MPEG-2

4C8: Digital Media Processing

Ussher Assistant Professor François Pitié

2021/2022

Department of Electronic & Electrical Engineering , Trinity College Dublin

adapted from original material written by Prof. Anil Kokaram.

As a sequel to the JPEG standards committee, the Moving Picture Experts Group (MPEG) was set up in the mid 1980s to agree standards for video sequence compression.

We shall not go into the detailed differences between these standards, but simply describe some of their important features.

A Brief History

Two major standards organisations define compression standards

- ITU-T - H.26x
- ISO – MPEG-x

H.261 was the first widely used compression standard. Finalised in 1988, it was designed for video phone applications, with low resolution (QCIF 176x144) transmission over an ISDN line (64 kb/s), using YCbCr colourspace and 4:2:0 subsampling (88x72 Cb and Cr values in total).

A Brief History

MPEG-1. Designed for CD-ROM applications (VCD) to compress VHS quality video (PAL or NTSC) at bit rates as low has 1.5 Mbps. The capacity of a CD-ROM is about 700MB – enough for about 1 hour of video at 1.5 Mbps. This format is better known for its audio standard MPEG-1 part 3 (i.e. mp3).

MPEG-2/H.262 (mid 90s). Designed for digital SD TV, PAL (720x576) or NTSC (720x480) at 4 to 10 Mbps. It supports interlaced videos, different aspect ratios (16:9 and 4:3) and frame rates also used for DVD (about 9 Mbps) & HDTV (15 to 20 Mbps).

A Brief History

H.263 (1996). Designed to supersede H.261 but also supports higher resolutions, it achieves equivalent quality with lower bit rates. Very popular format for early web as it could be used to transmit video over dial-up modems (ie. 28 kpbs / 56 kbps). It was adopted in flash video, the dominant format on the web at the time.

MPEG-4 (part 2) (1999). Based on H.263 and designed for all resolutions and bit rates Some DivX and Xvid codecs use this format.

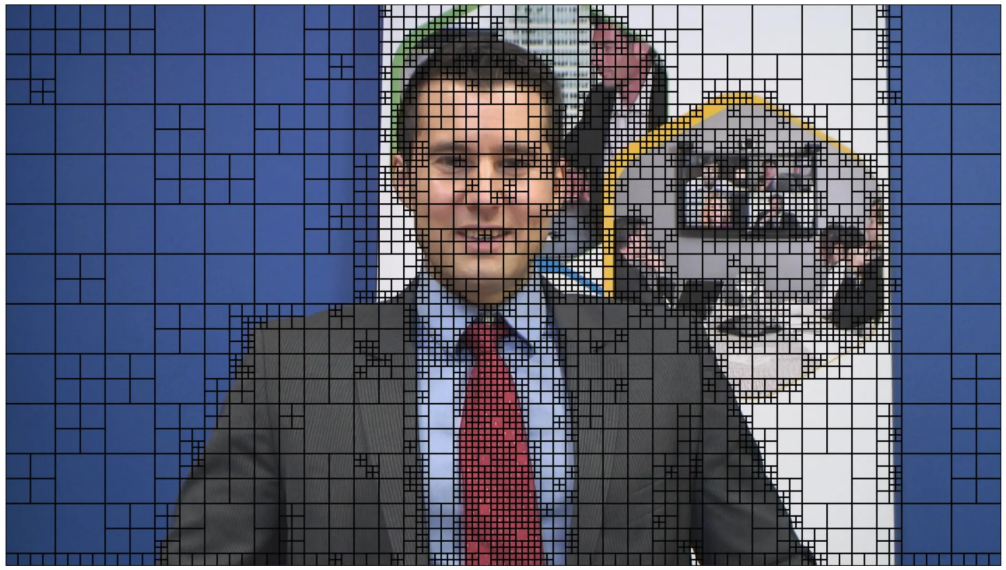
A Brief History

MPEG-4 (part 10)/H.264/AVC (2004). Initially designed to achieve lower bit rates than MPEG-2 and H.263 (50% reduction in most cases), widely adopted for use in digital TV (Saorview and Freeview HD) has increased error resilience so is suitable for use on mobiles.

VP8. Aims to avoid use of patented technologies Developed by ON2, acquired by Google in 2010.

A Brief History

HEVC/H.265 (2013), **VP9** (2013). Successors of H264 and VP8. Main improvements comes from the possibility to segment frames into blocks of variable sizes (from 4×4 to 64×64), in a process called picture partitioning. Both use arithmetic coding for entropy coding. Use of 1/8th pel motion compensation. These two codecs have become standards for 4k compression.



credits: https://sonnati.wordpress.com/2014/06/20/h265-part-i-technical-overview/

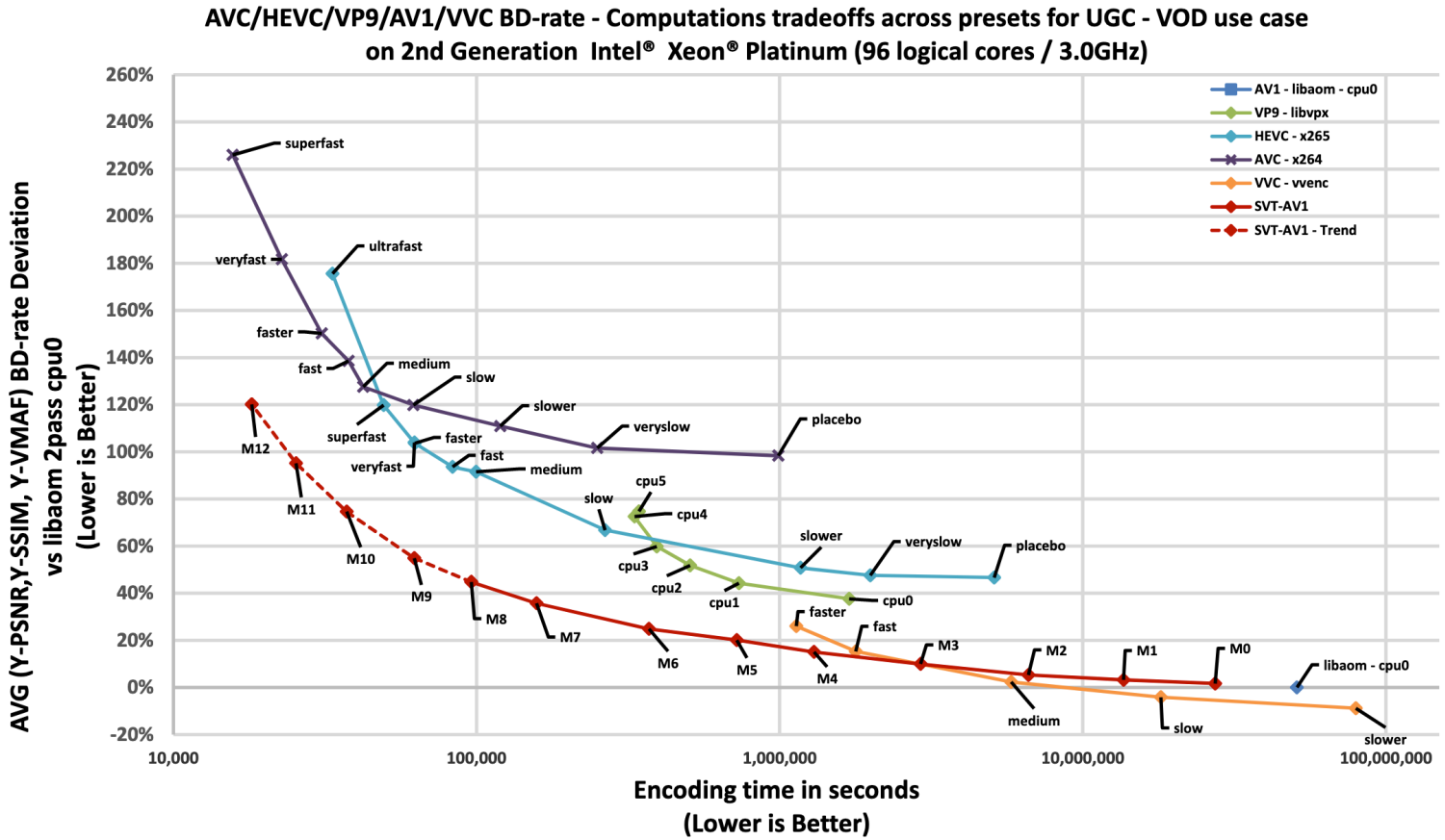
A Brief History

VVC/H266 (2020)/**AV1** (2018). Successors of H265 and VP9. In 2015, Google started the Alliance for Open Media (AOMedia), a consortium to sponsor the open source codec.

Newer codecs keep on improving performance but at the cost of significant computational encode time.

A Brief History

Basically each new codec brings a 50% bitrate improvement for the same picture quality. The trade-off is that each new codec requires $10\times$ the computational complexity. It can take hours to encode a simple 4s clip with the newest AV1 codec.



credits: Ping-Hao et al. 2021

MPEG-2

Video compression is concerned with coding image sequences at low bit rates. In an image sequence, there are typically high correlations between consecutive frames of the sequence, in addition to the spatial correlations which exist naturally within each frame.

Video coders aim to take maximum advantage of *interframe* temporal correlations (between frames) as well as *intraframe* spatial correlations (within frames).

Video codecs typically consider two types of frames: the key-frames that are encoded using a JPEG-type compression, and motion predicted frames, which are encoded using other previously decoded frames as reference.

Frame Types

These are three types of frame found in MPEG-2:

I-frames: These are *Intra* coded frames, which are coded as single frames as in JPEG, without reference to any other frames.

P-frames: *Predictive* coded frames, which are coded as the difference from a motion compensated prediction frame, generated from an earlier I or P frame in the GOP.

B-frames: *Bi-directional* coded frames, which are coded as the difference from a bi-directionally interpolated frame, generated from earlier and later I or P frames in the sequence (with motion compensation).

I-frames

For the *intraframe* spatial correlations, codecs encode keyframes using a JPEG-type encoding, ie. with partitioning of the frame into blocks, then apply the DCT and quantisation.

As with JPEG, the MPEG committee has selected default values for the quantisation matrices for the 8×8 DCT block.

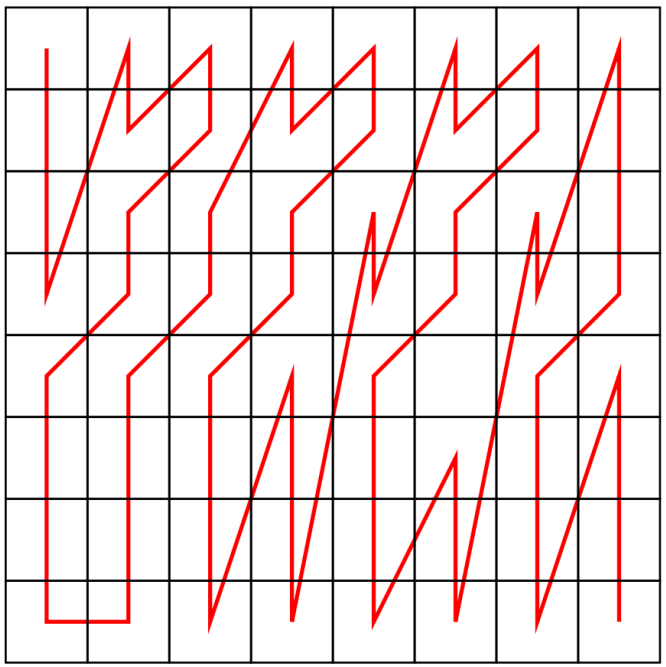
The Default Intra Quantisation Matrix in MPEG-2:

$$\begin{pmatrix} 8 & 16 & 19 & 22 & 26 & 27 & 29 & 34 \\ 16 & 16 & 22 & 24 & 47 & 49 & 34 & 37 \\ 19 & 22 & 26 & 27 & 29 & 34 & 34 & 38 \\ 22 & 22 & 26 & 27 & 29 & 34 & 37 & 40 \\ 22 & 26 & 27 & 29 & 32 & 35 & 40 & 48 \\ 26 & 27 & 29 & 32 & 35 & 40 & 48 & 58 \\ 26 & 27 & 29 & 34 & 38 & 46 & 56 & 69 \\ 27 & 29 & 35 & 38 & 46 & 56 & 69 & 83 \end{pmatrix}$$

Interlacing Fun

MPEG-2 being developed with Broadcasting in mind, it was critical to also include considerations for the dreaded interlacing.

Below is the alternate DC coefficient scan for 8×8 DCT block for interlaced frames.



Fun :-)

P-frames: Motion-Compensated Predictive Coding

Motion-compensated predictive coding (MCPC) is the technique that has been found to be most successful for exploiting interframe correlations. In MPEG-2 and other codecs, these frames are called *predictive* frames, or **P-frames**.

Motion-Compensated Predictive Coding

The idea is simply to encode the difference between the current image $I_n(\mathbf{x})$ and the motion compensated previous image $I_{n-1}(\mathbf{x}+\mathbf{d}_{n,n-1}(\mathbf{x}))$:

$$DFD_{n,n-1}(\mathbf{x}) = I_n(\mathbf{x}) - I_{n-1}(\mathbf{x} + \mathbf{d}_{n,n-1}(\mathbf{x}))$$

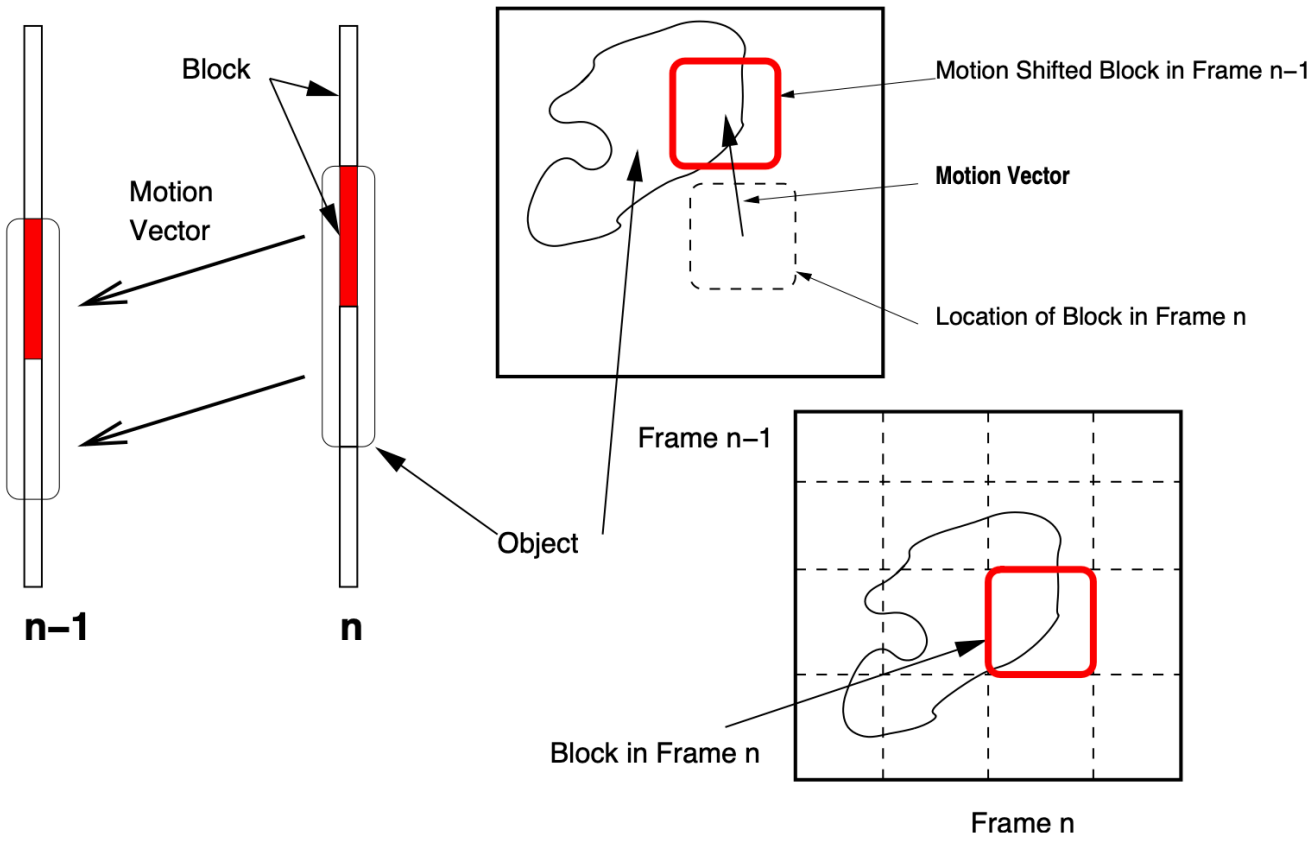
Then we can use a JPEG-like encoding on the DFD. This difference frame is usually known as the prediction error frame.

The assumption is that the DFD is correlated with the entropy. Large DFD equates to high bitrate and low DFD equates to low bitrate.

If the DFD is too large, we usually revert to a I-frame type encoding for that particular block.

Motion-Compensated Predictive Coding

A reminder of how motion compensation works:



Motion-Compensated Predictive Coding

Original Sequence

DFD without motion compensation

DFD with motion compensation

Motion-Compensated Predictive Coding

A subtlety is that the we can only use frames that are accessible at *decode time*. Thus, instead of the original I_{n-1}, we only have access to $\tilde{I}_{n-1}$, the *decoded* previous frame.

This means that the motion vectors $\tilde{\mathbf{d}}_{n,n-1}(\mathbf{x})$ are computed between I_n and $\tilde{I}_{n-1}$. Similarly the dfd is computed as:

$$\tilde{DFD}(\mathbf{x}) = I_n(\mathbf{x}) - \tilde{I}_{n-1}(\mathbf{x} + \tilde{\mathbf{d}}_{n,n-1}(\mathbf{x})).$$

To understand this, consider the simpler case where there is NO motion compensation, ie. we set $\mathbf{d_{n,n-1}} = \mathbf{0}$. If we incorrectly compress the original DFD, the reconstruction is as follows:

$$\tilde{I}_n(\mathbf{x}) = \tilde{I}_{n-1}(\mathbf{x}) + \mathbf{Encode}(I_n(\mathbf{x}) - I_{n-1}(\mathbf{x})) \tag{1}$$

$$= \tilde{I}_{n-1}(\mathbf{x}) + \left(I_n(\mathbf{x}) - I_{n-1}(\mathbf{x}) + \Delta_{n,n-1}\right) \tag{2}$$

where $\Delta_{n,n-1}(\mathbf{x})$ is the error introduced by the quantisation

$$\tilde{I}_n(\mathbf{x}) - I_n(\mathbf{x}) = \tilde{I}_{n-1}(\mathbf{x}) - I_{n-1}(\mathbf{x}) + \Delta_{n,n-1}(\mathbf{x}) \tag{3}$$

$$\Delta_n(\mathbf{x}) = \Delta_{n-1}(\mathbf{x}) + \Delta_{n,n-1}(\mathbf{x}) \tag{4}$$

As we can see, the error per frame Δ_n will compound at each frame. Instead, if we compress the DFD with the previously *decoded* frame:

$$\tilde{I}_n(\mathbf{x}) = \tilde{I}_{n-1}(\mathbf{x}) + \mathbf{Encode}(I_n(\mathbf{x}) - \tilde{I}_{n-1}(\mathbf{x})) \tag{5}$$

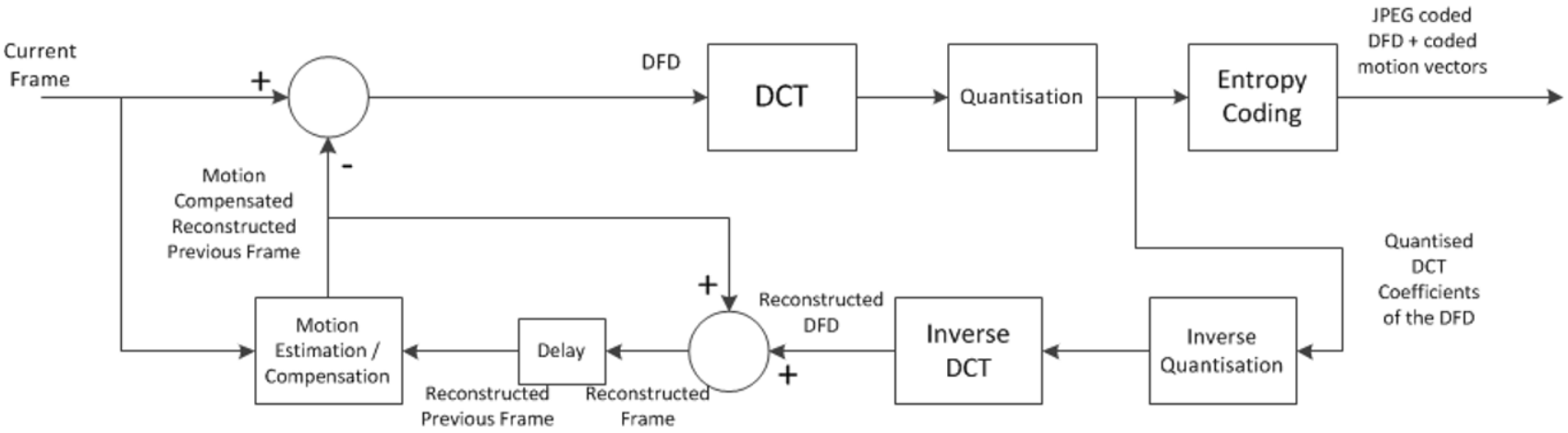
$$= \tilde{I}_{n-1}(\mathbf{x}) + \left(I_n(\mathbf{x}) - \tilde{I}_{n-1}(\mathbf{x}) + \tilde{\Delta}_{n,n-1}\right) \tag{6}$$

$$\tilde{I}_n(\mathbf{x}) - I_n(\mathbf{x}) = \tilde{\Delta}_{n,n-1}(\mathbf{x}) \tag{7}$$

The error doesn't accumulate.

Motion-Compensated Predictive Coding

The system looks as follows:



We first take the difference between the current frame and the motion compensated decoded previous frame. Then, we encode this difference using a JPEG-like compression (DCT, quantisation, entropy coding). The decoded difference is then used to reconstruct the decoded current frame, which is then used for ME and comparison with the next frame.

Motion-Compensated Predictive Coding

Motion estimation is thus the key to this kind of predictive coding. Note, however, that we are here not interested in finding the ground-truth motion vectors, but only in finding similar image blocks.

This means that our use of motion estimation cannot fail, or, more accurately, a failure of motion estimation only means that the resulting DFD error is high, and thus might require more bits to encode.

Thus, we do not need to invoke complex modern motion estimators and that a simple block-matching algorithm will be sufficient.

Quantisation of P-frames

As compression for the Inter-coded images is done on the DFD instead of the actual image, we should employ a different quantisation matrix.

The Default Inter Quantisation Matrix in MPEG-2:

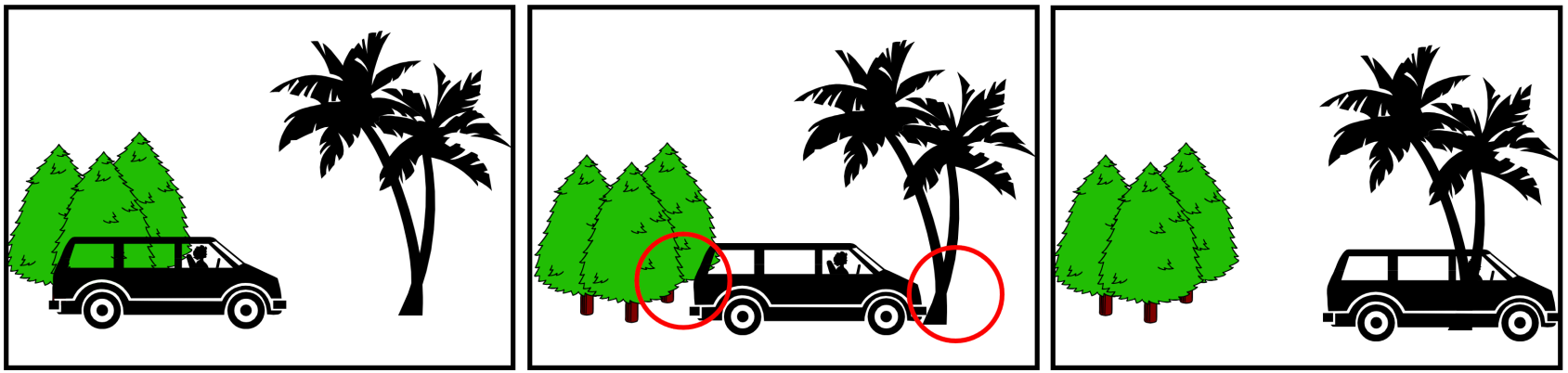
$$\begin{pmatrix} 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \\ 16 & 16 & 16 & 16 & 16 & 16 & 16 & 16 \end{pmatrix}$$

B-frames

MPEG-2 also introduced a *bi-directional* coded frames, or B-frames, which are a variant using two reference frames instead on one, as in P-frames.

The idea is now to compute the motion vectors between the current frame and two reference frames, and try to reconstruct the image from both.

B-frames



At frame Y, the areas, highlighted in red, are only visible in the previous frame X or in the forward frame Z. Hence it could be useful to predict images from multiple keyframes.

B-frames

Say we have already decoded forward frame I_{n+j} and backward frame I_{n-i}. Then, to reconstruct I_n, we have 2 DFDs:

$$DFD_b = I_n(\mathbf{x}) - I_{n-i}(\mathbf{x} + \mathbf{d}_{n,n-i}(\mathbf{x}))$$

$$DFD_f = I_n(\mathbf{x}) - I_{n+j}(\mathbf{x} + \mathbf{d}_{n,n+j}(\mathbf{x}))$$

We keep the vector that gives the best prediction:

$$\hat{I}_n(\mathbf{x}) = \begin{cases} I_{n-i}(\mathbf{x} + \mathbf{d}_{n,n-i}(\mathbf{x})) & \text{if } |DFD_f| > |DFD_b| \\ I_{n+j}(\mathbf{x} + \mathbf{d}_{n,n+j}(\mathbf{x})) & \text{if } |DFD_f| \leq |DFD_b| \end{cases}$$

Macroblocks and Motion Vectors Encoding

In MPEG-2 motion estimation and compensation is carried out on blocks of 16×16 and NOT 8×8. To avoid confusion, these blocks are called **macroblocks**. Thus only ONE vector is transmitted per 16×16 macroblock for P Pictures and up to TWO for B Pictures.

Motion vectors are encoded using difference coding. (ie. only the difference between a macro block motion vector and the previously estimated motion vector on the macroblock to the left is sent for entropy coding.)

In MPEG-2 motion vectors are quantised to ± 0.5 pixel accuracy.

Group of Pictures (GOP) Layer

A **group of pictures**, or **GOP structure**, specifies the order in which these frames types are arranged in a batch of frames.

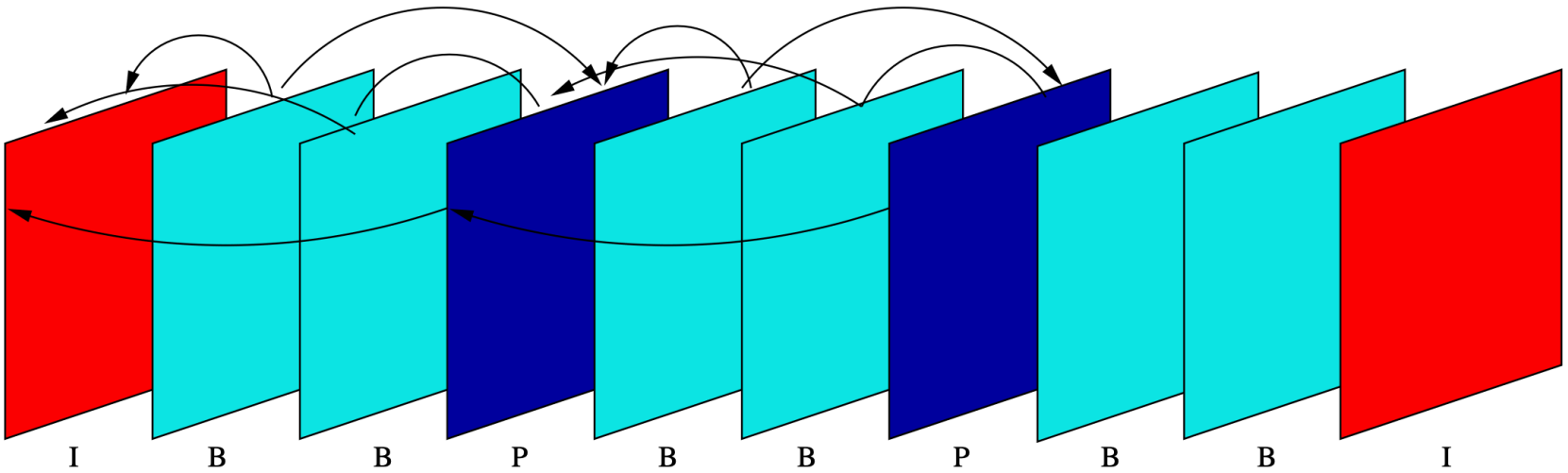
The GOP Layer typically contains thus a small number of frames, for about half a second of video, coded so that they can be decoded completely as a unit, without reference to frames outside of the group.

The most typical GOP structure is made of 12 frames and is as follows:

IBBPBBPBBPBB (repeated)

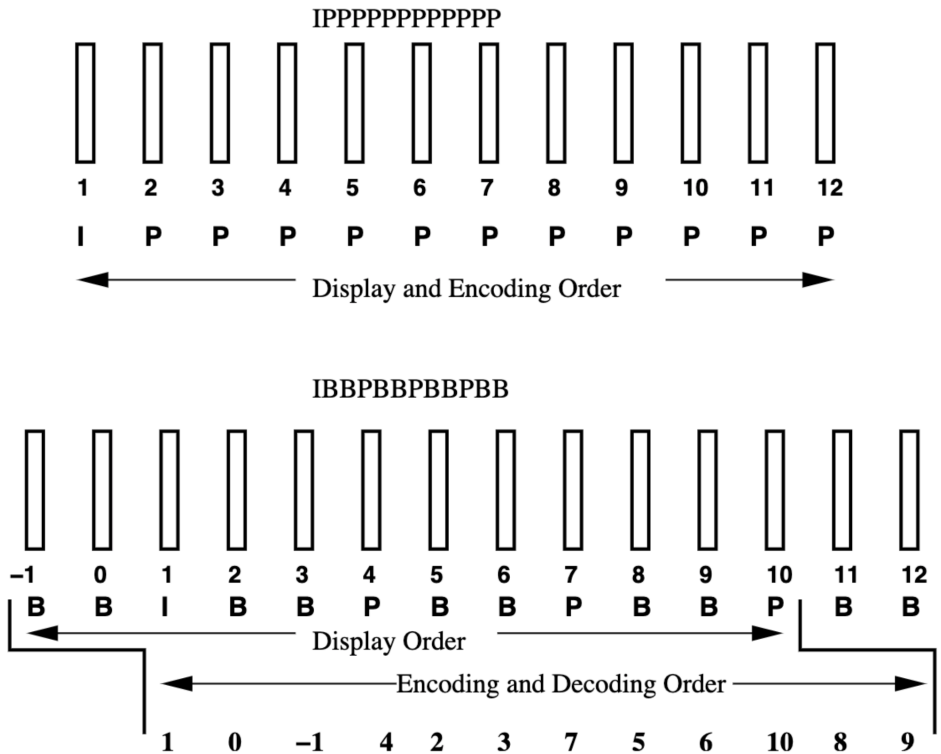
Group of Pictures (GOP) Layer

Here is a shorter Group of Pictures (GOP):



Group of Pictures (GOP) Layer

Note that, because of B-frames, the decode order of the frames in a GOP is not the same as the display order, as we might need to decode frames that ahead in the sequence.



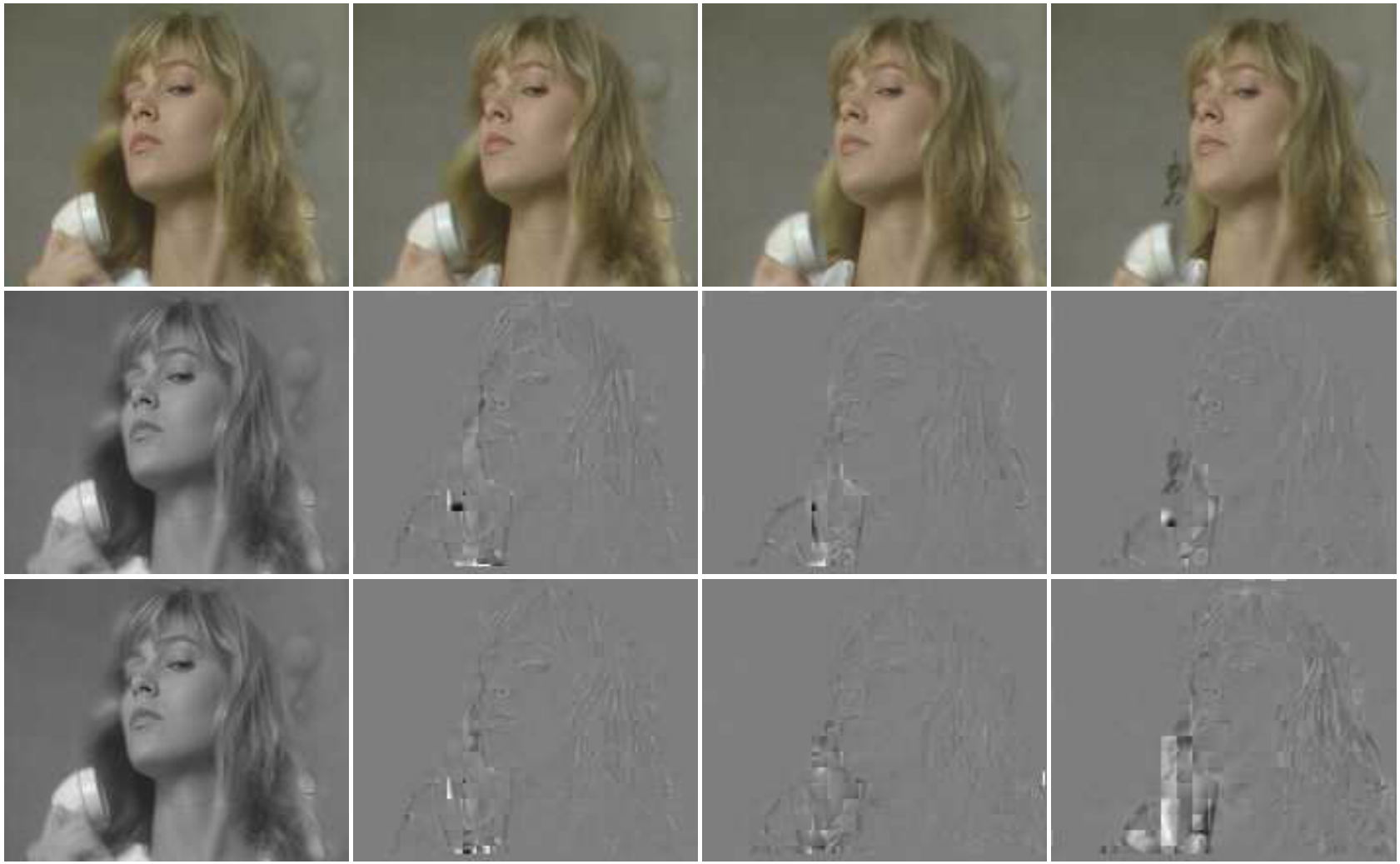
Group of Pictures (GOP) Layer

The main purpose of the GOP is to allow editing and splicing of video material from different sources and to allow rapid forward or reverse searching through sequences.

For instance, what would be the advantages and disadvantages of a IPPPPPPPP...structure?

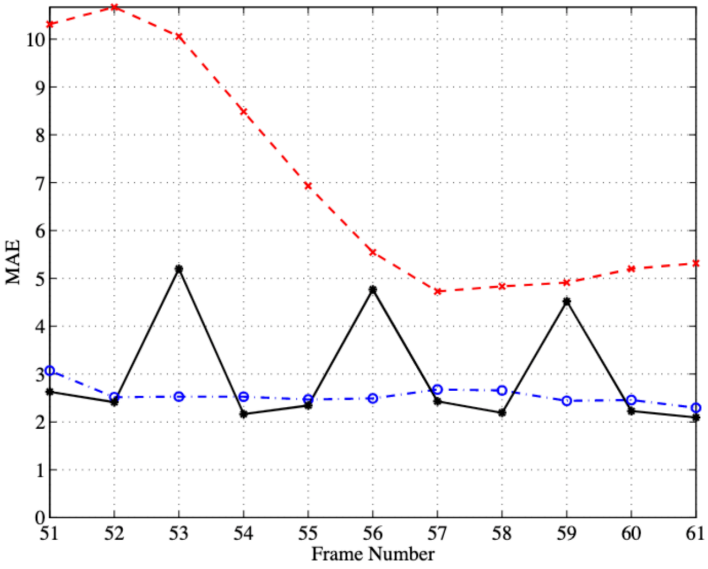
IPPP vs IBBP

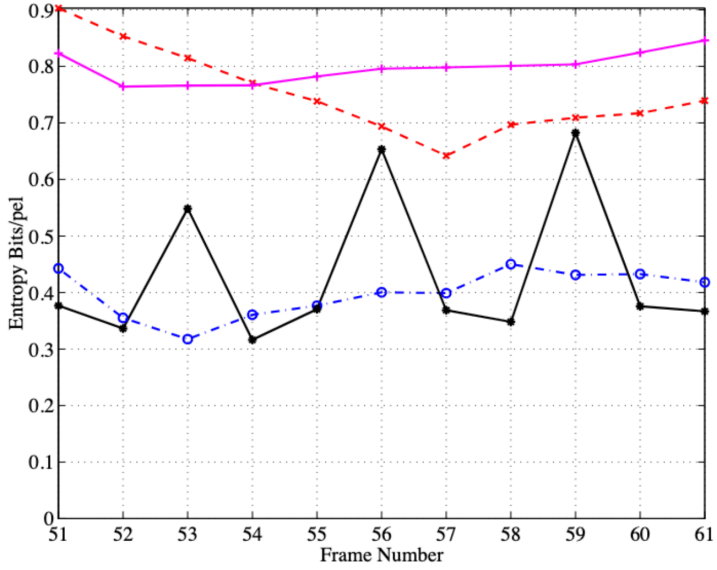
Let's compare in more detail the performance of IPPP and IBBP.



Top: original sequence. Middle: Motion compensated DFD with IPPP.
Bottom: Motion compensated DFD with IBBP.

IPPP vs IBBP





Comparison of MAE (left) and Entropy (right) of the DCT for same seq. Original (+), no motion compensation (x), IPPP (o), IBBP (*).

Spot the peaks in the IBBPBBP plots. Also, the Entropy for the P frames in the IBBP sequence is higher than for IPPP. Why?

MPEG-2 Bitstream Organisation

The MPEG-2 bitstream is organised with a strict hierarchy, like an onion. The layers are as follows:

1. **Sequence Layer:** contains an entire video sequence with thousands of frames.
2. **GOP Layer:** frame-ordering and description of frame-types.
3. **Picture Layer:** contains all the encoded data referring to a single I, P or B frame.
4. **Slice Layer:** a sequence of macroblocks within a row. Registers are reset at start of slice, so as to avoid error propagation.
5. **Macroblock Layer:** contains four 8×8 blocks, as well as a motion vector if using a non I-frames.
6. **Block Layer:** contains the DCT coefficients for a single block.

Other Considerations

Other Considerations

MPEG-2 is about more than video coding

Part 1 Systems (describes how audio and video are plugged together)

Part 2 Video

Part 3 Audio (an extension of the MPEG-1 audio standards)

Part 4 conformance testing

Part 5 software simulation

Part 6 Digital Storage Media Command and Control – (eg. rewind forward etc etc)

Part 7 Advanced Audio Coding (AAC) – a 2nd audio standard there are even more parts

Other Considerations

In any multimedia transmission system that involves compression the following issues become important.

Other Considerations

Data Type. Multimedia data sources can be pictures, audio and text. Different compression techniques are needed for each data type. Each piece of data has to be identified with unique codewords for transmission.

Other Considerations

Sequencing. The compressed data from each source is scanned into a sequence of bits. This sequence is then packetised for transport. The problem here is to identify each different part of the bitstream uniquely to the decoder, *e.g.* header information, DCT coefficient information.

Other Considerations

Multiplexing The audio and video data (for instance) has to be decoded at the same time to create a coherent signal at the receiver. This implies that the transmitted elementary data streams should be combined so that they arrive at the correct time at the decoder. The challenge is therefore to allow for identifying the different parts of the multiplexed stream and to insert information about the timing of each elementary data stream.

Other Considerations

Media. The compressed and multiplexed data has to be stored on some digital storage and then later (or live) broadcast to receivers across air or other links. Access to different Media channels is governed by different constraints and this must somehow be allowed for in the standards description.

Other Considerations

Errors. Errors in the received bitstream invariably occur. The receiver must cope with errors such that the system performance is robust to errors or it degrades in some graceful way.

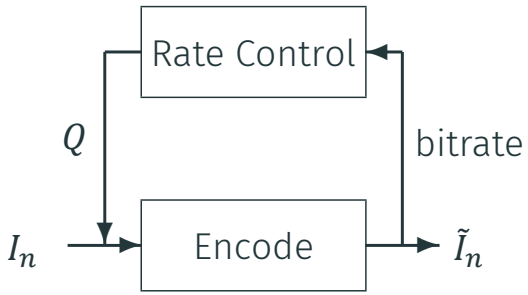
Other Considerations: Rate Control

Bandwidth. The bandwidth available for the multimedia transmission is limited. The transmission system must ensure that the bandwidth of the bitstream does not exceed these limits.

The problem of tuning the encoder to meet these expectations is called **Rate Control** and applies both to the control of the bitrate of the elementary data streams and the multiplexed stream.

Other Considerations: Rate Control

Rate Control is a complicated business, which has seen many improvements over the years.



The principle is however quite simple. If the bitrate is currently too high/too low, the rate control algorithm needs to increase/decrease the quantisation for the next frame. This is a feedback loop and the difficulty is to predict the most efficiently the updated value of Q.

Other Considerations: Rate Control

If the parameters are not well adjusted, it may take some time for the rate controller to adjust Q, and the first few frames may be of poor quality (a bit like in a PID system). This is especially a problem if the video content is very dynamic.

The problem can be partially solved by using a two-pass approach, where a quick first pass is used to estimate the relative size of each frame in a sequence. Scaling these values allow us to better estimate the required bits allocation per frame.

...but you still need to somehow to estimate what value Q will give you a targeted bitrate, which is a hard problem.

Note that Rate Control falls outside of the codec specifications and is left to the encoder. As it has a big impact on performance, it is currently an active area of research, even for older codecs (eg. it can help Netflix or YouTube deliver the best quality at a particular bitrate).

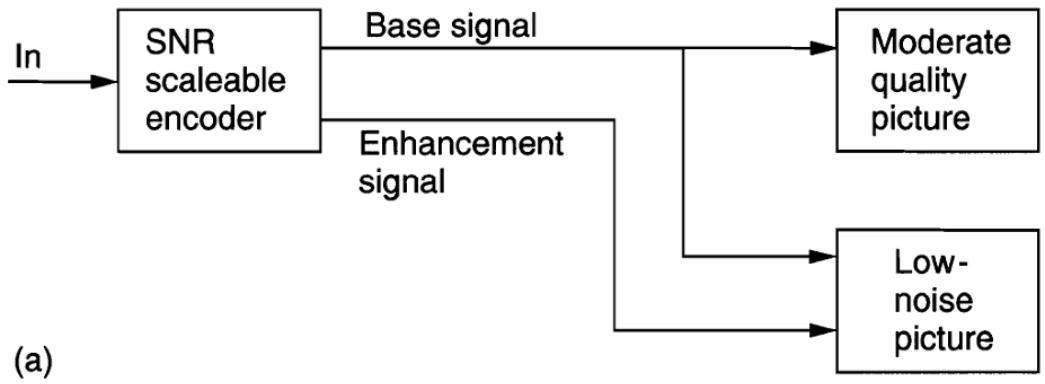
Other Considerations

Multiplatform. The coded bitstream may need to be decoded on many different types of device with varying processor speeds and storage resources. It would be interesting if the transmission system could provide a bitstream which could be decoded to varying extents by different devices. Thus a low capacity device could receive a lower quality picture than a high capacity device that would receive further features and higher picture quality. This concept applied to the construction of a suitable bitstream format is called Scalability.

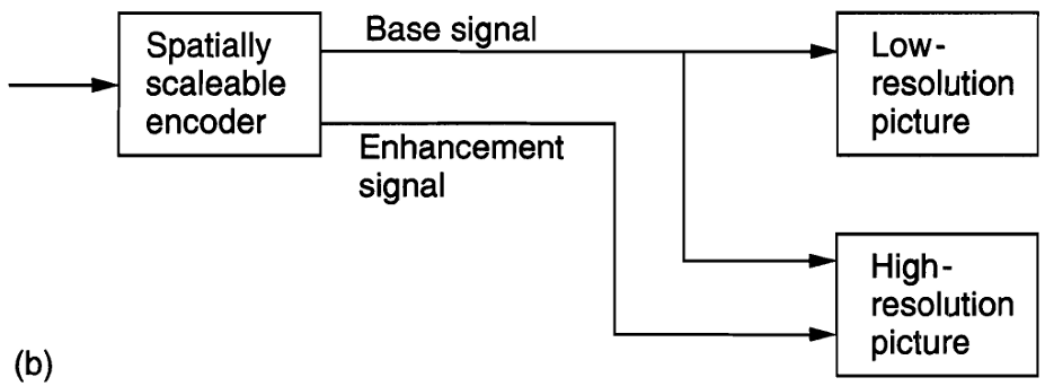
Scalable Video Coding

The idea behind spatial/SNR scalability is that a base layer carries a compressed lower spatial resolution/lower quality video and additional enhancement layer(s) carry the encoded difference/improvements.

(a) SNR Scalability



(b) Spatial Scalability



Profiles and Levels

The MPEG standards have been designed to cope with a wide variety of picture formats and frame rates. However not all decoders and encoders will be able to cope with all possible combinations of input picture formats. Therefore a hierarchy of data formats was specified such that the capability of a Codec could be defined in terms of a combination of various input data options allowed by the standard.

The standard does not define encoders or decoders themselves, it only defines the format of a bitstream. This format implicitly defines much of the decoder but creation of the bitstream (the encoder) is outside the standard.

Profiles

Profile	Supported Features
HIGH	Supports the SPATIAL Scalable Profile + 3 layers with SNR and Spatial Scalable coding modes 4:2:2 YUV picture format
SPATIAL	Supports the SNR Scalable Profile + Spatial Scalable coding modes (2 layers) 4:0:0 YUV
MAIN	Supports SIMPLE profile + B-picture prediction modes
SIMPLE	Supports coding progressive and interlaced video random access I, P-picture prediction modes 4:2:0 YUV

Levels

Level	Upper Bound on Parameters
HIGH	1920 samples/line , 1152 lines/frame 60 frames/sec 80 Mbits/sec
HIGH 1440	1440 samples/line, 1152 lines/frame 60 Frames/sec 60 Mbits/sec
MAIN	720 samples/line, 576 lines/frame 30 Frames/sec 30 Mbits/sec
LOW	352 samples/line, 288 lines/frame 30 frames/sec 4 Mbits/sec

Multiplexing

Audio and video streams are divided into packets that are interleaved by the encoder.

To solve multiplexing problems, synchronisation between packets is achieved by the use of time stamps.

The **Decoding Time Stamp (DTS)** flag tells the decoder when to decode a packet. The **Presentation Time Stamp (PTS)** flag tells the decoder when to pass the decoded frame to the output device for display.

Going Further

The book *Digital Video: An Introduction to MPEG-2, Barry Haskell, Atul Purui and Arun Netravali, Chapman and Hall 1997* is an excellent overview of MPEG-2.

Prof. Inald Lagendijk at the Delft University of Technology in The Netherlands has an excellent MPEG-2 demo tool called VCDEMO which he uses to teach an image and video coding course. It is available free on the web.