

## **Video Compression MPEG: Motion Picture Expert Group**

This is the standard designed by the Motion Picture Expert Group, including the updated versions MPEG-1, MPEG-2, MPEG-4, and MPEG-7.

Most popular standards include the ISO's MPEG-1 and the ITU's H.261.

It is built upon the three basic common analogue television standards: NTSC, PAL and SECAM.

There are two sizes of SIF (Source Input Format): SIF-525 (with NTSC video) and SIF-625 (for PAL video).

The H.261 standard uses CIF (Common Intermediate Format) and QCIF (Quarter Common Intermediate Format).

Video compression is based on two types of redundancies among the video data, namely spatial redundancy and temporal redundancy.

1. Spatial redundancy means the correlation among neighbouring pixels in each frame of image. This can be dealt with by the techniques for compressing still images.
2. Temporal redundancy means the similarity among neighbouring frames, since a video frame tends to be similar to its immediate neighbours.

We have studied the theory of encoding now let us see how this is applied in practice.

We need to compress video (and audio) in practice since:

1. Uncompressed video (and audio) data are huge. In HDTV, the bit rate easily exceeds 1 Gbps. -- big problems for storage and network communications.

For example:

One of the formats defined for HDTV broadcasting within the United States is 1920 pixels horizontally by 1080 lines vertically, at 30 frames per second. If these numbers are all multiplied together, along with 8 bits for each of the three primary colors, the total data rate required would be approximately 1.5 Gb/sec. Because of the 6 MHz. channel bandwidth allocated, each channel will only support a data rate of 19.2 Mb/sec, which is further reduced to 18 Mb/sec by the fact that the channel must also support audio, transport, and ancillary data information. As can be seen, this restriction in data rate means that the original signal must be compressed by a figure of approximately 83:1. This number seems all the more impressive when it is realized that the intent is to deliver very high quality video to the end user, with as few visible artifacts as possible.

2. Lossy methods have to employed since the *compression ratio* of lossless methods (e.g., Huffman, Arithmetic, LZW) is not high enough for image and video compression, especially when distribution of pixel values is relatively flat.

The following compression types are commonly used in Video compression:

- Spatial Redundancy Removal - Intraframe coding (JPEG)
- Spatial and Temporal Redundancy Removal - Intraframe and Interframe coding (H.261, MPEG)

These are discussed in the following sections.

## **H. 261 Compression**

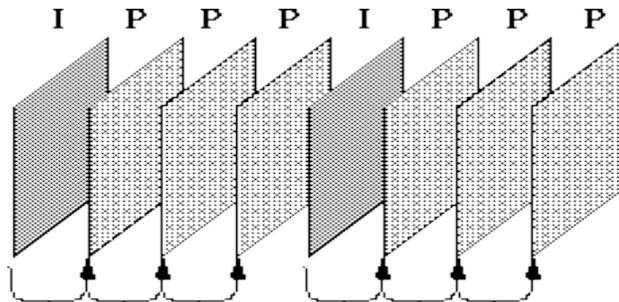
H. 261 Compression has been specifically designed for video telecommunication applications:

- Developed by CCITT in 1988-1990
- Meant for videoconferencing, video telephone applications over ISDN telephone lines.
- Baseline ISDN is 64 kbits/sec, and integral multiples ( $px64$ )

### Overview of H.261

The basic approach to H. 261 Compression is summarised as follows:

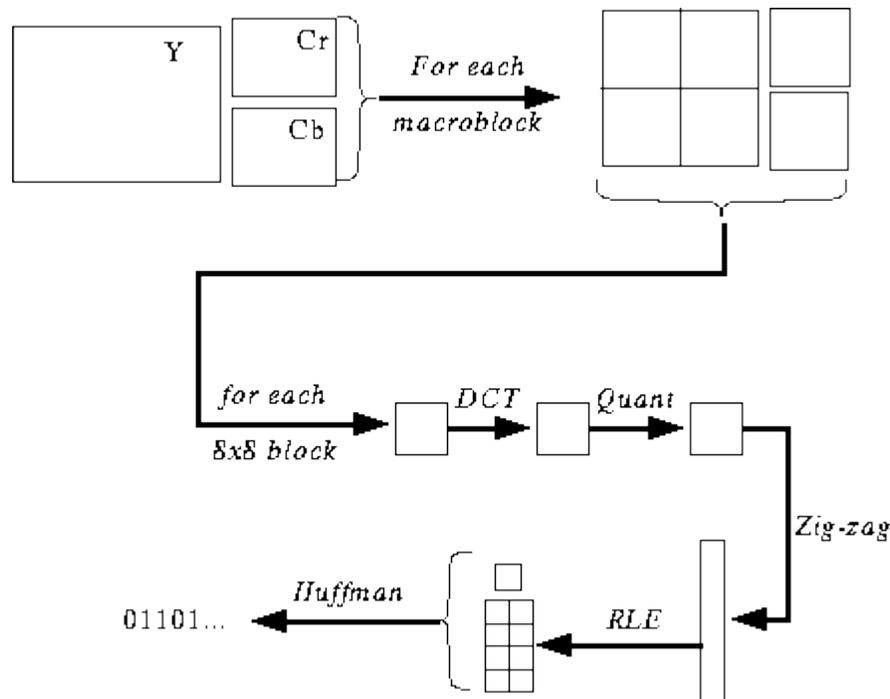
- Decoded Sequence



- Frame types are CCIR 601 CIF (352x288) and QCIF (176x144) images with 4:2:0 subsampling.
- Two frame types: Intraframes (*I-frames*) and Interframes (*P-frames*)
- I-frames use basically JPEG
- P-frames use *pseudo-differences* from previous frame (predicted), so frames depend on each other.
- I-frame provide us with an accessing point.

## Intra Frame Coding

The term *intra frame coding* refers to the fact that the various lossless and lossy compression techniques are performed relative to information that is contained only within the current frame, and not relative to any other frame in the video sequence. In other words, no temporal processing is performed outside of the current picture or frame. This mode will be described first because it is simpler, and because non-intra coding techniques are extensions to these basics. Figure below shows a block diagram of a basic video encoder for intra frames only. It turns out that this block diagram is very similar to that of a JPEG still image video encoder, with only slight implementation detail differences.



The potential ramifications of this similarity will be discussed later. The basic processing blocks shown are the video filter, discrete cosine transform, DCT coefficient quantizer, and run-length amplitude/variable length coder. These blocks

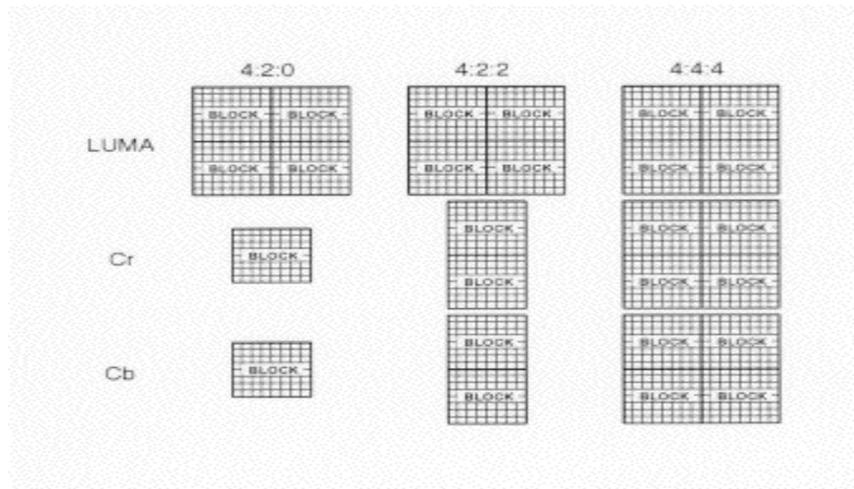
are described individually in the sections below or have already been described in JPEG Compression.

**This is a basic Intra Frame Coding Scheme is as follows:**

- Macroblocks are 16x16 pixel areas on Y plane of original image.  
A *macroblock* usually consists of 4 Y blocks, 1 Cr block, and 1 Cb block.

In the example HDTV data rate calculation shown previously, the pixels were represented as 8-bit values for each of the primary colors red, green, and blue. It turns out that while this may be good for high performance computer generated graphics, it is wasteful in most video compression applications. Research into the **Human Visual System (HVS)** has shown that the eye is most sensitive to changes in luminance, and less sensitive to variations in chrominance. Since absolute compression is the name of the game, it makes sense that MPEG should operate on a color space that can effectively take advantage of the eyes different sensitivity to luminance and chrominance information. As such, H/261 (and MPEG) uses the YCbCr color space to represent the data values instead of RGB, where Y is the luminance signal, Cb is the blue color difference signal, and Cr is the red color difference signal. A macroblock can be represented in several different manners when referring to the YCbCr color space. Figure below shows 3 formats known as 4:4:4, 4:2:2, and 4:2:0 video. 4:4:4 is full bandwidth YCbCr video, and each macroblock consists of 4 Y blocks, 4 Cb blocks, and 4 Cr blocks. Being full bandwidth, this format contains as much information as the data would if it were in the RGB color space. 4:2:2 contains half as much chrominance information as 4:4:4, and 4:2:0 contains one quarter of the chrominance information. Although MPEG-2 has provisions to handle the higher chrominance formats

for professional applications, most consumer level products will use the normal 4:2:0 mode.



### Macroblock Video Formats

Because of the efficient manner of luminance and chrominance representation, the 4:2:0 representation allows an immediate data reduction from 12 blocks/macroblock to 6 blocks/macroblock, or 2:1 compared to full bandwidth representations such as 4:4:4 or RGB. To generate this format without generating color aliases or artifacts requires that the chrominance signals be filtered.

**The Macroblock is coded as follows:**

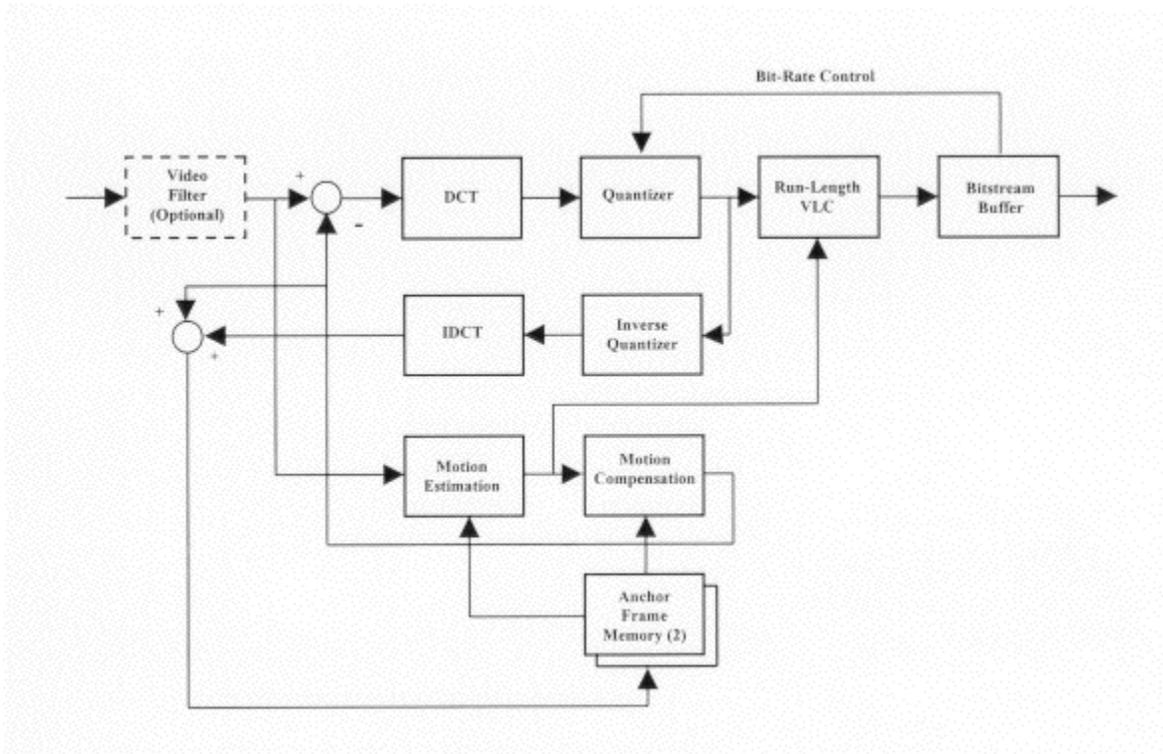
Addr	Type	Quant	Vector	CBP	b0	b1	...	b5
------	------	-------	--------	-----	----	----	-----	----

- Many macroblocks will be exact matches (or close enough). So send address of each block in image -> *Addr*
- Sometimes no good match can be found, so send INTRA block -> *Type*

- Will want to vary the quantization to fine tune compression, so send quantization value -> *Quant*
- Motion vector -> *vector*
- Some blocks in macroblock will match well, others match poorly. So send bitmask indicating which blocks are present (Coded Block Pattern, or *CBP*).
- Send the blocks (4 Y, 1 Cr, 1 Cb) as in JPEG.
- Quantization is by constant value for all DCT coefficients (i.e., no quantization table as in JPEG).

## **Inter-frame (P-frame) Coding**

The previously discussed intra frame coding techniques were limited to processing the video signal on a spatial basis, relative only to information within the current video frame. Considerably more compression efficiency can be obtained however, if the inherent temporal, or time-based redundancies, are exploited as well. Anyone who has ever taken a reel of the old-style super-8 movie film and held it up to a light can certainly remember seeing that most consecutive frames within a sequence are very similar to the frames both before and after the frame of interest. Temporal processing to exploit this redundancy uses a technique known as block-based motion compensated prediction, using motion estimation. A block diagram of the basic encoder with extensions for non-intra frame coding techniques is given in Figure below. Of course, this encoder can also support intra frame coding as a subset.

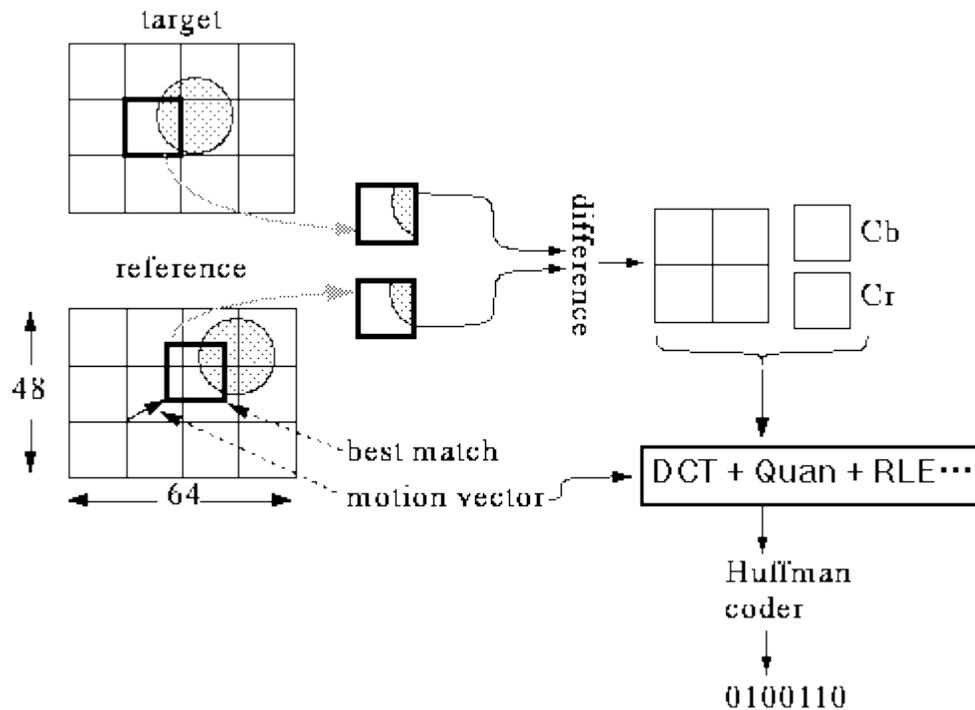


## P-Frame Coding

Starting with an intra, or I frame, the encoder can forward predict a future frame. This is commonly referred to as a P frame, and it may also be predicted from other P frames, although only in a forward time manner. As an example, consider a group of pictures that lasts for 6 frames. In this case, the frame ordering is given as I,P,P,P,P,I,P,P,P,P,

Each P frame in this sequence is predicted from the frame immediately preceding it, whether it is an I frame or a P frame. As a reminder, I frames are coded spatially with no reference to any other frame in the sequence.

P-coding can be summarised as follows:



- An Coding Example (P-frame)
- Previous image is called *reference image*.
- Image to code is called *target image*.
- Actually, the difference is encoded.
- Subtle points:

1. Need to use decoded image as reference image, *not* original. Why?

2. We're using "Mean Absolute Difference" (MAD) to decide best block. Can also use "Mean Squared Error" (MSE) =  $\sum(E \cdot E)$