

What's New in Computers?

Evolving Video Compression Standard: MPEG

Vijnan Shastri

In this article we discuss the concepts that go into the MPEG (Motion Picture Experts Group) video compression standard and also look at applications that will use this technology.

It is only fairly recently that PCs have begun to possess enough capabilities to playback good quality video – ushering in the era of digital video on the desktop and bringing the PC and TV closer to each other. These capabilities encompass not just the raw processing power but also storage and the ability to transport video data at high speeds within the PC desktop system to the display system. The power of the PC alone is not sufficient for digital video applications to attain their full potential. For that to happen, the World Wide Web must play its role, enabling new application areas such as interactive digital television. The Web is expanding at a blistering pace, both in terms of applications and reach. However, the Web would need to possess the ability to carry high volume digital video data at high speeds (referred to as bandwidth) to actually deliver video to the desktop. Hence, when one speaks of digital video, the twin issues of huge storage requirements and high speed transmission (of video data) assume over-riding importance. Unlike text, video is more than a behemoth in terms of data size requirements. Compression technology has enabled this problem to be tackled effectively albeit demanding high processing power from the CPU (for decompression and playback of video).

Problems with Raw Video Data

The standard size for an image is 640 * 480 pixels. (Refer to *Box 1* for a brief explanation of a pixel.) Since there are three



Vijnan Shastri works at the Centre for Electronics Design Technology in the Indian Institute of Science, Bangalore. His areas of interest are multimedia systems, microprocessor systems, storage subsystems and systems software.



Box 1. What is a Pixel?

A digital image on a computer screen is represented by an area of densely packed minute dots called pixels, very similar to the dots in a television screen. Each pixel in turn is built from three distinct and very minute areas: a red, green and blue area. These areas are 'lighted up' by a beam projected on to the screen. The brightness of each of these three minute areas collectively determines the colour of the pixel as perceived by the human eye. Each pixel is represented in memory by three bytes: one each for red, blue and green. There is circuitry in the display subsystem to convert byte values into signals which control the intensity of the beam. This in turn decides the brightness of each of the three basic components which finally determines the colour of the pixel. The fact that a byte is allotted to each colour implies that there can be 256 levels for each colour enabling 16 million colours in all.

bytes per pixel, one image eats up $640 \times 480 \times 3 = 900$ Kbytes of data. Playing such images in succession at a certain rate creates a motion video effect. This is exactly what is done in movies where successive pictures on film are played back to create a movie effect. We need to playback images at 30 pictures per second to get a video effect. This implies that we need to store $900 \times 30 = 27$ Mega Bytes for every second of video on disk. Coupled with this intimidating storage requirement comes the retrieval rate (from storage systems) and delivery rate (to the display monitor) requirement of 27MB/s as well. The problem of video delivery worsens if one expects video to be delivered from a network rather than the disk. This is because network bandwidths are lower by several orders of magnitude than local bus systems on which disk and display systems reside.

To address these twin problems of storage and delivery of video, digital still image compression and digital video compression technologies have been combined in the MPEG series of standards. Digital image compression deals with the problem of compressing a single image by taking advantage of redundancies in it – such as areas of similar colour. Video compression refers to the technologies used to take advantage of redundancies between successive pictures of a scene – such as the same background in successive pictures.



Digital Still Image Compression

Compression can be of two types: lossy and lossless. In lossy compression techniques image data to which the eye is relatively insensitive, is simply dropped – without causing significant degradation in image quality (perceived by the human eye). Experiments have shown that the human eye is more sensitive to variations in brightness or intensity than to variations in colour. This implies that not much is lost in an image if colour components are dropped in a limited way. Experiments have also quantified the contribution of the basic colours Red (R), Green (G) and Blue (B) to intensity, enabling the transformation of R, G and B to Intensity (Y) and two colour (or chroma) components. Thus the first step in image compression is to do this conversion and to selectively subsample (or drop) the chroma components. This subsampling is done by dropping every second chroma sample in the image. Then the image is processed using the following techniques.

In lossless compression techniques redundant data are represented by compact codes which require reduced space. Consider an example of a series of numbers such as 4444440000000003333333. This series of numbers can be represented as pairs: 64, 90, 73 where the first digit indicates the number of successive occurrences and the second digit represents the number itself. Certainly, this scheme works well only in situations where such 'runs' of numbers occur and is referred to as *Run Length Encoding* (RLE). Once this is done, a scheme known as *Huffman coding* is used. In this scheme, the most frequently occurring numbers are represented by the least number of bits – rather than representing all numbers by the same number of bits. An interesting technique is employed to make the image amenable to the above techniques.

We are all familiar with the Fourier transform which transforms a signal of any shape (wave form) in the time domain into

Experiments have shown that the human eye is more sensitive to variations in brightness or intensity than to variations in colour.



In general, the human eye is less sensitive to higher spatial frequency than to lower ones.

a sum of sine waves of different frequencies each of different amplitude, in the frequency domain. The Discrete Fourier Transform uses the same principles for sampled waveforms. Similar techniques are used (and with remarkable effectiveness) on images where the Discrete Cosine Transform (DCT) is used. However there is no time element here – remember, we are dealing with still image compression right now. The frequency here refers to the variation in intensity or colour (Chroma) values across the dimensions of the image. This is referred to as spatial frequency. In general, the human eye is less sensitive to higher spatial frequency than to lower ones. The concept of high and low spatial frequency will be clear if you look at *Figure 1*. So, analogous to time domain signals, an image can be represented as a sum of various different spatial frequencies in two dimensions. What has all this got to do with finding redundancies in an image – thus enabling better compression? This is explained in the next paragraph. Just to clarify at this point – redundancies means putting similar looking stuff together (think of RLE).

The image is first divided into areas of $8 * 8$ pixels and then the transformation of these blocks from a pixel domain representation to a frequency domain representation is done by using the DCT. The result is spectacular – the values (technically, the transformed coefficients) tend to position themselves like if

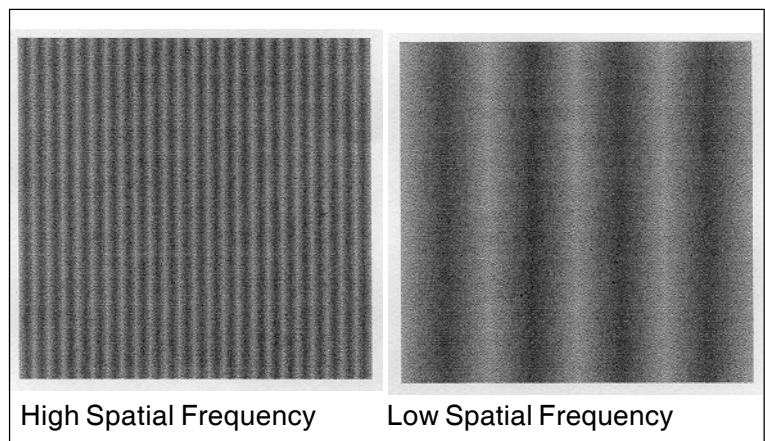


Figure 1.



you keep a sheet of paper flat on a table and lift one corner off the table. The height of each point on the paper relative to the table represents a value. The highest value (the average or 'DC' value) is in the corner and they gradually reduce as we move to the other corner. Lossy techniques are used to round-off values and they are then weighted and quantized. The weights serve to take into account the sensitivity of the human eye. Quantization refers to the fact that a single number represents a range of values. Now the values are scanned in a zigzag fashion starting at the highest point (the corner off the table) and coming down to finally reach the diagonally opposite corner, and the differences are taken. Then RLE and Huffman coding are used on this data to achieve a compact representation. All this effort reduces the image size by a factor of about 15–20. This is the same technique used in the JPEG (Joint Photographic Experts Group) standards for still images. In MPEG, pictures which are coded by this technique are known as *I pictures*, (the I standing for Intra-coded).

In MPEG, successive pictures are organized in groups called Group of Pictures or GOP.

The above technique will be clear when you think about the following analogy: suppose you were descending from the peak of a mountain (with a fairly uniform slope) in a zigzag fashion and keep measuring regularly the heights as you do so. When you reach the bottom, if you take the difference in heights, you will notice that all these differences will have very similar values i.e, they will all lie within a certain range. Since they are similar you can use RLE (explained earlier) to represent this data (the heights at various points on the mountain slope) in a compact form. The same holds good for the values obtained after the DCT process on an image. Now lets look at motion video.

Motion Video Compression

In MPEG, successive pictures are organized in groups called Group of Pictures or GOP. Each GOP contains typically 12–20 pictures. Pictures are of three types: I pictures (Intra-Coded),



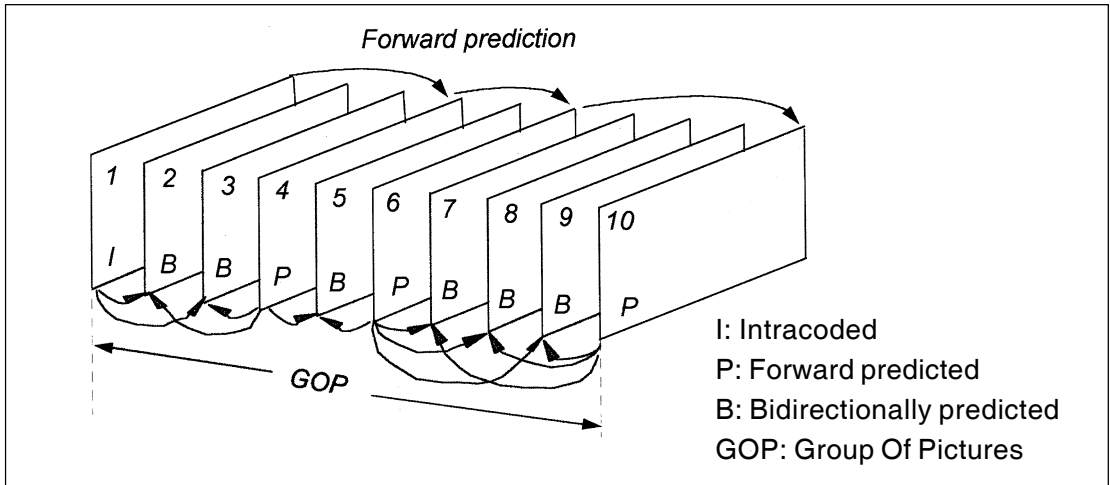


Figure 2.

P pictures (Forward Predicted Pictures) and B pictures. (Bidirectionally predicted) pictures. See *Figure 2*.

Intra coded pictures are ‘key-pictures’ because building up an I picture does not depend on any other picture. There is an I picture in every GOP. Being a key-picture, it is a crucial picture in a GOP and all other P and B pictures depend on it – directly or indirectly. So, if for some reason the data of an I picture is corrupted then the rest of the pictures in that GOP cannot be reconstructed. I pictures are built up as described in the previous section.

P pictures are built up using information from a previous I picture or a previous P picture, as shown in *Figure 2*. As we have mentioned earlier, this is possible because content of pictures tends to persist for some time (such as background or people in a picture), rather than change in every consecutive picture. The picture is built up in terms of ‘tiles’ of 16*16 pixels and these tiles are called ‘macroblocks’. For example, if there is a video shot taken with the sky and mountains as a backdrop, many areas will be similar in consecutive pictures of the video shot. Hence the encoded information in the P frame for each of the tiles essentially says “ look at the pixels of tile x in the previous picture – my pixels are the same as tile x – so just copy



them”. Technically speaking P pictures are ‘forward predicted’ from I pictures. A difference value is also stored (and added at the time of building up the picture) since exact matches cannot be found.

The central point is that smaller the magnitude of the numbers that we store, lesser the number of bits that are required to do this and hence greater the compression.

B pictures are built up from a previous I or P picture and a future P picture or I. Future picture? This sounds a bit confusing – you may ask how can I build something from a picture that has not yet arrived? Here is the explanation: the pictures are stored in such a way that the ‘future picture’ arrives in time for the build-up of the B picture but is not shown until it is time to show it. In the meanwhile it is used for building up intermediate B pictures. Thus the stored order of the pictures is different from the display order. When we say a B picture is built up from a past and future picture we mean that an average is taken between the previous and future picture. Why is this done? It turns out that this increases the efficiency of compression since the difference values get reduced even further by the averaging process.

The result of all this is that the storage needed by I, P and B pictures are of the order of 10:5:2. Hence motion video achieves a final compression factor of about 100–150 as compared to the original uncompressed images.

Here are further details: We had mentioned earlier that the image is divided into macroblocks and that the actual build-up of the P and B pictures is done in units of macroblocks. Macroblocks are classified as I macroblocks, P macroblocks, B macroblocks and skipped macroblocks. Conceptually, the I, B and P have the same meaning as in pictures. The I pictures have only I macroblocks, the P pictures have both I and P macroblocks and B pictures can have all three types of

Smaller the magnitude of the numbers that we store, lesser the number of bits that are required to do this and hence greater the compression.



MPEG has a double benefit when used for broadcasting: quality of video and number of channels.

macroblocks. Who decides what type a macroblock should be? It is the 'encoder' that does this. While doing the arduous task of compressing images, the encoder proceeds macroblock by macroblock. For a particular macroblock in the current picture it searches a limited area in the previous picture (or a future picture) for a reasonable match. If it finds a match within some tolerance it codes this (with the difference information) as a P or B macroblock; else it codes the macroblock as an I macroblock.

So that gives you a broad idea of how motion video compression is done. Encoders are usually costly pieces of complex hardware. They accept an analog video signal as input from a camera or tape, digitize the pictures, employ the above compression techniques and produce a digitized, compressed MPEG bitstream as output.

Hardware based decoders convert this bitstream back into analog video. However these hardware decoders are fast disappearing as PCs become more powerful and the entire decoding process can be done through software alone. Software encoders do exist but they take more time to do their task when compared to hardware encoders.

MPEG Standards

MPEG is in fact a series of standards. There is MPEG-1, which supports normal TV quality video, then there is MPEG-2 which supports HD-TV (High Definition TV) resolution and quality and MPEG-4 is the about to be released standard. MPEG-4 addresses the problem of transmitting video on low-bitrate channels and integrating synthetic video (produced through virtual reality tools) with camera-captured video. Low bit rate applications are used mainly for video conferencing and some web hosted applications. MPEG-2 is emerging as the de-facto broadcasting standard. MPEG has a double benefit when used for broadcasting: quality of video and number of channels. The



rapidly pervading DSS (Digital Satellite System) systems or equivalent DirectTV systems deliver flawless quality TV-video to consumers through 46cm diameter dish antennas and cheap set-top boxes. Set top boxes are devices which decode satellite signals, decompress MPEG video and deliver the TV signal to a regular TV set. The process of MPEG encoding results in an increase by a factor of six (over conventional systems) the number of channels broadcasters can transmit on their usual transmission mediums since compressed data is being transmitted.

MPEG applications are expected to get a big boost when the DVD-ROM optical disk replaces the CD-ROM in about a year from now.

MPEG Applications

Digitizing video and the existence of powerful machines to process this data gives birth to a myriad of applications – not possible in the current analog video processing and transmission scenario. Remember, many of the special effects in movies today are possible because of digital video processing.

We said in the beginning of this article that PCs (or a variant of the PC) and TVs are on a convergence path towards an appliance where we do not just watch a video (as we do on TV today) but we interact with it in many different ways.

One of the important ones is video-on-demand where the user chooses the movie and the part he or she wants to see at will – this is in contrast to normal cable TV channels where one has to see what the cable operator dishes out. Other interactive applications are education-on-demand, news-on-demand, digital libraries, hypervideo and interactive video games.

MPEG applications are expected to get a big boost when the DVD-ROM optical disk replaces the CD-ROM in about a year from now. DVD-ROM stands for Digital Versatile Disk - Read Only Memory and can store 8 times as much data (4.7 Giga Bytes) as a CD-ROM. This means that they can store a 133 minutes of MPEG-2 video – enough for many of the movies.



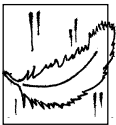
Incidentally, DVD-ROMs are the same physical size as CD-ROMs and achieve their high data storage capacity by using a thinner laser beam and thus achieving higher bit density. In the near future, DVD-ROMs of 8.5GB and 17GB will also become available.

Although the focus of this article has been video, multimedia content consists of both video and audio. Both MPEG-1 audio and MPEG-2 audio compression standards have been defined. MPEG-1 provides CD-audio quality sound. MPEG-2 video applications however, rather than using MPEG-2 audio compression standard, use a standard called Dolby AC-3 which supports six channel surround sound. A discussion on audio compression is beyond the scope of this article.

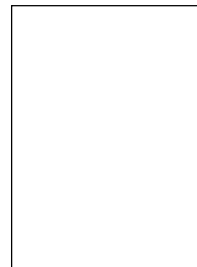
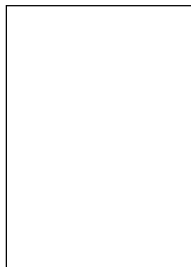
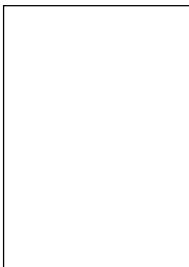
To summarize, MPEG technology enables digital video to be stored and transmitted in a cost-effective manner – promising to bring in radical new applications on the web, and perhaps, in the near future, changing the definition of what we now call television.

Address for Correspondence

Vijnan Shastri
Centre for Electronics
Design Technology
Indian Institute of Science
Bangalore 560 012, India
email:
vshastri@cedt.iisc.ernet.in
Fax:(080) 334 1683



Some of the stamps released in honour of Nobel Laureates



From the stamp collection of R G Sangoram

