**MPEG Compression**

The acronym MPEG stands for Moving Picture Expert Group, which worked to generate the specifications under ISO, the International Organization for Standardization and IEC, the International Electrotechnical Commission. What is commonly referred to as "MPEG video" actually consists at the present time of two finalized standards, MPEG-11 and MPEG-22, with a third standard, MPEG-4, was finalized in 1998 for ***Very Low Bitrate Audio-Visual Coding***. The MPEG-1 and MPEG-2 standards are similar in basic concepts. They both are based on motion compensated block-based transform coding techniques, while MPEG-4 deviates from these more traditional approaches in its usage of software image construct descriptors, for target bit-rates in the very low range, < 64Kb/sec. Because MPEG-1 and MPEG-2 are finalized standards and are both presently being utilized in a large number of applications, this paper concentrates on compression techniques relating only to these two standards. Note that there is no reference to MPEG-3. This is because it was originally anticipated that this standard would refer to HDTV applications, but it was found that minor extensions to the MPEG-2 standard would suffice for this higher bit-rate, higher resolution application, so work on a separate MPEG-3 standard was abandoned.

The current thrust is MPEG-7 "Multimedia Content Description Interface" whose completion is scheduled for July 2001. Work on the new standard MPEG-21 "Multimedia Framework" has started in June 2000 and has already produced a Draft Technical Report and two Calls for Proposals.

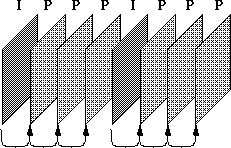
MPEG-1 was finalized in 1991, and was originally optimized to work at video resolutions of 352x240 pixels at 30 frames/sec (NTSC based) or 352x288 pixels at 25 frames/sec (PAL based), commonly referred to as Source Input Format (SIF) video. It is often mistakenly thought that the MPEG-1 resolution is limited to the above sizes, but it in fact may go as high as 4095x4095 at 60 frames/sec. The bit-rate is optimized for applications of around 1.5 Mb/sec, but again can be used at higher rates if required. MPEG-1 is defined for progressive frames only, and has no direct provision for interlaced video applications, such as in broadcast television applications.

MPEG-2 was finalized in 1994, and addressed issues directly related to digital television broadcasting, such as the efficient coding of field-interlaced video and scalability. Also, the target bit-rate was raised to between 4 and 9 Mb/sec, resulting in potentially very high quality video. MPEG-2 consists of profiles and levels. The profile defines the bitstream scalability and the colorspace resolution, while the level defines the image resolution and the maximum bit-rate per profile. Probably the most common descriptor in use currently is Main Profile, Main Level (MP@ML) which refers to 720x480 resolution video at 30 frames/sec, at bit-rates up to 15 Mb/sec for NTSC video. Another example is the HDTV resolution of 1920x1080 pixels at 30 frame/sec, at a bit-rate of up to 80 Mb/sec. This is an example of the Main Profile, High Level (MP@HL) descriptor.

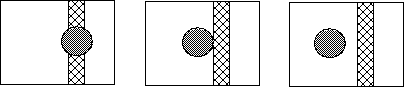
## MPEG Video

MPEG compression is essentially a attempts to over come some shortcomings of H.261 and JPEG:

* Recall H.261 dependencies:

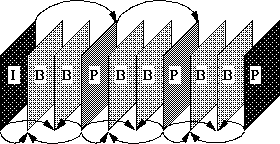


* The Problem here is that many macroblocks need information is **not** in the reference frame.
* For example:



* The **MPEG solution** is to add a third frame type which is a bidirectional frame, or *B-frame*
* B-frames search for macroblock in *past* and *future* frames.
* Typical pattern is IBBPBBPBB IBBPBBPBB IBBPBBPBB

Actual pattern is up to encoder, and need not be regular.

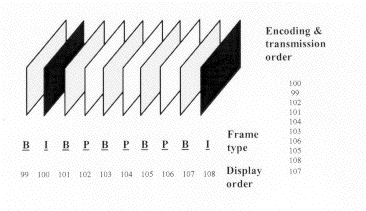


### MPEG Video Layers

MPEG video is broken up into a hierarchy of layers to help with error handling, random search and editing, and synchronization, for example with an audio bitstream. From the top level, the first layer is known as the video sequence layer, and is any self-contained bitstream, for example a coded movie or advertisement. The second layer down is the group of pictures, which is composed of 1 or more groups of intra (I) frames and/or non-intra (P and/or B) pictures that will be defined later. Of course the third layer down is the picture layer itself, and the next layer beneath it is called the slice layer. Each slice is a contiguous sequence of raster ordered macroblocks, most often on a row basis in typical video applications, but not limited to this by the specification. Each slice consists of macroblocks, which are 16x16 arrays of luminance pixels, or picture data elements, with 2 8x8 arrays of associated chrominance pixels. The macroblocks can be further divided into distinct 8x8 blocks, for further processing such as transform coding. Each of these layers has its own unique 32 bit start code defined in the syntax to consist of 23 zero bits followed by a one, then followed by 8 bits for the actual start code. These start codes may have as many zero bits as desired preceding them.

### B-Frames

The MPEG encoder also has the option of using forward/backward interpolated prediction. These frames are commonly referred to as bi-directional interpolated prediction frames, or B frames for short. As an example of the usage of I, P, and B frames, consider a group of pictures that lasts for 6 frames, and is given as I,B,P,B,P,B,I,B,P,B,P,B,Š As in the previous I and P only example, I frames are coded spatially only and the P frames are forward predicted based on previous I and P frames. The B frames however, are coded based on a forward prediction from a previous I or P frame, as well as a backward prediction from a succeeding I or P frame. As such, the example sequence is processed by the encoder such that the first B frame is predicted from the first I frame and first P frame, the second B frame is predicted from the second and third P frames, and the third B frame is predicted from the third P frame and the first I frame of the next group of pictures. From this example, it can be seen that backward prediction requires that the future frames that are to be used for backward prediction be encoded and transmitted first, out of order. This process is summarized in Figure [7.16](http://www.cs.cf.ac.uk/Dave/Multimedia/node258.html#fig:bframe). There is no defined limit to the number of consecutive B frames that may be used in a group of pictures, and of course the optimal number is application dependent. Most broadcast quality applications however, have tended to use 2 consecutive B frames (I,B,B,P,B,B,P,Š) as the ideal trade-off between compression efficiency and video quality.

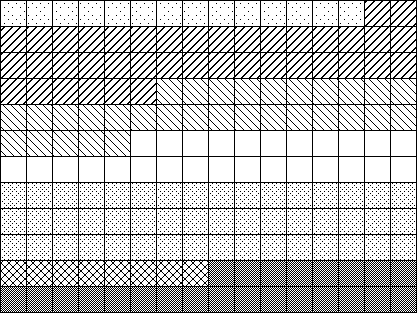
   
**B-Frame Encoding**

The main advantage of the usage of B frames is coding efficiency. In most cases, B frames will result in less bits being coded overall. Quality can also be improved in the case of moving objects that reveal hidden areas within a video sequence. Backward prediction in this case allows the encoder to make more intelligent decisions on how to encode the video within these areas. Also, since B frames are not used to predict future frames, errors generated will not be propagated further within the sequence.

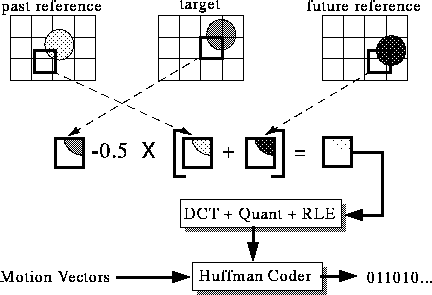
One disadvantage is that the frame reconstruction memory buffers within the encoder and decoder must be doubled in size to accommodate the 2 anchor frames. This is almost never an issue for the relatively expensive encoder, and in these days of inexpensive DRAM it has become much less of an issue for the decoder as well. Another disadvantage is that there will necessarily be a delay throughout the system as the frames are delivered out of order as was shown in Figure [[*]](http://www.cs.cf.ac.uk/Dave/Multimedia/node258.html#fig:brame). Most one-way systems can tolerate these delays, as they are more objectionable in applications such as video conferencing systems.

### Differences from H.261

* Larger gaps between I and P frames, so expand motion vector search range.
* To get better encoding, allow motion vectors to be specified to fraction of a pixel (1/2 pixels).
* Bitstream syntax must allow random access, forward/backward play, etc.
* Added notion of *slice* for synchronization after loss/corrupt data. Example: picture with 7 slices:



* B frame macroblocks can specify *two* motion vectors (one to past and one to future), indicating result is to be averaged.



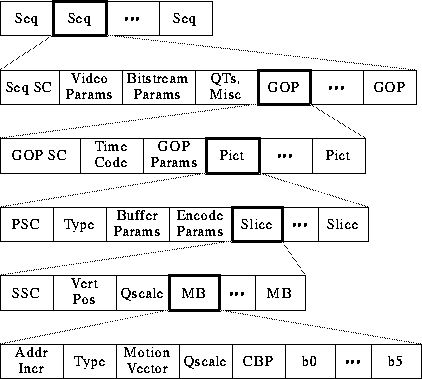
* Compression performance of MPEG 1
* ------------------------------
* Type Size Compression
* ------------------------------
* I 18 KB 7:1
* P 6 KB 20:1
* B 2.5 KB 50:1
* Avg 4.8 KB 27:1

------------------------------

## The MPEG Video Bitstream

The MPEG Video Bitstream is summarised as follows:

* Public domain tool **mpeg\_stat** and **mpeg\_bits** will analyze a bitstream.



* Sequence Information

**1.**

*Video Params* include width, height, aspect ratio of pixels, picture rate.

**2.**

*Bitstream Params* are bit rate, buffer size, and constrained parameters flag (means bitstream can be decoded by most hardware)

**3.**

Two types of QTs: one for intra-coded blocks (I-frames) and one for inter-coded blocks (P-frames).

* Group of Pictures (GOP) information

**1.**

*Time code*: bit field with SMPTE time code (hours, minutes, seconds, frame).

**2.**

*GOP Params* are bits describing structure of GOP. Is GOP *closed*? Does it have a dangling pointer *broken*?

* Picture Information

**1.**

*Type*: I, P, or B-frame?

**2.**

*Buffer Params* indicate how full decoder's buffer should be before starting decode.

**3.**

*Encode Params* indicate whether half pixel motion vectors are used.

* Slice information

**1.**

*Vert Pos*: what line does this slice start on?

**2.**

*QScale*: How is the quantization table scaled in this slice?

* Macroblock information

**1.**

*Addr Incr*: number of MBs to skip.

**2.**

*Type*: Does this MB use a motion vector? What type?

**3.**

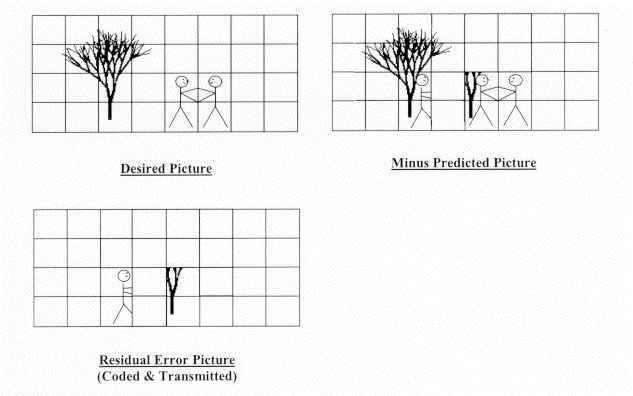
*QScale*: How is the quantization table scaled in this MB?

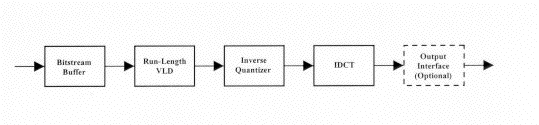
**4.**

*Coded Block Pattern (CBP)*: bitmap indicating which blocks are coded.

### Intra Frame Decoding

To decode a bitstream generated from the encoder of Figure [7.20](http://www.cs.cf.ac.uk/Dave/Multimedia/node264.html#fig:icode), it is necessary to reverse the order of the encoder processing. In this manner, an I frame decoder consists of an input bitstream buffer, a Variable Length Decoder (VLD), an inverse quantizer, an Inverse Discrete Cosine Transform (IDCT), and an output interface to the required environment (computer hard drive, video frame buffer, etc.). This decoder is shown in Figure [[*]](http://www.cs.cf.ac.uk/Dave/Multimedia/node264.html#idecode).

   
**Intra Frame Encoding**

   
**Intra Frame Decoding**

The input bitstream buffer consists of memory that operates in the inverse fashion of the buffer in the encoder. For fixed bit-rate applications, the constant rate bitstream is buffered in the memory and read out at a variable rate depending on the coding efficiency of the macroblocks and frames to be decoded.

The VLD is probably the most computationally expensive portion of the decoder because it must operate on a bit-wise basis (VLD decoders need to look at every bit, because the boundaries between variable length codes are random and non-aligned) with table look-ups performed at speeds up to the input bit-rate. This is generally the only function in the receiver that is more complex to implement than its corresponding function within the encoder, because of the extensive high-speed bit-wise processingnecessary.

The inverse quantizer block multiplies the decoded coefficients by the corresponding values of the quantization matrix and the quantization scale factor. Clipping of the resulting coefficients is performed to the region 2048 to +2047, then an IDCT mismatch control is applied to prevent long term error propagation within the sequence.

### MPEG-2, MPEG-3, and MPEG-4

* MPEG-2 target applications
* -------------------------------------------------------------------
* Level size Pixels/sec bit-rate Application
* (Mbits)
* --------------------------------------------------------------------
* Low 352 x 240 3 M 4 consumer tape equiv.
* Main 720 x 480 10 M 15 studio TV
* High 1440 1440 x 1152 47 M 60 consumer HDTV
* High 1920 x 1080 63 M 80 film production
* --------------------------------------------------------------------
* Differences from MPEG-1

**1.**

Search on fields, not just frames.

**2.**

4:2:2 and 4:4:4 macroblocks

**3.**

Frame sizes as large as 16383 x 16383

**4.**

Scalable modes: Temporal, Progressive,...

**5.**

Non-linear macroblock quantization factor

**6.**

A bunch of minor fixes (see MPEG FAQ for more details)

* MPEG-3: Originally for HDTV (1920 x 1080), got folded into MPEG-2
* MPEG-4: Originally targeted at very low bit-rate communication (4.8 to 64 kb/sec). Now addressing video processing...