AVIONIC SYSTEMS STANDARDISATION COMMITTEE

VIDEO SYSTEMS WORKING GROUP

Doc No : - ASSC/130/2/77[*]

Issue 1 - April 1997

ADVISORY DOCUMENT ON AVIONIC VIDEO COMPRESSION TECHNOLOGY[*]

0. EXECUTIVE SUMMARY

Compression techniques are discussed in the context of a model of compression which allows reversible and potentially irreversible data coding operations to be separated. Based on this model, a number of the more prominent coding techniques are outlined. Following this, the more significant established and evolving standards for video compression are discussed in the context of the compression techniques which they employ.

---

[*] **This report is published by the Avionic Systems Standardisation Committee (ASSC) to advance the role of standardisation in avionics. The use of this is entirely voluntary, and its applicability and suitability for any particular use, including any patent infringement arising therefrom, is the sole responsibility of the user.** Copies of this paper are obtainable from the ASSC Agency as an Avionic Systems Standardisation Committee publication.

Table of contents **Page no.**

Table of contents (cont) **Page no.**

Figures

Tables

1 INTRODUCTION

1.1 SCOPE

This document describes the techniques and standards of compressed digital video information. The definition of compression used here refers to the processing of a stream of digital images for the purpose of reduction in the amount of raw digital data in the signal. Whilst this definition appears straightforward, it must be borne in mind that often a video compression technique is used in combination with an image format to form its own video format eg MPEG II is an image format, but uses compression techniques to reduce the image bandwidth.

At present there are no agreed standards for avionic digital video formats. Therefore, a commercial studio quality digital standard, ITU-T Rec 601 is assumed. This recommendation specifies the techniques for digitally encoding an analogue component video system such as PAL or NTSC. This results in a raw digital data rate of some 216 Mbit/s.

Many of the techniques described in this document are based on commercial activities intended to reduce this rate to enable broadcast transmission over cable or RF links for the purposes of home entertainment. Therefore, many of these compression techniques use the features of the human visual system to reduce data rates. For example, the human visual system is unable to perceive high frequency images such as closely spaced vertical bars, and thus the high frequency portions of the image need not be transmitted in such a system.

1.2 PURPOSE

This advisory document is intended to give those involved in specifying avionic video systems an executive level understanding of the technology, terminology and standards for compressed digital video. It also intends to provide some insight into the benefits and drawbacks of compression techniques and standards which might be encountered in future avionic video systems. Since digital image compression is a relatively new area of standardisation, and commercial developments are evolving rapidly, it is not intended to draw any conclusions or select any techniques or standards at this stage.

1.3 RAW VIDEO FORMATS

Compression of digital data begins with the manner in which the raw data is represented. In the case of digital video, there are a number of factors which determine the format of the raw image. These include :

- the spatial resolution, that is number of lines and number of points per line
- the frame refresh rate
- the dynamic range and therefore the number of digital bits associated with the colour and intensity information associated with each point
- the manner in which the colour information is represented

- the relative rates at which intensity and colour are sampled in the vertical and horizontal directions.

In terms of spatial resolution, there are a number of widely used standard formats such as ITU-T 601, CIF and QCIF (see Table 1).

Format	**Spatial Resolution (Intensity Component)**	
	Number of lines	**Pixels per line**
CCIR 601	576.00	720.00
16CIF	1152	1408
4CIF[1]	576.00	704.00
CIF[1]	288.00	352.00
SIF[2]	240.00	352.00
QCIF[1]	144.00	176.00
Sub-QCIF	96.00	128.00

Notes

[1] CIF = common intermediate format.

[2] SIF = standard image format

Table 1 : Commonly used image resolutions

The dynamic range of the intensity and colour information associated with each pixel relates to the number of bits to which these components are quantised. Eight bit quantisation is relatively standard, although there are some applications which require a dynamic range of up to 12 bits.

In principal, three values are required to represent the colour and intensity of a single point, and this is inherent in RGB representations where the colour and intensity of a point is defined by its red, green and blue primary colour components. However, as indicated below, there are benefits in representing colour and intensity separately. Under such circumstances, the intensity component of a point is usually referred to as its luminance and the colour components its chrominance. While the CIE standard exists for defining colour, it is not usually adopted to represent colour in digital video applications. Instead, a colour difference scheme is frequently used, in which the chrominance of a point is represented as the difference between two primaries. This approach has the benefit of reducing the dynamic range requirements.

One of the great benefits of separating the luminance and chrominance components of an image, is that it is possible to exploit the human visual system's relative insensitivity to rapid changes in colour. This allows the chrominance components of an image to be sub-sampled without any perceptible loss of image quality. It is possible to sub-sample the colour component of an image (relative to the luminance component) by different factors in the vertical and horizontal directions. To provide some level of standardisation, the relative rates at which the chrominance and luminance components of an image are sampled in each direction are commonly specified in the form 4:x:y, where x and y are integers in the range 0 to 4. Use of the 4:x:y specification is illustrated in Table 2.

4:x:y specification	**Ratio of sampling rates for intensity and chrominance components**		**Relative storage requirements**
	vertical-direction	**horizontal-direction**	
4:4:4	1:1	1:1	100 %
4:2:2	1:1	2:1	66 %
4:2:0	2:1	2:1	50 %
4:1:1	1:1	4:1	50 %
4:1:0	4:1	4:1	37.5 %

Table 2 : Use of the 4:x:y specification for chrominance sub-sampling

2 REQUIREMENTS FOR AVIONIC DIGITAL IMAGE COMPRESSION

2.1 CLASSIFICATION

When considering compression in avionic video systems it is essential to first determine the use to which the video signal will be put. This leads to a broad distinction between those systems intended primarily for human viewing such as cockpit displays, those intended for purely automated processing such as target tracking and surveillance. Applications which involve both humans and automated processing must be consider both aspects according to their importance to the function.

It is likely that the on-platform distribution networks for digital images will have sufficient bandwidth so that compression is not needed. However, there are two specific video system areas where technology and physical limitations currently require its use - recording and off-platform transmission.

2.2 COMPRESSION FOR HUMAN VIEWING

As mentioned previously, most of the main commercial techniques are for this purpose. However, the requirements for a digital system for avionic use are much more stringent than those of a home entertainment system.

There is no information regarding the use of compression in the following applications. Any detail will be added in future revisions of this document.

Cockpit Display
Remote Control Systems
Remote Surveillance Systems

2.3 COMPRESSION FOR AUTOMATED PROCESSING

The requirements for avionic digital image compression for automated processing applications are not well understood. This comes as a result of the variety of processing tasks which might be performed. It is inappropriate to detail all the potential automated digital procession functions here, but two examples are given for illustration. In an automated processing system, image compression can be regarded as a form of simple pre-processing. The requirement will be to prevent any detrimental effect on the main processing function, whilst at the same time reducing the image bandwidth. Depending on application, these requirements may be conflicting or even complementary.

2.3.1 Target Cueing

On a ground attack aircraft there exists a requirement to process an IRST image. The processing will consist of identification of potential targets by their thermal emission. Parameters involved in identifying an object in an image as a target might include maximum temperature of the object, identification of rapid thermal gradients, and the shape of the object.

The requirements for such a function are not well understood and further research is needed.

2.3.2 CGI/Real Image Overlay

No detail of the above is currently available.

3 COMPRESSION TECHNIQUES AND THEIR APPLICATION

3.1 OVERVIEW

When considering video compression, it is useful to discriminate between algorithms and the way in which these may be applied in a lossy or lossless video compression scheme. In general the algorithms which are employed for video compression attempt to reformat the imagery such that the redundancy which exists may be exploited. Compression is achieved through appropriate encoding of the reformatted imagery as indicated in Figure 1. With a lossless compression scheme, the reformatted imagery is encoded in a reversible manner. That is, the reformatted imagery and therefore original imagery may be recovered without error from the encoded data. In contrast, in a lossy coding scheme, it is not possible to guarantee that the encoded imagery can be recovered from the encoded data.

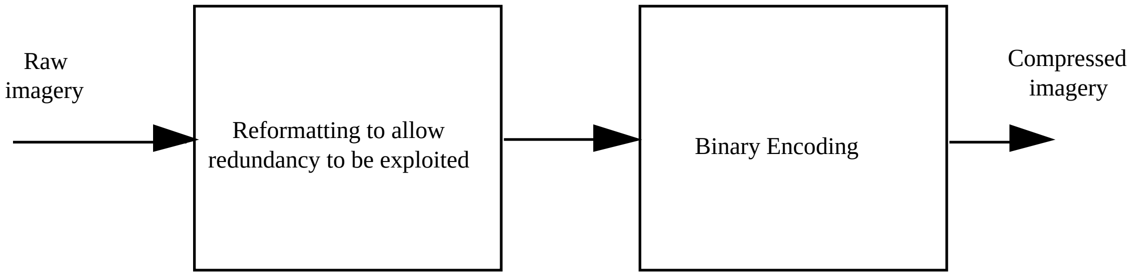

Figure 1 : Generic view of video compression

The vast majority of the practical implementations of video compression employ lossy coding. This is because lossless coding schemes are not able to guarantee levels of compression much greater than 2:1 for a wide range of images. Nevertheless the lossy implementations which are used are highly optimised in order to minimise the perceptibility of coding errors and artefacts. In developing such coding schemes, a number of measures of image quality have been employed. The mean squared difference between the raw and decoded frames is a commonly used metric, although it is generally accepted that this only provides a broad indication of image quality. More systematic measures are based on empirical subjective evaluations using a group of human viewers, for example in accordance with CCIR Recommendation 500-3 [Method for subjective assessment of the quality of television pictures]. In addition, through the European Commission funded MOSAIC research project, work is currently being undertaken into the development of evaluation techniques specifically tailored to the requirements of video compression. Although there is at present no information on non-human metrics, it should be noted that these are likely to be application specific.

Compression techniques may also be divided into two categories on the basis of the type of redundancy which they exploit in the image. Those which exploit the redundancy which exists within a frame are termed intra-frame, while those which exploit the redundancy which exists between frames are termed inter-frame.

3.2 MIXING COMPRESSED SIGNALS

It is desirable to be able to mixed compressed video signals which are derived from different sources. However, this is not a trivial matter. The following issues arise:

1. The compatibility of the original images in terms of their spatial resolution, their representation of colour and their frame rate.
2. The fact that techniques which offer the higher levels of compression such as those employed in MPEG1/MPEG2 and H.261/H.263, exploit temporal redundancy between frames and as a result of this the compressed data stream represents frames in a time iterative manner. Hence if a single frame from one source is required, there is a problem that in its encoded format, this frame is described in terms of previous frames, and therefore there is a need for some form of translation of its encoded format.
3. The fact that data from different sources may have been compressed to varying levels. For example, consider two original sources containing images of the same spatial, chromatic and temporal format. One of these may be compressed by a factor 15 the other by a factor of 30. If these two sources are mixed on a frame-by-frame basis, the quality of the resulting sequence will appear to fluctuate.

There have been a number of attempts to address the problem of mixing compressed video sources, particularly in the television broadcasting industry where these is an obvious need for such a process. The obvious approach is to decompress and then to recompress following mixing of the sources in their original format. However, as well as requiring relatively sophisticated mixing electronics, process in which coders are cascaded, that is an image is subject to multiple compression / decompression operations between its source and display, have been shown to result in unusual and highly undesirable artefacts as the two or more encoding processes interact. Since this is such an important issue to the television broadcast industry, a major European Commission funded research project is being carried out which specifically addresses this matter. This project which is called ATLANTIC and is sponsored by the 4th Framework ACTS programme, has the following scope and high level objectives.

1. It is focused exclusively on MPEG2 compressed video streams and mainly main-level/main profile (ML@MP).
2. It assumes that video will originate at a high compressed bit rate of about 15 Mbps, which is associated with a relatively low level of compression, and therefore the inherent quality of the source material will be high.
3. The scope of the work will include the develop of methodologies which operate on the compressed data stream which will allow mixing of compressed video streams from different sources, and which will allow transcoding, that is reducing the bit-rate of the compressed data stream. In addition, prototype hardware will be produced to allow the methodologies to be implemented in real-time.
4. The project started at the end of 1995 and is due to finish at the end of 1998.
5. The organisations involved are broadcasters and electronic components and system manufacturing organisations with strong interests in compressed digital video.

Assuming it is successful, and based on the rate of conversion of innovative technology into products in the television broadcast industry, this project is likely to result in commercial solutions to the problem of mixing compressed video streams in 1999. Although these will probably only be relevant to MPEG2 ML@MP.

3.3 ENCODING

3.3.1 Fault Tolerance for Encoding

The introduction of errors into a compressed video stream can potentially be a significant problem, leading to the loss of one or a number of frames. To overcome this problem, some form of error control coding is usually applied. The form that this takes is dependent upon the error characteristics of the medium in which the data is to be stored or through which it is to be transmitted. The techniques employed are well known ones such as block coding, convolutional coding and interleaving. Some video compression standards specify particular error control coding configurations.

3.3.2 Scalar Quantisation

Scalar quantisation is a process which restricts the value of a data item to one of a finite number of discrete values. In doing this it achieves compression since it limits the number of bits which must be allocated to the data item.

The process is usually designated by Q and may be described mathematically as follows:

$$x_{quant} = fix\,()$$

where:

x	is the value of the data prior to quantisation
x_{quant}	is the quantised data value
q	is a positive scalar which defines degree of quantisation; that is, larger values of q result in coarser quantisation
$fix\,()$	is a function which returns the integer part of its argument by truncating any fractional component (as opposed to rounding to the nearest integer).

The reverse process, de-quantisation, is usually designated by the term Q^{-1} and may be described mathematically by:

$$x^{\sim} \qquad = q\, x_{quant}$$

Here, $x^{\sim}$ is the decoded version of the original data value. It may or may not be equal to x, the difference between these two being the quantisation error.

Quantisation is used extensively in image compression as a means of achieving lossy data compression. It may be applied to both raw and reformatted data values.

3.3.3 Vector Quantisation

In the same way that scalar quantisation restricts the value of a single data item to one of a finite number of values, vector quantisation restricts the number of values which a vector may take to one of a finite number of values. Note that in this context, a vector is a related group of scalar values.

When applied to the compression of images, such a group may be a set of raw pixel values in close proximity, or a set of reformatted data values, although it has most commonly been used with raw data. High levels of compression can be achieved. For example, if only 16 vectors are used to present the imagery in an 8 pixel by 8 pixel region of an 8-bit monochrome image, then the level of compression achieved is given by:

$$compression = = = 128$$

However, the problem with this approach is identifying the most appropriate set of 16 vectors during the coding process. A widely accepted, computationally efficient technique for this has yet to be identified, and the lack of such a technique means that cost-effective real-time encoding is not feasible.

3.3.4 Fractal Coding

Fractal compression is a lossy encoding technique based upon the mathematics of iterated function systems, which was pioneered by Michael Barnsley in the early 1980s. When coded using fractal compression, an image is defined in terms of a set of geometric and luminance mappings, for example:

- rotate by 30 degrees
- reduce scale by 20% (some form of contraction must always be applied)
- increase luminance by 4 %.

For general images, the coding is implemented using partitioned iterated function systems in which the image is first segmented into blocks, and a set of transformations of the type described above is derived for each block. The decoding process begins with an arbitrary image such as a plain grey, and the function sequences are iterated in their respective image segments a number of times until the required image resolution is obtained. The efficiency of the techniques stems from the small amount of information required to represent the transforms (assuming they are suitably constrained).

The main challenges in fractal coding are segmenting the raw data and deriving the mappings. While there are a number of published academic algorithms on this subject, the only commercially available algorithms are embedded in executable software supplied by Iterated Systems Incorporated, the company set up by Michael Barnsley.

There are various claims about the levels of compression which can be achieved with fractal coding. Early publications presented results which suggested extremely high level of compression (up to 10,000:1). However, these were favourable images and were encoded manually. In practice automated encoding tends to be able to achieve levels of compression in the range 8:1 to 30:1 without perceptible loss of quality.

3.3.5 Statistical Coding

Statistical coding is a process by which a set of data values are encoded in a manner which reflects their statistical distribution. This may be used for lossless compression of video data, the aim being to allocate the smallest number of bits to the most frequently occurring data values. The video data has often been compressed by other methods before hand. One of the most well known statistical coding methods is Huffman coding, which optimally (in terms of data compression) allocates whole unique code words on the basis of the statistics of data values.

Huffman coding is widely used since it is easily implemented in software and electronic hardware. Another approach, arithmetic coding, gives greater levels of compression than Huffman coding, since it can effectively achieve fractional code-words. However, its implementation is more complex and therefore it is less frequently used.

3.4 ALGORITHMS

3.4.1 Overview

A large number of algorithms exist for reformatting imagery such that it is more amenable to compression. The scope of this section has been limited to the following which are the more widely adopted:

- run-length coding
- differential coding
- predictive coding
- transform coding
- subband and wavelet coding

3.4.2 Run-length Coding

Run-length coding has been widely used to compress data for facsimile transmissions for many years now. The principal is to encode sequences of data values in terms of runs at a particular intensity level. For example the data stream sequence:

1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0

could be encoded as:

1 (level), 8 (run-length)
0 (level), 3 (run-length)
1 (level), 1 (run-length)
0 (level), 7 (run-length).

The technique works well for raw data with a low dynamic range (small number of grey levels) in which typical run lengths are long. It has been successfully applied to the encoding of binary images in facsimile transmissions, and combined with other techniques in hybrid compression schemes.

When applied in isolation, run-length coding results in lossless compression. However, bit errors induced during storage or transmission can result in the corruption of line segments.

3.4.3 Differential Coding

Differential encoding, sometimes called differential pulse code modulation (DPCM), exploits the strong localised correlation that usually exists in images. It does this by calculating the difference between successive pixel values in the direction of scan, and encoding this rather than the raw pixel values. By virtue of the spatial correlation, these inter-pixel differences usually have a smaller dynamic range than the raw data values and therefore can be represented by less bits on average; the result of this coding is illustrated in Figure 2.

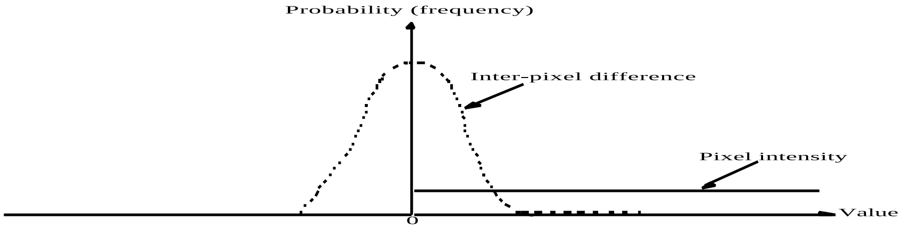

Figure 2 : **Statistical distribution of raw pixel values and inter-pixel differences**

In practical DPCM encoders, statistical coding, quantisation or a combination of these two are applied to the inter-pixel difference. Lossless implementations employ statistical coding only, while lossy implementations employ quantisation. Optimal encoders adapt the quantisation or statistical coding to suit the localised image characteristics and are usually termed ADPCM encoders.

Since recursion is employed during the decoding process, errors can propagate along a line in a similar manner to run-length coding.

3.4.4 Error propagation

Both DPCM and predictive coders are subject to error propagation since they employ process which are spatially iterative to encode the image. Hence it is usual to employ frame formatting techniques in order to provide some robustness to error propagation when these techniques are employed in practice. There are no universally agreed approaches for such frame formatting processes. However, similar problems can occur in facsimile transmissions which also employ spatially iterative coding techniques. Hence, the frame formatting techniques specified in facsimile transmission standards are probably a good starting point for someone developing a custom DPCM of predictive coder, who needs to provide robustness to error propagation.

3.4.5 Predictive Coding

Predictive coding has some similarity to differential coding in the manner in which it exploits the localised correlations which normally exist within image data. The basis of the technique is illustrated in Figure 3, which shows how a prediction of the value of a pixel is made from a weighted sum of the surrounding pixels values.

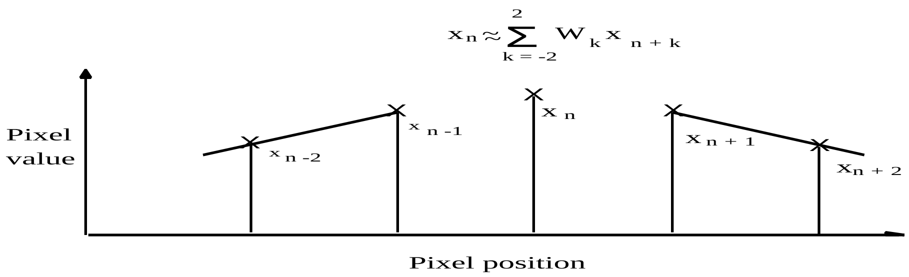

Figure 3 : Prediction based on local samples or pixels

The main difference between differential and predictive coders when applied to intra-frame coding, is that predictive coders can exploit the correlation in two orthogonal directions, while differential coders are limited to exploiting one-dimensional correlation. Hence, optimal predictive coders give higher coding gains (levels of compression) than comparable optimal differential encoders, although the implementation of a predictive coder is more complex.

As with differential coders, it is possible to have both a lossy and a lossless implementation of predictive coder. The error characteristics are similar to differential coding, although when two-dimensional prediction is used, transmission errors can introduce diagonal line artefacts which influence several lines due to the recursive nature of the decoding process.

3.4.6 Transform Coding

One of the most widely known transform techniques is the discrete Fourier transform in which image data is manipulated such that it is represented in terms of a sum of complex sinusoids. The discrete Fourier transform is one of a wider class of linear transforms which may be represented in matrix form as follows:

$$\mathbf{y} = A\,\mathbf{x}$$

where:

$\mathbf{x}$	is a vector which represents the pixel values
$\mathbf{y}$	is a vector which represents the transformed pixel values
A	is a matrix which defines the transform.

In the case of the discrete Fourier transform, the rows of matrix A are complex sinusoids, although other forms are used by alternative transforms. For example, in the discrete sine and cosine transforms, the rows are real-valued sinusoid sequences, while in the case of the Hadamard transform the rows are binary signal sequences.

For data compression applications, the transform defined by A is always reversible (assuming infinite precision arithmetic), that is, it is possible to recover the raw data **x** from the transformed data **y** as follows:

$$\mathbf{x} = A^{-1}\,\mathbf{y}$$

In order to achieve compression, it is necessary to utilise a transform which concentrates the energy (or more important bits of information) in only a few of the elements of the transform vector **y**. If this is the case, then the less important elements of **y** can be quantised relatively coarsely so reducing the number of bits required to represent that data.

For a particular set of images, it is possible to derive a transformation matrix A which gives the optimum level of compaction. This is referred to as the Karhunen-Loève transform. However, since its implementation requires the transformation matrix for the given data set to be calculated (a computationally intensive task) prior to implementing the compression, the Karhunen-Loève transform is rarely used in practice. The transforms which are commonly used for data compression are the frequency based ones such as the discrete cosine transform (DCT). Although these are not optimal, they do have good data compaction properties for natural images and they are readily implemented using cost-effective hardware.

The application of the DCT for image compression is illustrated for an image in Figure 4. Here the image is divided into non-overlapping blocks and the corresponding discrete cosine transform (DCT) of each of these is calculated. Figure 4a shows the original image and Figure 4b the magnitude of the DCT coefficients corresponding to the non-overlapping blocks. Notice how predominately the energy of the coefficients is concentrated in the top left corner of the transform. The top left hand corner of the transforms shown in Figure 4b correspond to DC (frequency of zero) and illustrates how the more significant information in visual data is concentrated in the low-frequency components. The data trails off in the higher frequency AC (non DC) coefficients.

Figure 4b is a plot of the magnitude of the DCT coefficients for corresponding non-overlapping 8x8 pixel blocks in Figure 4a. The intention of this figure is to illustrate how by using the DCT it is possible to concentrate a great deal of the significant information about a sub-image into just a few values. This is apparent from the centre of the image where most of the data in the transform blocks is concentrated at the top left-hand edge. The general variation in blocks away from the centre also illustrates that this concentrating of significant information does not always occur in a consistent manner. Note that it is necessary to show the magnitude, because with the exception of the d.c. coefficient, the values are signed.

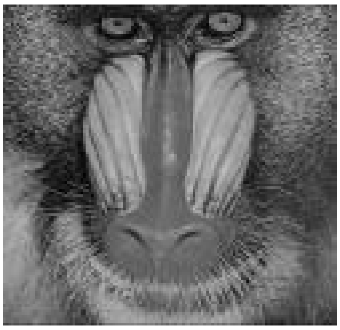

Figure 4a

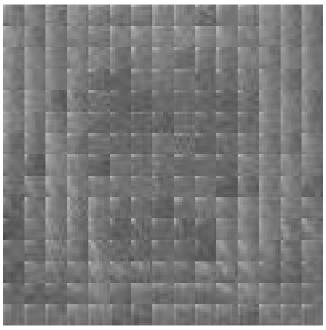

Figure 4b

Figure 4 : Discrete cosine transform (DCT) of non-overlapping sub-images (a) : original image, (b) : magnitude of DCT coefficients

The DCT has been widely adopted for intra-frame coding in image compression standards including JPEG, MPEG and H.263. Its use in these standards is indicated in Figure 5. Here the image is divided into non-overlapping 8x8 pixel blocks; each block is transformed and the transformed coefficients quantised with the level of coarseness of the quantisation increasing with spatial frequency. Typically, this results in a compression level of between 10:1 and 20:1 for each 8x8 block. Further gains in compression are made by run-length coding the AC coefficients within a block when these are scanned in a zig-zag manner which reflects the most frequent directions of correlation. In addition, the DC coefficients from successive blocks (corresponding to the average block intensity) may be differential encoded in order to give further coding gains.

Artefacts induced by transform coding broadly divide into two main categories:

- blockiness
- Gibbs phenomenon.

Blockiness relates to the inherent use of 8x8 pixel blocks. This can result from inconsistent quantisation of the AC coefficients in a texture region, in which case the texture of one decoded block will contrast with that of surrounding blocks. Blockiness can also occur when the errors in the DC coefficient become too high, in which case the overall intensity of a block can contrast significantly with its neighbours.

Gibbs phenomenon occurs where the AC coefficients in a region containing significant high frequency components are over-quantised. A typical example is ringing in the vicinity of a sharp edge. The problem can be reduced by pre-filtering the raw image. The image would be digitally low-pass filtered in order to reduce the very high frequency components around edges which lead to Gibbs phenomenon.

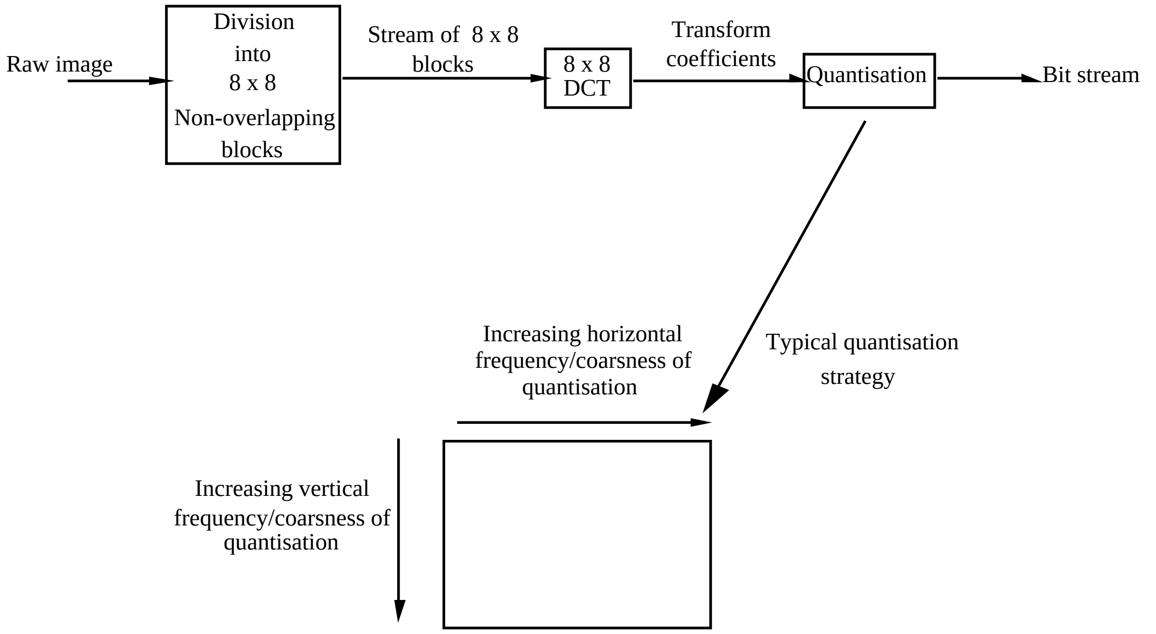

Figure 5 : Use of the DCT for image compression

3.4.7 Subband and Wavelet Coding

Subband coding in its simplest form (one-dimensional signals) is illustrated in Figure 6. Here the signal is passed through a pair of parallel filters, one with a high-pass response, the other a low-pass response. The pass band of the low-pass filter extends from d.c. to one quarter of the sampling rate, while that of the high-pass filter extends from one quarter of the sampling rate to the Nyquist rate. Thus, the filters divide the signal into two sequences each with a bandwidth of , and so it is possible to reduce the sampling rate by a factor of two (as indicated by the downwards pointing arrow) without loss of information. The two resulting signals are referred to as subbands.

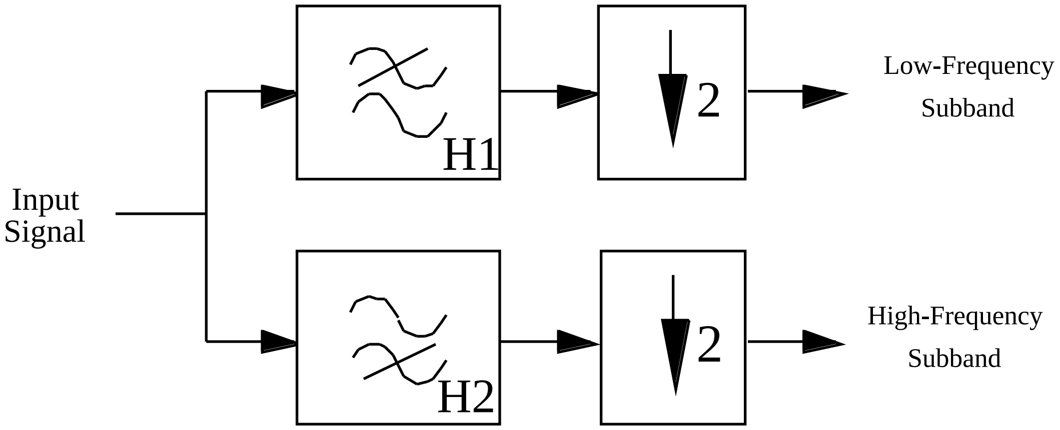

Figure 6 : **Dividing a signal into two subbands**

Figure 7 shows how the original signal can be recovered from the two subbands produced using the filtering structure shown in Figure 6. Here the upwards pointing arrows depict a doubling of the sampling rate which is achieved by inserting zeros between successive values of the subband signal. The two filters interpolate the signals in order to reproduce the subband components at the original sampling rates. Finally, the two reconstructed subbands are added together in order to produce the original signal.

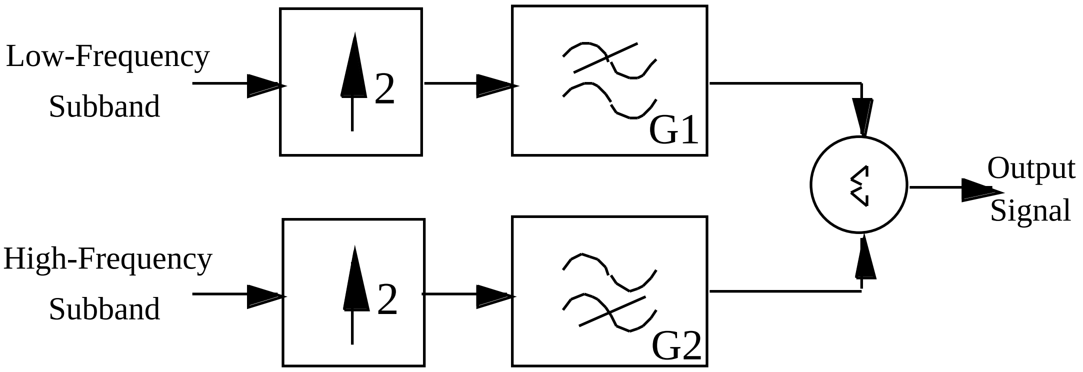

Figure 7 : **Reconstructing a signal from two subbands**

The key parameters of the structures shown in Figures 6 and 7 are the coefficients associated with the filters H1, H2, G1 and G2. Under realistic conditions the filters H1 and H2 do not have "brick-wall" responses and so they have a finite level of response to components in their respective stop bands. In the case of the low-pass filter H1, it is clear that the decimation operation (halving the sampling frequency) will result in aliasing of the high frequency components which pass through the filter. A similar process arises for low frequency components in the subband associated with the high-pass filter. However, it is possible to select the filters H1, H2, G1 and G2 such that these alias components are cancelled when the signal is reconstructed. Under such circumstance the overall system is said to provide perfect reconstruction. It is possible to produce multiple subbands either by splitting the original signal into more than two bands of equal width, or by repeating the subband decomposition process for the low frequency subband in order to achieve a logarithmic spacing of the subbands; both of these are illustrated in Figure 8.

The application of a simple two channel subband filter to an image is illustrated in Figure 9. Here the process is applied successively in the vertical and horizontal directions and this results in a total of four images, each of the size of the original image. The image which results from applying the low-pass filter in both the horizontal and vertical directions is a reduced resolution version of the original image, while the remaining images represent the information which must be added to this in order to give the original full-resolution image.

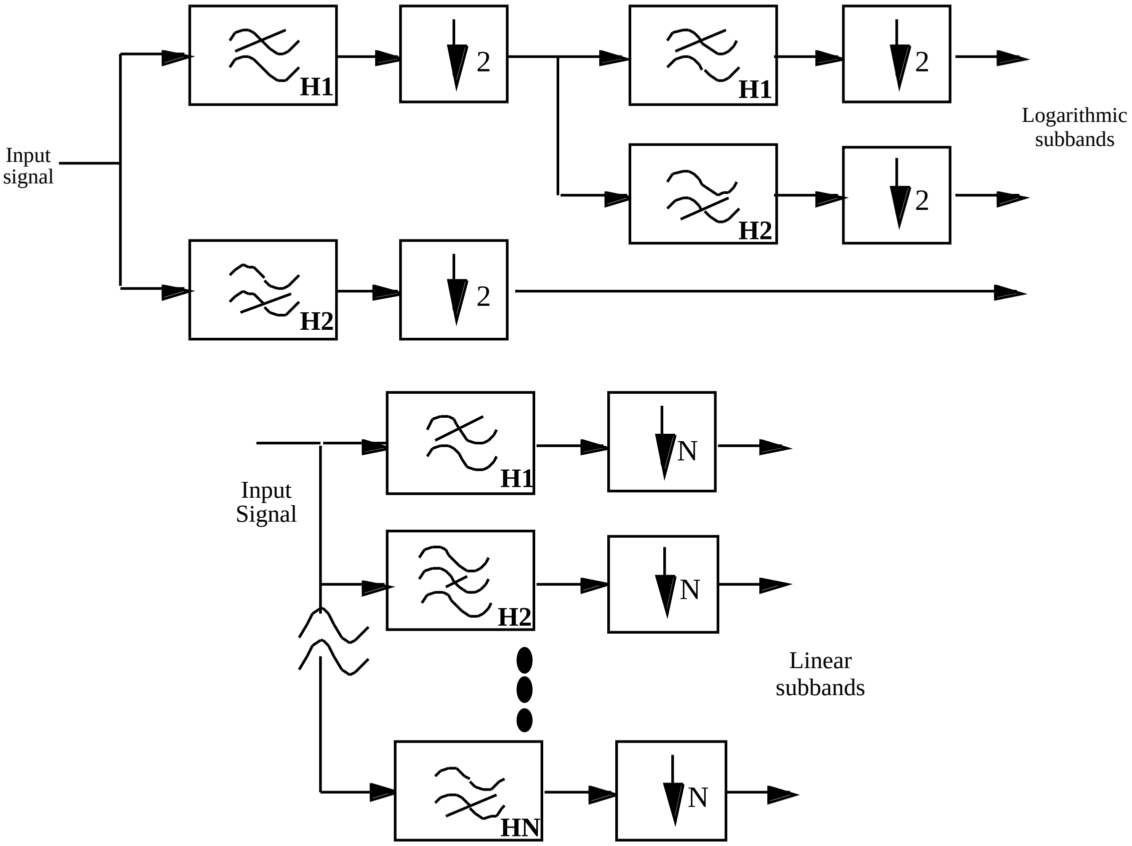

Figure 8 : **Multiple subbands**

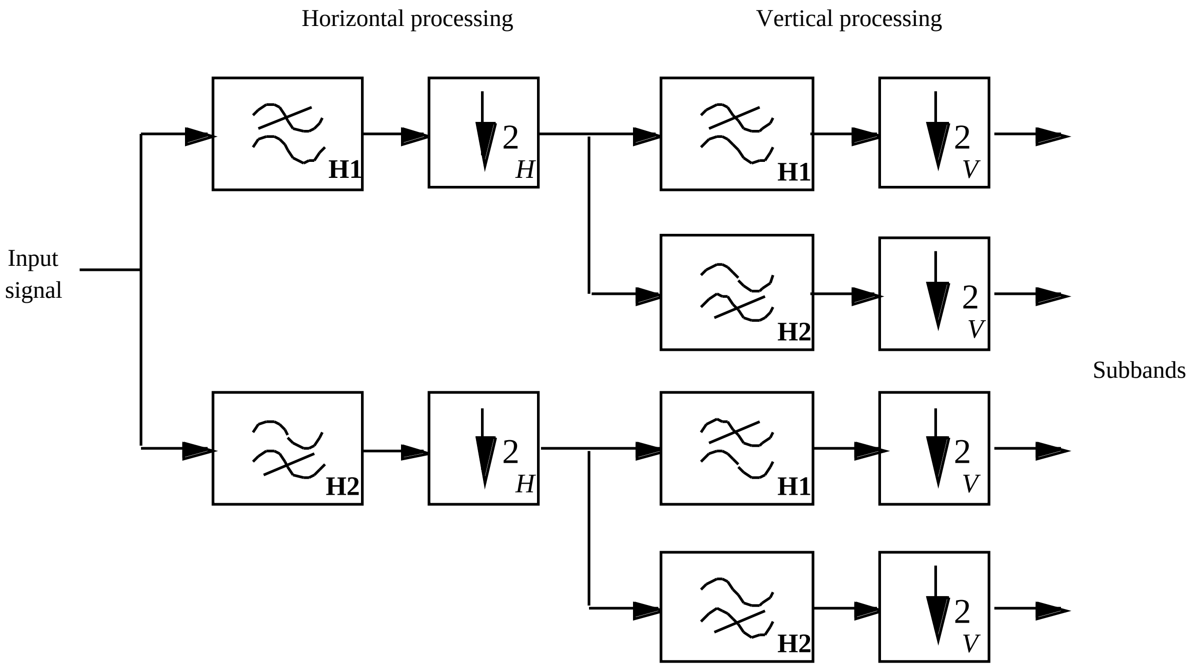

Figure 9 : **Image encoding with subband filters**

Subband coding is employed in video compression in order to divide the raw image into bands which have different levels of subjective significance. It is then possible to achieve compression by applying more lossy (that is error inducing) coding in the less significant bands. A further potential future use for subband coding is to provide compatibility between different levels of definition in image compression applications. For example, a subband filter can be applied to a high definition video signal in order to produce a standard definition signal (the lowest subband) along with the additional information required to reproduce the high definition image.

Subband filtering has considerable similarity to wavelet filtering. The main difference between the two is the basis on which the filters are derived. In the case of subband filtering, the filters are derived on the basis of the removal of aliasing (frequency domain analysis), while maximum compaction is the basis on which wavelet filters are designed.

The errors which may arise when subband coding is applied in a non-reversible manner are similar to the aliasing artefacts which can occur when using the DCT of 8x8 blocks. That is ringing at edges, ghosting, the occurrence of banding in areas with a highly structured texture and so on. However, unlike the DCT in which these are localised, it is possible for such effects to extend for a whole frame in the case of subband coding. These problems can be avoided by careful planing of the quantisation strategy for the different subbands.

3.4.8 Motion Estimation / Compensation

A relatively high proportion of video sequences contain only a limited amount of motion. For example the head of a newscaster or the balls on a snooker table. In such sequences very little changes between subsequent frames and therefore there is a considerable level of redundancy.

A simple way to exploit the temporal redundancy present in image sequences, is to calculate the difference between successive frames and to store or transmit this rather than the original frames. However, this does not work well for scenes in which there is some form of simple global motion, for example when a camera zooms. The approach also falls down when there is complex motion in a large area, for example a rotating disk. Hence, a more general approach is to assume that each point is moving within the plane of the frame. For this purpose, it is possible to associate a vector with each point as shown in Figure 10. Here the direction of the vector is equal to the direction of motion of the point, while the magnitude of the vector is proportional to the speed at which the point is moving. This view of the image as an array of moving points is commonly referred to as a motion field, and the majority of the interframe compression techniques applied to images attempt to estimate this motion field and compensate for it such that it is possible to predict future frames.

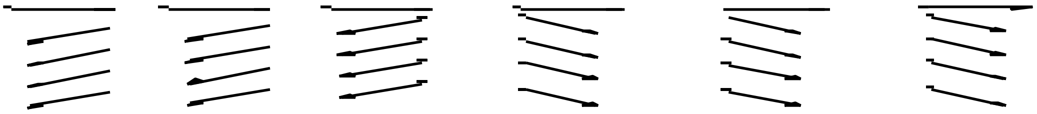

Figure 10 : **Vectors representing the motion of points - a motion field**

The manner in which motion estimation/compensation is applied to image compression is illustrated in Figure 11. Here the encoder estimates the motion field within the current frame and uses this to predict the next frame. It then calculates an error image which is equal to the difference between its prediction of and the actual value of the next frame. Finally the error image is encoded using an intraframe coding technique and the result sent along with the estimate of the motion field to the decoder. At the decoder, an estimate of the next frame is reproduced from the current frame and the motion vectors. Then the decoded error image is added to this estimate to reproduce the actual value of the next frame (within these limits of the motion estimation and intraframe coding processes used).

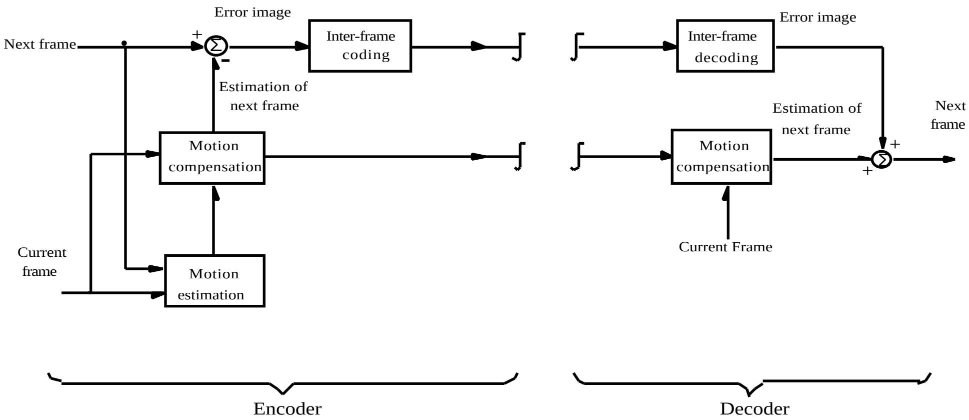

Figure 11 : **Use of motion estimation / compensation for video compression**

Potentially, estimating a full motion field of the form indicated in Figure 10 is a highly computationally intensive task. In addition, the amount of information associated with such a dense motion field will almost certainly exceed that of the original image frame.

Therefore, practical video compression schemes use a cruder estimate of the motion field. The most commonly applied technique, which is incorporated in standards such as MPEG and H.261/H.263, is block based motion estimation. This technique divides the image up into non-overlapping blocks and estimates the average displacement of the block between successive frames. While this approach to motion estimation suffers from that fact that it can produce a rather unnatural blocky estimate of the next frame, it does have the benefit of a relatively low-complexity hardware implementation which is why it is so popular in current video compression standards.

Motion estimation/compensation is generally accepted as one of the main weak areas in current video compression standards. Therefore, it is currently the subject of considerable academic research, with a wide range of sophisticated approaches being explored including:

- temporal / spatial segmentation
- multi-resolution motion estimation
- the use of more complex motion models such as warping
- the use of interpolation in the encoder and decoder to derive a dense motion field from a set of coarsely spaced estimates.

The incentives for developing these relatively sophisticated techniques are to increase the visual quality of decoded images achieved at lower bit rates, and to allow existing levels of quality to be achieved at higher compression rates.

4 COMPRESSION STANDARDS

4.1 OVERVIEW

Open video compression standards primarily define the syntactical format of the binary data stream which results when the standard is applied to compress image data. In doing this they may indirectly define or constrain:

- the algorithms or encoding techniques employed to compress the data
- the bandwidth of the compressed data steam
- the format of the raw imagery.

Various standards which exist for the compression of images have been developed for specific applications. The broadest distinction is between those developed for the compression of still images and those developed for the compression of video sequences. Standards for video sequences can be further divided into particular applications, for example video telephony, medium resolution video for multi-media applications and broadcast quality video.

Typical levels of compression achieved by standards for still and moving imagery are as follows:

still images :	2:1 .. 30:1

video sequences : 16:1 .. 40:1

Some sources quote higher levels of compression than the figures shown. However, the figures shown are the those which the standards have been designed for. It is possible to apply the standards and achieved higher levels of compression, but it is generally accepted that the quality of the resulting decoded video sequences are too low for their intended use. In addition, it is possible in some cases to apply the standards and achieve compression which can be guaranteed to be lossless, although the resulting level of compression is very low (of the order of 2:1).

As indicated above, standards for the compression of video sequences generally give higher levels of compression than those for stills. This is because they can employ techniques which exploit temporal redundancy as described in Section 3. However, these higher levels of compression cannot be achieved if the correlation between successive frames is low. This might be the case in a fast moving scene, or in a scene with a moderate level of motion where the frame update rate has been reduced in an attempt to further increase the level of compression.

The key standards which have been widely adopted for commercial applications include:

- JPEG
- H.261 / H.263
- MPEG / MPEG-2.

All of these standards employ an intra-frame coding techniques which is based on the DCT of non-overlapping 8 by 8 pixel blocks as described in Section 3. In addition, with the exception of JPEG which only applies to still images, block-based motion estimation/compensation as described in Section 3 is used to exploit temporal redundancy.

Spatially and temporally hierarchical encoding is used in most standards. For example the hierarchy in MPEG-1 is as follows:

highest level :	group of pictures - sequence of frames encoded using a mixture of intra and inter-frame compression techniques
:	frame
:	macro block - four 8x8 luminance blocks and two 8x8 chrominance blocks
:	8x8 block of pixels
lowest level :	pixel.

4.2 COMPRESSION OF STILL PICTURES

The activities of the Joint Photographic Experts Group (JPEG) are defined in international standard "ISO/IEC 10918 (JPEG) Information technology - Digital compression and coding of continuous-tone images" which has four parts:

Part 1 - Requirements and guidelines
Part 2 - Conformance testing
Part 3 - Extensions.
Part 4 - Registration procedures for JPEG profile

This standard defines the form of compressed data streams for still images. Up to 255 colour components per frame are allowed, with the encoding process being applied independently to each colour component; this allows for example, encoding of satellite imagery at many frequencies from infrared to ultraviolet. In addition, a dynamic range of up to 11 bits can be accommodated for each colour component.

The standard allows four distinct operating modes:

- sequential
- progressive
- lossless
- hierarchical.

Implementation of baseline JPEG is designed to ensure portability. To the JPEG data stream certain marker information is added as an interchange format header to set up essential values at the decoder. The addition of Part 3 of the standard defines a new file format (Still Picture Interchange File Format, SPIFF) allows considerable flexibility in defining free format text and other data relating to an image.

JPEG can be applied to video sequences in which case intra-frame coding is employed for each frame in the sequence.

A technical guide to JPEG has been published and is available from BSI as PD0006.

4.3 COMPRESSION OF MOVING PICTURES

4.3.1 MPEG-1

The Motion Picture Experts Group (MPEG) defined MPEG-1 (frequently referred to as MPEG) primarily for the transfer of data at a rate of approximately 1.5 megabits per second. This allowed video to be replayed in real-time from CD-ROMs and was therefore relevant to multi-media applications.

An important aspect of the standard is that it allows a Constrained Parameters mode of operation which is intended to promote compatibility. A flag in the bit stream signifies conformance with this mode, which implies the following constraints:

Horizontal picture size :	up to 768 pixels
Vertical picture size :	up to 576 lines
Picture area :	up to 396 macro blocks
Pixel rate :	up to 30 Hz

Bit rate : up to 1.856 megabits per second.

The standard assumes that the raw images are non-interlaced with colour being 4:2:0 encoded.

MPEG-1 is formally defined in the international standard "ISO/IEC 11172 - Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s". This has five parts:

Part 1 - Systems
Part 2 - Video
Part 3 - Audio
Part 4 - Compliance testing
Part 5 - Technical report on software simulation for MPEG-1

Part 1 specifies the system coding layer of the standard. It defines a multiplexed structure for combining audio and video data and the means of representing the timing information needed to replay synchronised sequences in real-time. Parts 2 and 3 specify the coded representation of video and audio respectively, as well as the decoding process required to reconstruct pictures and audio. Part 4 specifies procedures to determine the characteristics of coded bit-streams and to test compliance of bit-streams and decoders with the requirements specified in parts 1, 2 and 3.

4.3.2 MPEG-2

MPEG-2 is a second generation of the MPEG-1 standard. This includes a large number of enhancements and extensions, the key ones being as follows:

- the ability to accommodate interlaced images
- a variety of raw images formats for a number of diverse applications ranging from low-quality SIF through to broadcast quality high definition television
- the facility for 4:2:2 and 4:4:4 colour encoding
- scaleable modes (scaleable spatial resolution and scaleable picture quality)
- user definable DCT quantisation matrices and alternatives to the standard zig-zag scan
- non-linear DCT quantisation allowing a wider dynamic range than in MPEG 1
- enhanced motion prediction including half-pixel accuracy in motion vectors.

MPEG-2 is formally defined in the international standard "ISO/IEC 13818 Information technology - Generic coding of moving pictures and associated audio", will have nine parts:

Part 1 : Systems
Part 2 : Video

Part 3 : Audio
Part 4 : Conformance testing
Part 5 : Software simulation
Part 6 : Digital storage and medium command and control (DSMCC)
Part 7 : Audio extension for non-backwards compatible coding
Part 8 : withdrawn
Part 9 : Extension for real-time interface for system decoders
Part 10 : Conformance extension for DSMCC

To ensure compatibility between MPEG-2 encoders and decoders designed for different types of application, the standard defines a number of optional levels and profiles. These are analogous to MPEG-1's Constrained Parameters, in that they limit the format of the images, the bit-rate and the coding techniques associated with conformant bit-streams.

The standard incorporates an SNR scaleable mode which divides the bit steam into a lower quality layer (by higher quantisation) and enhancement layers (the lower quantised differences). This allows higher amounts of error protection to be added to the lower, most important layer, and less to the layers contributing fine detail.

Overall, the standard offers many options and therefore considerable scope for optimisation in order to meet the requirements of a particular application. However, it is also very complex and in-depth understanding of what it has to offer requires investment in time.

4.3.3 H.261/H.263

H.261 has primarily been defined for video telephony applications employing communications links which have a capacity that is a multiple of 64 kilobits per second (nominally an ISDN line). H.263 is a recent revision of the H.261 standard which includes a number of enhancements aimed at improving fidelity and levels of compression, and allowing the transfer of image sequences via communications links with a capacity less than 64 kilo bits per second. The standards themselves are defined in the following documents:

H.261	ITU-T Recommendation H.261, Video codec for audio-visual services at px64 kbit/s
H.263	ITU-T Recommendation H.263, Video coding for narrow telecommunication channels at < 64 kbit/s.

Raw images are constrained to have resolutions which are the various standard multiples and divisors of CIF, while colour must be 4:2:0 encoded.

4.3.4 MPEG-4 : The Next Generation Compression Standard

Current video compression standards have resulted from many man-years of effort and have wide ranging practical application. However, there are still areas in which they have significant weaknesses. In particular when applied to very low bit-rate transmission the quality of the decoded imagery falls short of most peoples expectations. In addition, a number of the artefacts which arise at all transmission rates are either undesirable or unacceptable for specific applications. Current work in the field of video compression is attempting to address these issues, and a large proportion of this work is targeted at MPEG-4 which will be the next major standard issued by ISO/IEC JTC 1/SC 29 - Coded representation of picture, audio, multimedia, and hypermedia information.

Currently it is intended to produce the MPEG-4 standard in 1998. This will cover a wide range of low to medium bandwidth applications requiring video transmission. Advanced coding techniques will be used and a number of candidates are currently being considered. In the main these provide alternatives to the current intra-frame coding which is based on the DCT of 8x8 pixel blocks, and the current block based motion estimation/compensation techniques. It is envisaged that much of the work on the development of the standard will focus on system issues allowing considerable flexibility in terms of the coding techniques used. For example, transmission of the decoding algorithm in the bit-stream is currently a viable option. One of the main objectives of the new standard is to allow more functionality for use in multi-media systems, such as the ability to decode different objects (background, foreground) in an image separately. Error robustness is now a functionality being considered inside the standard.

5 COMPRESSION IMPLEMENTATION TECHNOLOGY

Depending upon the real-time requirements digital video compression techniques can be implemented either in software, hardware or a combination of these two.

A number of software implementations of image and video coding standards are available. These take the form of encoder and decoder pairs. For example, source code for public domain encoder / decoder simulators are available on the internet as follows:

H.263 - Telenor Research
    (FTP site “ftp://bonde.nta.no/pub/tmn/software/tmn-1.6.tar.gz”)

JPEG - The independent JPEG group
    (FTP site “ftp.uv.net:/graphics/jpeg/jpegsrc.usb.tar.jz”)

MPEG1/2 -MPEG software simulation group
    (Web site “http:/www.mpeg.org/index.html/MSSG/”

These simulators are intended for research purposes only and have not been optimised for real-time implementation. In fact, constraints apply to their use in commercial products. Commercial software products are also available for the implementation of image and video coding standards. These include packages optimised for real-time implementation which take the form of libraries, as well as integrated packages when simplify the use of the encoder or decoder.

5.1 SOFTWARE

Software is also available from a variety of sources for the implementation of non-standard coding techniques. For example Iterated systems have a number of products for fractal image coding. In addition, a number of academic institutions and small commercial companies have produced optimised encoders and corresponding decoders based on non-standard coding techniques. These can offer better coding rates that is available from video coding standards. The commercial conditions associated with access to and use of such coders from academic institutes varies considerably. At one extreme the encoder and decoder software may be freely available in source code format with the only conditions on its use being to acknowledge the originator, while at the other extreme, access to the software is subject to negotiation of a contract which varies on a case-by-case basis.

5.2 HARDWARE

In terms of hardware, VLSI implementations of components of standardised video encoders or decoders have been available for a number of years. For example, devices to implement 8x8 DCTs at pixel rates corresponding to full frame rate video. More recently, a number of devices capable of implementing a complete coder have also appeared. For example Quantel produce a fully integrated JPEG encoder/decoder and IBM a fully integrated MPEG2 encoder/decoder. There are a number of board-level products available which allow encoders and/or decoders for standard video compression techniques to be implemented in real-time. These include PC plug-in cards and stand-alone embedded configurations.

Texas Instruments released their TMS320C80 video processor last year. This device has five separate closely coupled programmable processors, and is capable of implementing encoders and decoders for a number of standard video compression techniques at full frame rates.

It should be noted that the bandwidth requirements vary between JPEG and MPEG standard based devices. For example, for CCIR 601 format image resolution, a PCI based JPEG device requires 20-25 Mbit/s bandwidth for broadcast quality transmissions with no visible artefacts. In contrast, MPEG 1 and 2 will only require 4-6 Mbit/s bandwidth for the same quality transmission. A range of board products are now available to implement video compression at full frame rates.

6 REFERENCES

PD0006	The Technical Guide to JPEG, digital compression of photographic images
ITU-R Rec. 601-4	Encoding parameters of digital television for studios
ISO/IEC 10918 (JPEG-1)	Information technology - Digital compression and coding of continuous-tone images including Part 1 - Requirements and Guidelines; Part 2 - Conformance testing; Part 3 - Extensions.

ISO/IEC 11172 (MPEG-1)	Information technology - Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s including Part 1 - Systems; Part 2 - Video; Part 3 - Audio; Part 4 - Conformance testing (MPEG-1); Part 5 - Technical report on software simulation for MPEG-1.
ISO/IEC 13818 (MPEG-2)	Generic coding of moving pictures and associated audio including : Part 1 - Systems (1995); Part 2- Video; Part 3 - Audio; Part 4 - Compliance testing; Part 5 - Software simulation; Part 6 - Extension for DSM-Command and control; Part 7 - Extension for non backward compatible audio coding; Part 8 - Extension for 10-bit video coding; Part 9 - Extension for real-time interface for system decoders; Part 10 - Conformance extensions for DSM-CC.