-4 prevede l'impiego di soli elementi triangolari per la definizione delle mesh.
- Una mesh 2D può essere resa dinamica attraverso la specifica della mesh iniziale e dei vettori di moto dei node points all'interno di un intervallo temporale.

- Le mesh dinamiche sono di particolare utilità per modellare tessiture animate.

- Lo standard prevede infatti di poter applicare una tessitura ad una mesh:
  - quando i node points della mesh vengono spostati dai vettori di moto, alla tessitura di ogni elemento triangolare viene applicato un mapping parametrico che porta ad un warping della tessitura in modo da rimanere ancorata alle nuove coordinate dei node points.

Manipolazione di VOs

- Realtà aumentata:
  - fonde insieme immagini virtuali (generate da computer) con immagini reali in movimento (video) per creare informazione di visualizzazione migliorata.
- Trasformazione/animazione di oggetti sintetici:
  - sostituzione di un oggetto video naturale in un video clip con un altro oggetto video. L'oggetto video di sostituzione può essere estratto da un altro video clip naturale o può essere trasformato da una immagine singola usando l'informazione di moto dell'oggetto da rimpiazzare (richiede una rappresentazione temporalmente continua del moto).
- Interpolazione spazio-temporale:
  - modellazione del moto delle mesh fornisce una interpolazione temporale compensata dal moto più robusta.

Compressione di VOs

- La modellazione di mesh 2D può essere usata per la compressione se è scelto di trasmettere mappe di tessitura solo a key-frame selezionati e animare queste mappe di tessitura per i frames intermedi.

Indicizzazione video basata su contenuto

- La rappresentazione con mesh consente la creazione di snapshot chiave animati per la sintesi visuale del movimento di oggetti.
- La rappresentazione con mesh fornisce informazione accurata sulla traiettoria di un oggetto che può essere usata per ritrovare VOs con moto specifico.
- La rappresentazione con mesh fornisce rappresentazione di forma di un oggetto basata su vertici che è più efficiente della rappresentazione con bitmap per la ricerca di oggetti basata sulla forma.

Mesh 3D generiche

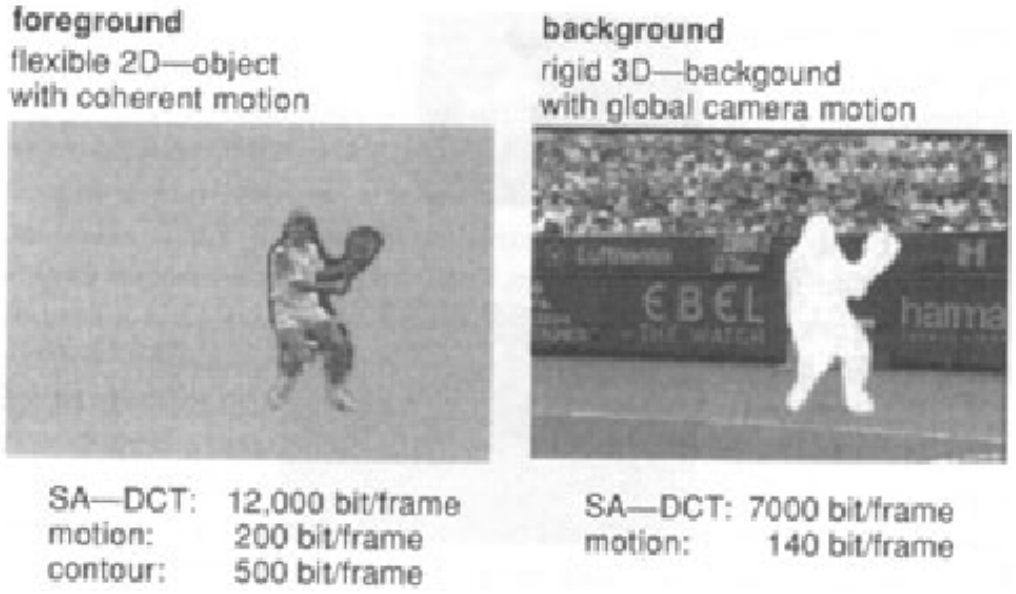
- Mpeg-4 supporta anche mesh generiche per rappresentare oggetti 3D sintetici.
  - Queste mesh supportano proprietà come colore, normali per effetti di ombreggiatura, coordinate di tessitura per il mapping di tessiture naturali, immagini e video sulle mesh.

- Sono forniti algoritmi per:
  - compressione efficiente di mesh generiche;
  - *(livello di dettaglio, LOD) scalabilità di 3D meshes* - consente al decoder di decodificare un sottoinsieme dello stream di bit totale per ricostruire una versione semplificata della mesh contenente meno vertici dell'originale. Tali rappresentazioni semplificate sono utili per ridurre il tempo di rendering di oggetti che sono distanti dall'osservatore (gestione del LOD), e consente anche una minor potenza dei motori di rendering per renderizzare l'oggetto ad una qualità ridotta.
  - *Scalabilità spaziale* - consente al decoder di decodificare un sottoinsieme dello stream totale di bit generato dal codificatore per ricostruire la mesh ad una ridotta risoluzione spaziale. Questa caratteristica è più utile quando usata combinata con la scalabilità LOD.

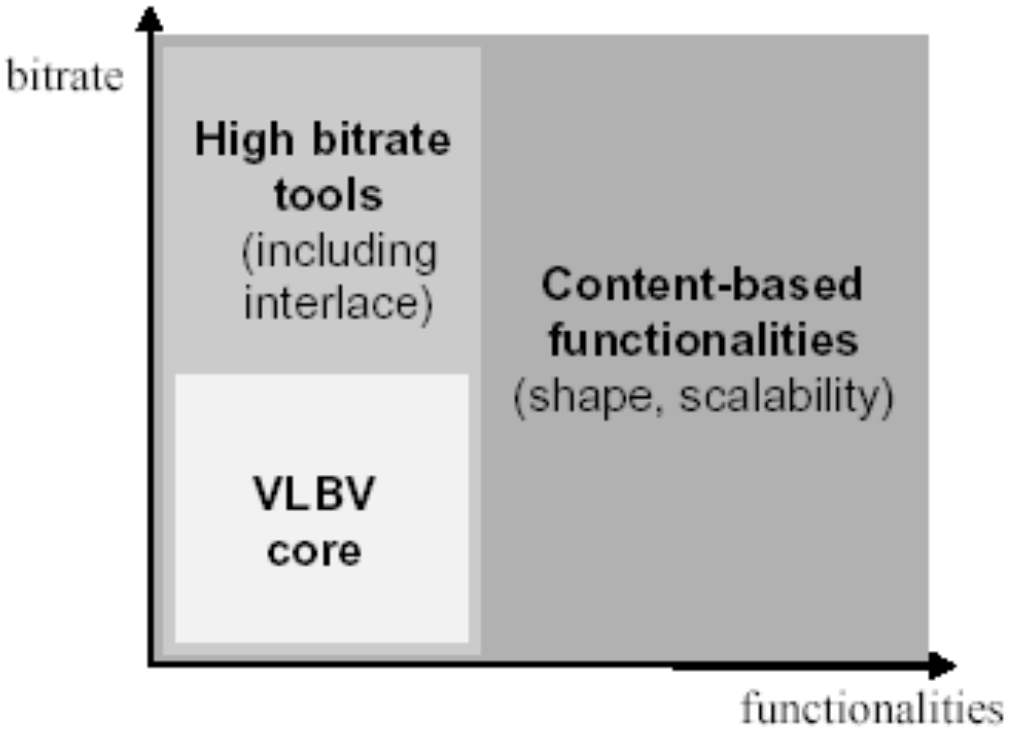
La compressione

- La base della compressione MPEG4 è quella di MPEG1 e 2:
  - Sequenze di frame rettangolari a vari:
    - Livelli di bitrate
    - Livelli di frame rate
    - Formati di input
    - Scalabilità di qualità
    - Scalabilità spaziale
    - Scalabilità temporale

- MPEG4 aggiunge la gestione di oggetti di forma qualsiasi
  - Lo scopo è raggiungere la compressione content-based



- Progressive & interlaced
- SQCIF/QCIF/CIF/4*CIF/CCIR 601, fino a 2048*2048
- Y/Cb/Cr/Alpha
- Inizialmente 4:2:0, in futuro 4:2:2 e 4:4:4 per qualità studio
- Continuous variable frame rate



La figura fornisce una classificazione di base per i bit rates e le funzionalità attualmente fornite dallo standard Visual di Mpeg-4 per immagini naturali e video.

- VLBV: Very Low Bit-rate Video: algoritmi e tool per applicazioni che operano a bitrate tra 5 e 64 kbits/s:
  - sequenze di immagini a bassa risoluzione spaziale (max. CIF)
  - basso frame rate (fino a 15 fps).

VLBV Core

- Codifica di sequenze di immagini rettangolari
- Accesso diretto, fast forward e reverse

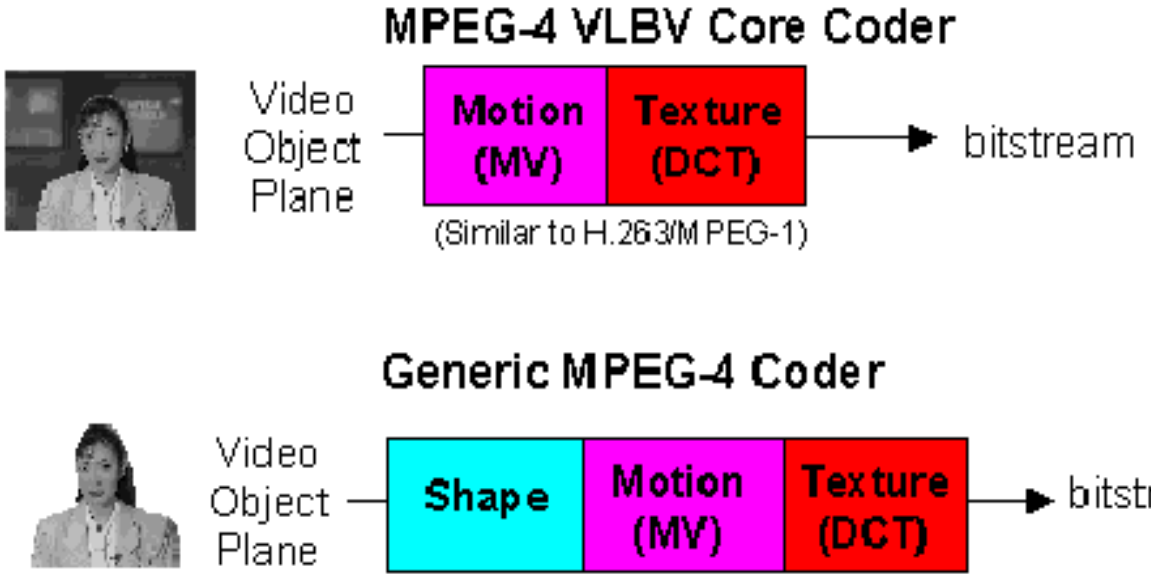
- Queste funzioni sono previste anche per bitrate più elevati, con input fino a ITU-Rec. 601

Funzioni Content-based

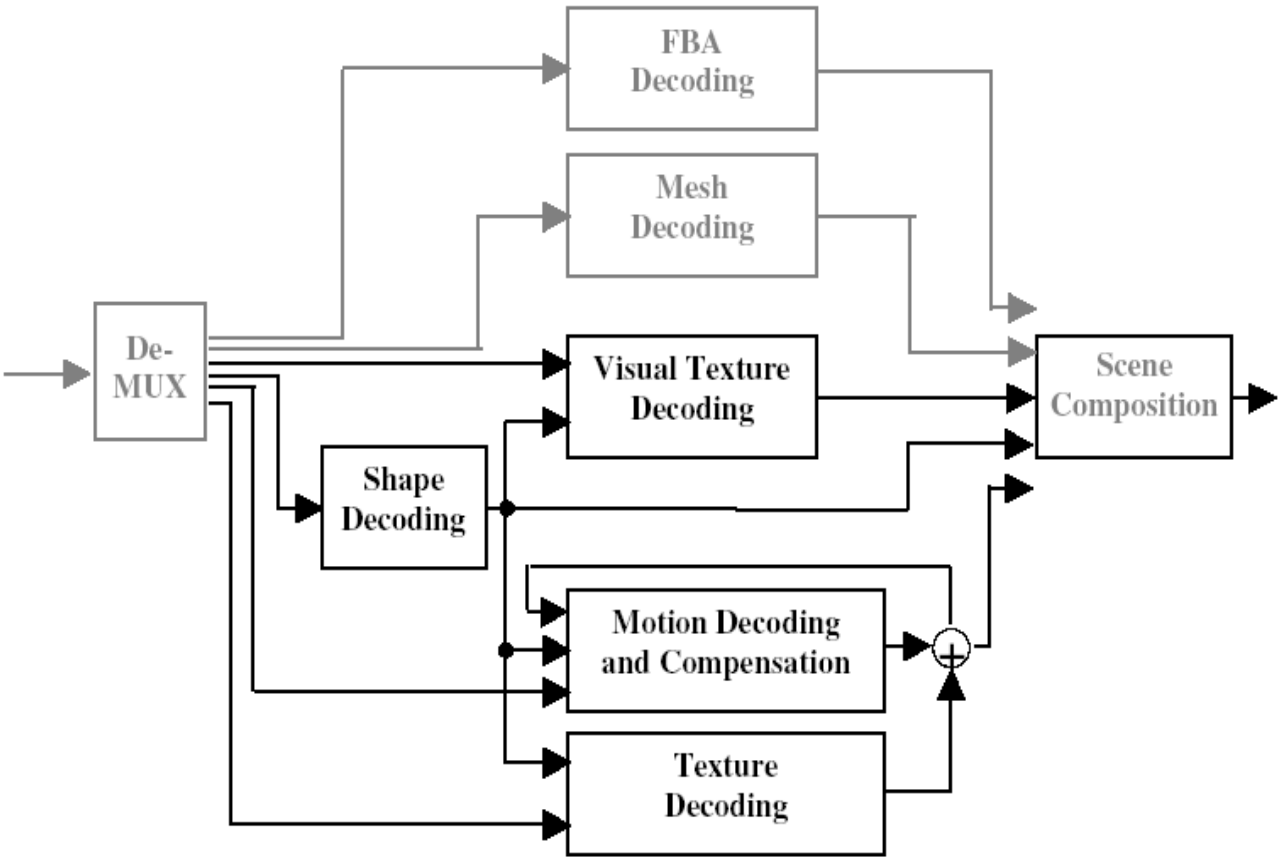
- (de)codifica di contenuti

- Interattività
- rappresentazione flessibile
- manipolazione di VO nel dominio compresso

- Complemento della compressione VLBV e HBV



- La shape può essere codificata come canale alfa 8 bit) o bitmask



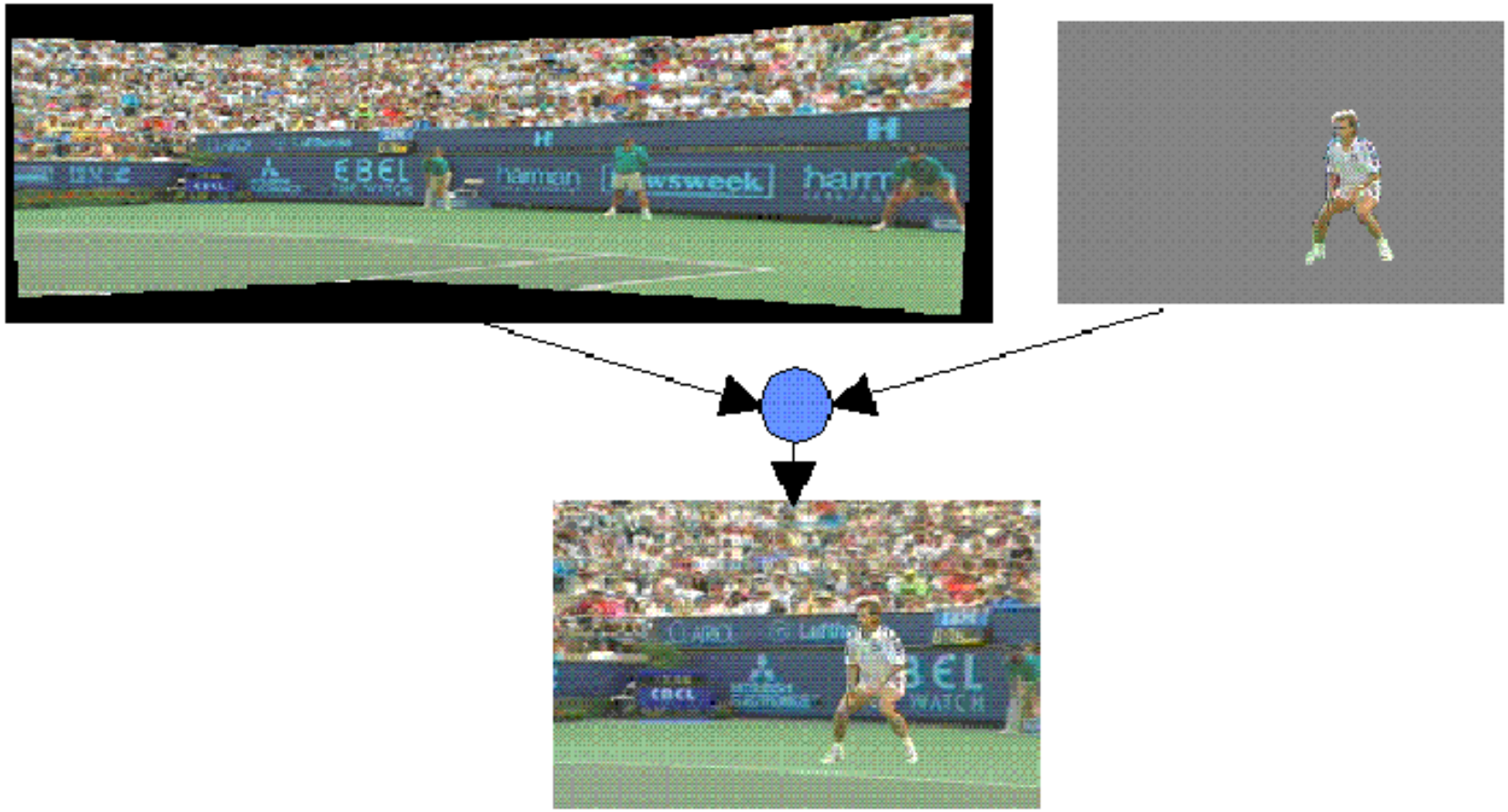
- Schema generale della codifica video MPEG4

- Si possono codificare immagini o sequenze video usando gli stessi tool:
  - Visual texture decoding: immagini
- Se non si tiene conto della shape si ha l'encoder basato su Motion compensation e Texture decoding:
  - Come MPEG1 e MPEG2
  - Anche in MPEG4 si usano i blocchi!

- È l'aggiunta dello shape decoding che consente la creazione della codifica content-based
  - Non solo forme rettangolari…
    - … e comunque anche quando si usa codifica object-based l'analisi dei frame è basata su blocchi!
  - La codifica delle forme rettangolari è comunque migliorata: bitrate più bassi a parità di qualità

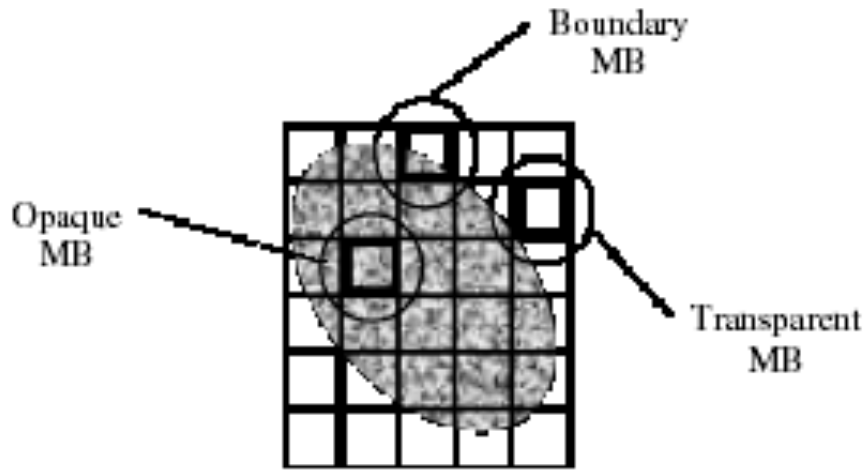
- La struttura base del codificatore include codifica di forma (per VOs di forma arbitraria), codifica di tessitura basata su compensazione di moto e DCT (utilizzando DCT 8x8 standard o DCT adattativa alla forma).
- Un importante vantaggio dell'approccio di codifica basato sul contenuto è che l'efficienza di compressione può essere migliorata in modo significativo per alcune sequenze video utilizzando tecniche di predizione del moto appropriate e dedicate basate su oggetti, per ciascun oggetto nella scena.

- Uno dei vantaggi della codifica content-based è che si può adattare la compressione secondo l'oggetto:
  - Es. background più compresso del foreground
  - Trasmissione di un'immagine panoramica (e parametri di moto della camera) + oggetto foreground come altro VO


EBEL
harman
harm

- L’algoritmo da usare per determinare la forma dell’oggetto da codificare non è definito, solo il bitstream per rappresentarlo
  - Ne esistono diversi completamente automatici, o assistiti

- Viene usata la bounding box dell'oggetto da codificare, dividendola in MB di 16x16. Eventualmente si ingrandisce la bb per renderla divisibile in MB



- La maggiore compressione di MPEG4 deriva da:
  - Motion compensation più precisa, si usa una risoluzione di ¼ di pixel (interpolando, ovviamente…)
  - Global motion compensation: si usa un solo set di parametri per l'intero VOP
  - Un miglioramento della predizione del moto nei B-VOP

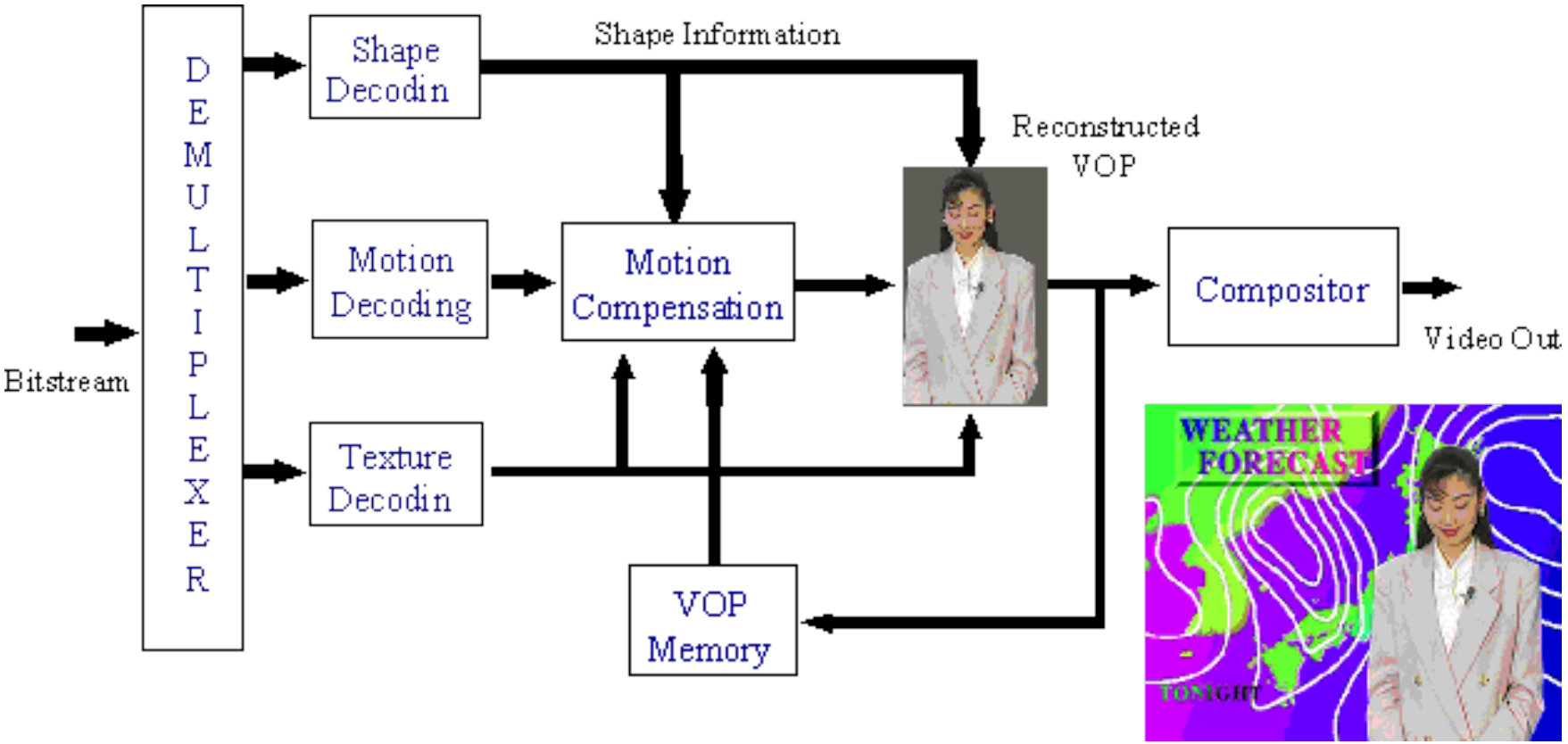
- Nota: se siamo nel caso in cui l'unico VO è il frame rettangolare allora VOP = frame di MPEG2

- Anche la codifica DCT è stata cambiata:
  - Due sistemi di quantizzazione diversi, uno stile MPEG2 ed uno che non usa tabella di quantizzazione (stile H.263)
  - Predizione di coefficienti AC/DC in intraMB
  - Due nuove scansioni della matrice risultante

Codifica tessiture e immagini

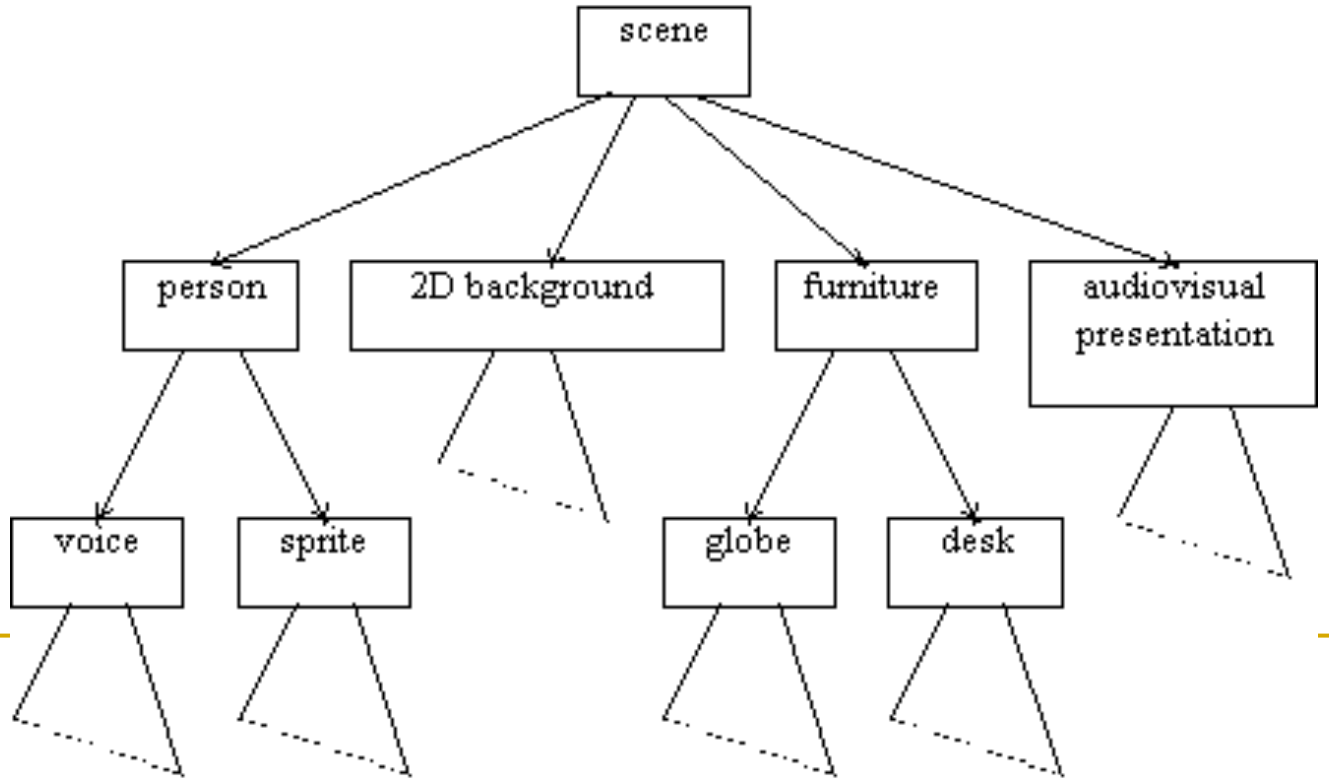
- Le immagini e le texture da applicare su mesh sono codificate con wavelet:
  - Ottima compressione
  - Scalabilità
  - Codifica shape di forma arbitraria

Esempio di decodifica video



Descrizione della scena

- In aggiunta a fornire il supporto per codificare singoli oggetti, MPEG-4 fornisce anche la possibilità di comporre un insieme di tali oggetti in una scena. L'informazione necessaria alla composizione, forma la descrizione della scena, codificata e trasmessa insieme con gli AVOs.



Raggruppamento degli oggetti

- Una scena MPEG-4 segue una struttura gerarchica che può essere rappresentata come un grafo diretto aciclico. Ogni nodo del grafo è un AVO.
  - La struttura non è necessariamente statica; gli attributi dei nodi (ad esempio parametri di posizione) possono essere cambiati mentre nodi possono essere aggiunti, rimpiazzati o rimossi.

Posizionamento nello spazio e nel tempo

- Nel modello MPEG-4, oggetti audiovisivi hanno una estensione sia spaziale che temporale. Ciascun AVO ha un sistema di coordinate locale.
  - Un sistema *locale di coordinate,* per un oggetto è un sistema in cui l'oggetto ha una locazione spazio-temporale e una scala fissate.
  - Il sistema di coordinate locale serve per manipolare l'AVO nello spazio e nel tempo. AVO sono posizionati in una scena specificando una trasformazione di coordinate dal sistema di coordinate locale all'oggetto, in un sistema di coordinate globale definito da uno o più nodi di descrizione della scena nell'albero.

Selezione del valore degli attributi

- Singoli AVO e nodi descrittori della scena espongono un insieme di parametri al layer di composizione attraverso i quali parte del loro comportamento può essere controllato.
  - Esempi includono la tonalità di un suono, il colore di un oggetto sintetico, l'attivazione o disattivazione di informazioni di accrescimento per codifica scalabile, etc.

Altre trasformazioni

- La struttura di descrizione della scena e la semantica dei nodi sono fortemente influenzate dal VRML, incluso il suo modello degli eventi. Questo rende disponibili a MPEG-4 un ampio insieme di operatori di costruzione della scena, includendo primitive grafiche che possono essere usate per costruire scene sofisticate.

Interazione utente

- MPEG-4 consente l'interazione dell'utente con il contenuto presentato. Questa interazione può essere distinta in:
  - interazione lato server;
  - interazione lato client.

- L'interazione lato server, implica manipolazione del contenuto che occorre al lato trasmittente iniziata da una azione dell'utente.

Interazione lato client

- L'interazione lato client implica manipolazione del contenuto localmente al terminale dell'utente finale e può assumere diverse forme:
  - modifica dell'attributo del nodo di descrizione di una scena (ad esempio, modifica della posizione di un oggetto, renderlo visibile o invisibile, modifica della dimensione del font per un nodo di testo sintetico) può essere implementata traducendo eventi dell'utente (click del mouse o comandi da tastiera) ad aggiornamenti della descrizione della scena.
  - I comandi possono essere elaborati dal terminale MPEG-4 esattamente nello stesso modo come se fossero stati originati dalla sorgente originale di contenuto.

Altre funzionalità

- La scalabilità spaziale, temporale e di qualità è a livello di VO
- Correzione e mascheramento degli errori

Implementazione

- L'implementazione di sistemi MPEG4 è piuttosto complicata: svariate migliaia di pagine di standard…
  - Ricerca attiva su alcuni algoritmi: es. tracking di oggetti all'interno di un video per estrarre VO

- Il formato dei file (basato su Quicktime) è *streamable*:
  - È adatto sia per uso locale che per lo streaming in rete
    - Fornisce informazioni al server di streaming mediante metadati contenuti nel file, è compito del server di streaming usarli all'interno di un protocollo di streaming

- Profili: subset delle funzioni di MPEG4,usati per applicazioni specifiche
- Ad ogni profilo si può associare un livello che limità la complessità computazionale
- Come in MPEG2: non importa implementare tutte le funzioni dello standard, mi posso limitare ad un profilo

Es.: profili visuali

- Es.: simple profile, codifica di oggetti visuali rettangolari con correzione di errori, adatto per applicazioni su terminali mobili
- Simple scalable profile: aggiunge il supporto per la scalabilità temporale e spaziale al simple profile. Adatto per servizi su Internet
- Core profile: aggiunge il supporto per oggetti di qualsiasi forma, scalabili temporalmente, al simple profile

- Esistono profili anche per:
  - animazioni facciali;
  - audio;
  - mesh;
  - grafica;
  - scene;
  - oggetti (IPR, object descriptor, etc.)

- Divx:
  - Nato da reverse engineering di implementazione Microsoft, per attivare funzioni altrimenti disattivate
- Opendivx / XVid:
  - Implementazione open-source
  - Simple profile