THE QUEEN'S AWARDS
FOR ENTERPRISE:
INNOVATION
2011

BBC

Research & Development

White Paper

WHP 238

August 2013

The VC-2 Low Delay Video Codec

Tim Borer

BRITISH BROADCASTING CORPORATION

White Paper WHP 238

The VC-2 Low Delay Video Codec

Tim Borer

Abstract

Much research has been focused on the development of codecs for distribution to end users via broadcasts, optical disks, the Internet and mobiles. But codecs are also required during the production of television and movies. This paper describes the VC-2 Low Delay codec (VC-2 LD), developed in BBC R&D and standardised by the Society of Motion Picture and Television Engineers (SMPTE). It describes the design requirements for a production, or “intermediate” codec. It provides an overview of the algorithms, and the innovative features, that are used in VC-2 LD. And it discusses the system features that make VC-2 LD useful in a variety of applications. Experimental results are provided for the performance of VC-2 LD. These results demonstrate that, in addition to its other advantages (low complexity and ultra low latency) at medium and high bit rates VC 2 LD provides performance comparable to or better than well known codes such as Apple ProRes or DNxHD.

Additional key words: discrete wavelet transform, lifting, integer transform, edge extension, quantisation matrix, interleaved exp-Golomb, texture masking, rate control, slices, early termination, ultra low latency, low complexity, concatenated coding, royalty free, Dirac Pro.

White Paper WHP 238

The VC-2 Low Delay Video Codec

Tim Borer

Contents

1 Introduction ... 2
2 Design Principles ... 4
3 Codec Algorithms ... 5
3.1 Wavelet Transform ... 5
3.2 Lifting ... 8
3.3 Integer Arithmetic ... 10
3.4 Arithmetic Precision ... 10
3.5 Boundary Handling (or Edge Extension) ... 11
3.6 Processing Order (of the Wavelet Transform) ... 12
3.7 DC Prediction ... 12
3.8 Slices ... 12
3.9 Quantisation Matrix ... 13
3.10 Quantisation ... 15
3.11 Quantisation using the Quantisation Matrix ... 16
3.12 Entropy Coding (Variable Length Coding) ... 17
3.13 Texture Masking & Rate Control ... 18
3.14 Coded Slice Structure & “Early Termination” ... 19
3.15 Codec Algorithm Summary ... 20
4 System Features ... 21
4.1 High Quality ... 21
4.2 Ultra Low Latency ... 21
4.3 Low Complexity ... 21
4.4 Low Concatenated Coding Loss ... 21
4.5 Royalty Free Open technology ... 22
4.6 Video Format Support ... 22
4.7 Algorithmic Flexibility (Wavelet Features) ... 23
5 Experimental Proceedure ... 24
6 Results & Discussion ... 25
7 Summary & Conclusions ... 31
8 References ... 32

White Paper WHP 238

The VC-2 Low Delay Video Codec

Tim Borer

1 Introduction

Video coding (a.k.a. compression) is ubiquitous in today's media industry. Without it video distribution to the end user, via digital broadcast, optical disks, the internet and mobiles, would not be possible. Much R&D effort has been devoted to "emission", or "direct to home", codecs such as the well known MPEG and ITU H26n series of codecs [1]. These codecs offer high compression, but at the expense of high complexity, and high latency (often many seconds for a broadcast encoder).

On the production side of the industry the advantages of compression, reduced storage, reduced bandwidth, lower specification equipment and so on, are just as alluring. But video production has different requirements of a video codec such as higher quality, low latency, and low complexity. The degree of compression is also less critical; 8:1 compression, or less, may suffice. For archive applications the quality requirement is higher still and very low compression ratios or lossless compression are appropriate.

Relatively little research and development effort has been devoted to video codecs intended for production and archive use. Consequently few standard codecs are designed specifically for these applications. Instead video production and archiving has, typically, used specialist profiles of emission codecs (MPEG-2, H264 & JPEG2000), even when they are ill suited to these applications. Such codecs, not designed for video production but nevertheless pressed into service for this application, include MPEG-2 (as SMPTE 356 a.k.a. IMX), MPEG-4 Part 2 (Simple Studio Profile used in the HDCAM SR tape format) and H264 (as SMPTE RP 2027 a.k.a. AVC-I).

VC-2 LD profile, the subject of this paper, is an intra-frame wavelet codec. It is targeted at high quality, low compression ratios for video production, and provides a counterpoint to the emission codecs, such as H264, used at the other end of the content chain. It has been standardised through the Society of Motion Picture and Television Engineers, the same process as Microsoft's VC-1 emission codec (a.k.a. Windows Media Video). Alternative production codecs include DNxHD, standardised as VC-3 by the SMPTE, and proprietary codecs such as Apple ProRes [2][3], Cineform (currently in the process of standardisation as SMPTE VC-5) and Grass Valley's HQX [4]. These codecs are particularly aimed at coding raw footage prior to colour grading or editing, and are sometimes known as "intermediate codecs". JPEG2000 [12] also enjoys considerable use for production and archiving but is restricted by its complexity.

This document describes the "Low Delay" profile of the VC-2 SMPTE 2042 coding standard (VC-2 LD) and provides an overview of its algorithms and performance. The Low Delay profile was specifically designed for use as video production codec rather than for end user distribution. "Main" and "Simple" profiles are also defined, but not considered in detail here. The codec was designed to be as flexible as possible to support a wide range of applications and provides unrestricted image size (large and small), unrestricted bit depth, 4:4:4, 4:2:2 & 4:2:0 colour subsampling. It also supports large choice of coding parameters to trade off compression efficiency, complexity and latency for a given application.

The VC-2 codec standard, published in 2009 by the Society of Motion Picture and Television Engineers (SMPTE) as SMPTE 2042, has three parts [5][6][7]. Part 1 is the core codec specification, part 2 defines levels and part 3 is a conformance specification (including many conformance test streams). In addition there are two application specific profiles defined in SMPTE 2047 parts 1 and 3 [8][10] and associated specifications (SMPTE 2047 parts 2 and 4 [9][11]), which define carriage of compressed data over standard (SDI) video interconnects.

Hardware based on the VC-2 standard is commercially available and in use by, amongst others, the British Broadcasting Corporation (BBC) and Österreichischer Rundfunk (ORF) the national, public service, broadcasters of the UK and Austria respectively. The two commercial applications are both designed to reuse legacy broadcast infrastructure. In one application high definition video is compressed to fit within a standard definition channel. This allows the re-use of existing standard definition infrastructure for transport of high definition within, or between, production or studio centres. It also allows transport of high definition video over longer cable runs than is possible with uncompressed high definition, due to lower clock rates. This is useful for some outside broadcasts. A second application is to compress “Full HD”, that is 1080 line progressive video at either 50 or 60 frames per second, over a channel designed for interlaced high definition, for a which a compression ratio of 5:2 is required. In both these applications a useful feature of the implementation is that a, low quality, representation of the compressed signal can be viewed directly on the link. So if, for example, the compressed high definition signal is connected to an unmodified low definition display, a (low quality) standard definition version of the video is displayed. The key to both these applications is low complexity and low latency. The latency in both these applications is less than a few milliseconds, less than can be achieved with other commercial codecs.

VC-2 Low Delay profile (VC-2 LD) is an atypical codec in several respects. It is strictly constant bit rate by design. Many other codecs are fundamentally variable bit rate codecs that achieve near a near constant bit rate by using a data buffer to smooth the rate, and adjusting the degree of quantisation applied to control the bit rate. This has the disadvantages that the buffer increases the overall latency. Furthermore, for a fixed bit rate channel, the nominal bit rate must be set lower than the maximum available to prevent the maximum bit rate exceeding the channel capacity. VC-2 LD, by contrast, does not suffer from these issues, which makes it particularly suitable for low latency, fixed bit rate applications, such as described above.

A second, less common, feature of VC-2 LD is that it is a parameterised codec. That is, a wide range of coding parameters are possible that trade off aspects of codec performance with other characteristics such as computational complexity, memory requirements, latency, video bit depth etc. By contrast, most other production codecs are specified for a fixed set of video formats and bit rates. For example, in the case of DNxHD/SMPTE VC-3, different variable length coding (VLC) is specified for different video formats and bit rates; so extension to new applications requires the publication (and standardisation) of new VLC tables. The flexibility of VC-2 LD means that it is easily adapted for many applications (without changes or additions to the standard), but it also means that it is difficult to characterise its performance over the whole of this multidimensional codec parameter space.

This paper provides a description and discussion of the innovative features of the VC-2 codec. In addition to those above this includes signal processing features used in the wavelet transform and quantisation that make VC-2 particularly suitable as an intermediate codec. The paper also presents the results of experiments to characterise the performance of VC-2 LD. Because of the codec’s flexibility the coding parameters are constrained so that the results are applicable to the types of applications described above. A low complexity, low latency set of coding parameters are investigated, together with a more complex set, which achieves better coding performance. In video production interlaced video is still widely used, as is progressive (i.e. “film”) content. So performance for both these video formats is considered. The experiments were conducted using high definition test material in the Y′CbCr 4:2:2 10 bit video format that is widely used for television production. However the VC-2 LD standard supports a wide range of other video formats too.

The next sections of this paper present an overview of the VC-2 Low Delay profile, its design principles and the algorithms it uses. It is just an overview since full details are available in the published specification [5]. However the standard only describes the definition of the codec, not the reasons for selecting those algorithms or the features that make it useful in practice. Furthermore the specification, like other codecs only defines the byte stream and how to decode it, but there are interesting issues on the encoder side too. The paper starts by considering the principles that were applied in the design process. It moves on to describe the algorithms used and highlights their innovative aspects. This is followed by an overview of the features of the codec, particularly their utility for production and archive applications.

After describing the codec the paper moves on to describe the experiments reported here. Firstly the experimental procedure is presented, followed by the results of the experiments, and then a discussion of these results. The paper concludes in the final section, which includes an overall summary of the paper.

2 Design Principles

This section discusses the principles that were applied in designing the VC-2 LD codec. The requirements of a production video codec differ significantly from a codec for end user delivery. Consequently the design principles are somewhat different too. Some important aspects of the design including image quality, complexity, latency, flexibility, ease of use and patents, are discussed below.

Complexity is a major factor for a production codec. In consumer equipment the design costs may be amortised over large production runs and so, perhaps, complexity is relatively unimportant. Volumes are much lower for professional video production equipment, so complexity directly affects the cost. But complexity also affects both the form factor and power consumption of equipment. Often, for example in outside broadcasts, equipment needs to be small, light and portable. Power consumption is important if equipment is powered by batteries and also for heat dissipation in compact equipment. In software applications (e.g. a video editor) low complexity supports responsive applications running on less powerful hardware. Low complexity makes the codec straightforward to implement and optimise. For all these reasons low complexity was essential in the design of VC-2 LD.

Latency is an important issue for some applications, such as “live” or “event driven” television programmes. Long latencies (many seconds) may be acceptable for end user distribution, but are unacceptable in a production context, For example delay is unacceptable where a presenter and interviewee are geographically separate. Interframe coding typically generates an unacceptably long latency (plus additional complexity), even though it provides better compression efficiency. Therefore VC-2 is an intra frame codec, which also simplifies applications such as switching streams and video editing. But even intra frame coding can generate too much latency if it codes whole pictures at a time. So VC-2 LD codes smaller regions to achieve ultra low latency (and yet VC-2 is a wavelet codec not a block transform codec – see below).

For production applications high image quality is essential. During the production process video is combined, processed and transcoded many times. At each stage quality may be lost, so headroom is required if the finished programme is to retain acceptable quality. Subject to the constraints of simplicity and low latency VC-2 LD tries to achieve the best possible decoded image quality. Image quality is one reason why VC-2 uses wavelet transforms, since they have proved (e.g. in JPEG2000 [12]) an effective compression technology.

A wide range of applications and video formats are used in video production. Ideally a production codec would be sufficiently flexible to accommodate this diversity. Applications vary in the amount of compression required, latency requirements, compression quality and complexity. Video formats vary in image size, bit depth, chroma subsampling (4:2:0, 4:2:2, 4:4:4) colour space (R′G′B′, Y′CbCr, primaries, colour matrix etc.) and whether they are interlaced or progressive. To accommodate these wide requirements all the parameters are configurable. The parameters are coded using variable length codes so that there are no inherent restrictions on parameter values.

With such a wide degree of flexibility it might be onerous, and error prone, to ensure that all parameters were correctly set in the byte stream. To make VC-2 straightforward to use it provides a set of defaults, for common video production formats, called the “base video formats”. Setting a specific base video format in the byte stream implicitly defines all the characteristics of that format. Not only does this simplify use, but it also minimises the scope for setting the video parameters incorrectly for common formats. For example, setting the base video format index to 11 implicitly defines the characteristics of standard, interlaced, HD at 60 fields/second (as used in the North America). So a base video format index of 11 implies 1920 pixels by 1080 lines, interlaced (top field first), with a 10 bit signal range, and appropriate colour primaries, colour matrix, 4:2:2 chroma subsampling and gamma curve. However all these default video format characteristics may be explicitly overridden in the byte stream. This allows less common, perhaps non-standard, video formats to be specified. For example within a video camera the sensor may have a non-standard resolution digitised to, for example, 12 or perhaps even 16 bits. By appropriately over-riding the base video format defaults the sensor's non-standard video format may be precisely specified in a compressed VC-2 stream.

An important aspect of codec design is intellectual property rights (patents). A plethora of video compression patents have been granted across the world. Commercial companies often use patents to control competition as part of their business strategy. International collaborations, such as MPEG [14][15] and ITU Video Coding Experts Group (VCEG) [16] typically pool their patents [17][18] to provide a revenue stream, which may be part of the motivation for contributing to codec standardisation. Whatever the rights and wrongs of patents it is clear that they affect the adoption of some coding technologies. Video production equipment is often made by (relatively) small companies and potential costs, legal uncertainties, and practical difficulties of managing patent licences sometimes discourage them from using patented technologies.

A principle in developing VC-2 was to try to avoid using patented technology. It is virtually impossible to prove the negative, that a codec does not infringe any patents. However, due to its simplicity and use of well established techniques, it is believed that VC-2 is patent free.

3 Codec Algorithms

The VC-2 LD is an intra-frame wavelet codec that uses the well known principles of transformation, prediction, quantisation and entropy coding, illustrated in figure 1, to achieve compression. This section describes how these techniques are implemented and highlights some of the innovative features embodied in the codec.

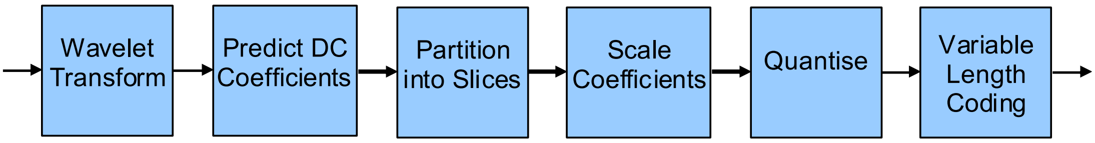

Figure 1: VC-2 LD signal processing chain

3.1 Wavelet Transform

VC-2 uses a wavelet transform, rather than a block transform such as the DCT, to achieve the best image quality and to provide a computationally efficient implementation. Other things being equal, wavelet transforms produces slightly better compression than DCTs or other block transforms [19], at least for intra frame coding. Wavelet transforms also avoid the blocking artefacts, that plague block transform codecs, because they don't divide the picture into juxtaposed "blocks" but, rather, transform the picture as a whole. And wavelet transforms can be implemented reversibly using integer arithmetic, which provides a simple, easily tested, implementation, and supports lossless coding.

This section only provides a brief overview of the theory of discrete wavelet transforms. There are many references (e.g. [20][21][22][23][24][25]) available which provide a more detailed discussion. This section tries to highlight the unusual features of VC-2, where this may differ from the approach taken by other codecs such as JPEG2000.

The discrete wavelet transform may be implemented in different ways. The theoretical derivation of the discrete wavelet comes from the study of filter banks using FIR filters, and wavelet transforms can be implemented in this way (known as convolutional implementation). In practice wavelet transforms are usually implemented using the more efficient "lifting" technique described below.

The discrete wavelet transform may be derived from the theory of the perfect reconstruction 2 channel filter bank, illustrated in figure 2. This splits a (one dimensional) signal into two signals at half the sampling rate, one containing predominantly low frequencies, the other containing predominantly high frequencies. By careful selection of the filters A to D the output can be a perfect reconstruction of the input.

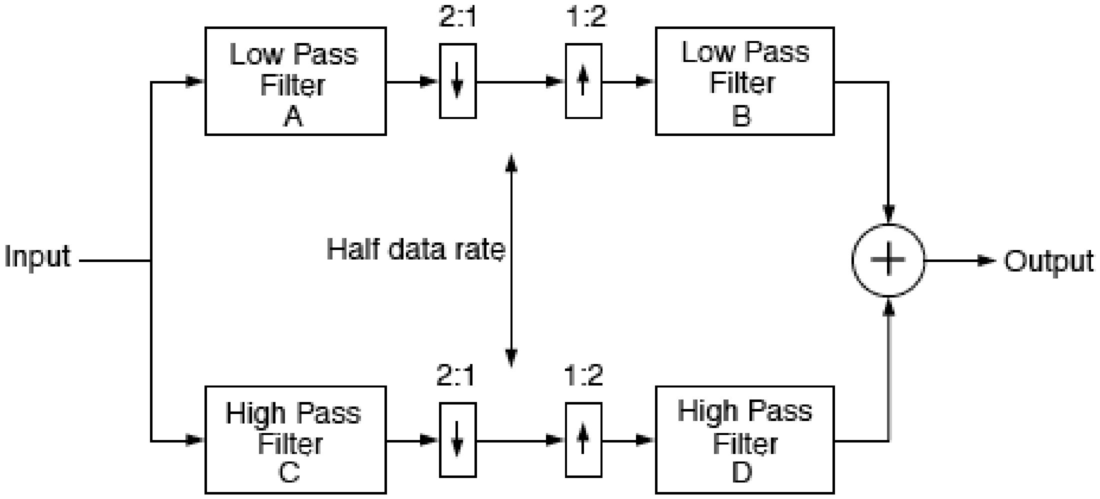

Figure 2: Perfect reconstruction 2 channel filter bank

The filters need not be perfect low or high pass filters, which is necessary for them to be implemented in practice. However this means that "low pass" signal also contains some high frequency aliasing and the "high pass" signal contains some low frequency aliasing, as illustrated in figure 3 (in which ω_n is the Nyquist frequency for the input signal). In figure 2 the alias components, shown hashed in figure 3, cancel perfectly. However when the low and high pass signals are quantised before reconstruction, as happens in any codec, the alias cancellation is no longer perfect, resulting in some distortion in the output signal. This has the important consequence of requiring additional accuracy bits for the low frequency signal, which is discussed below.

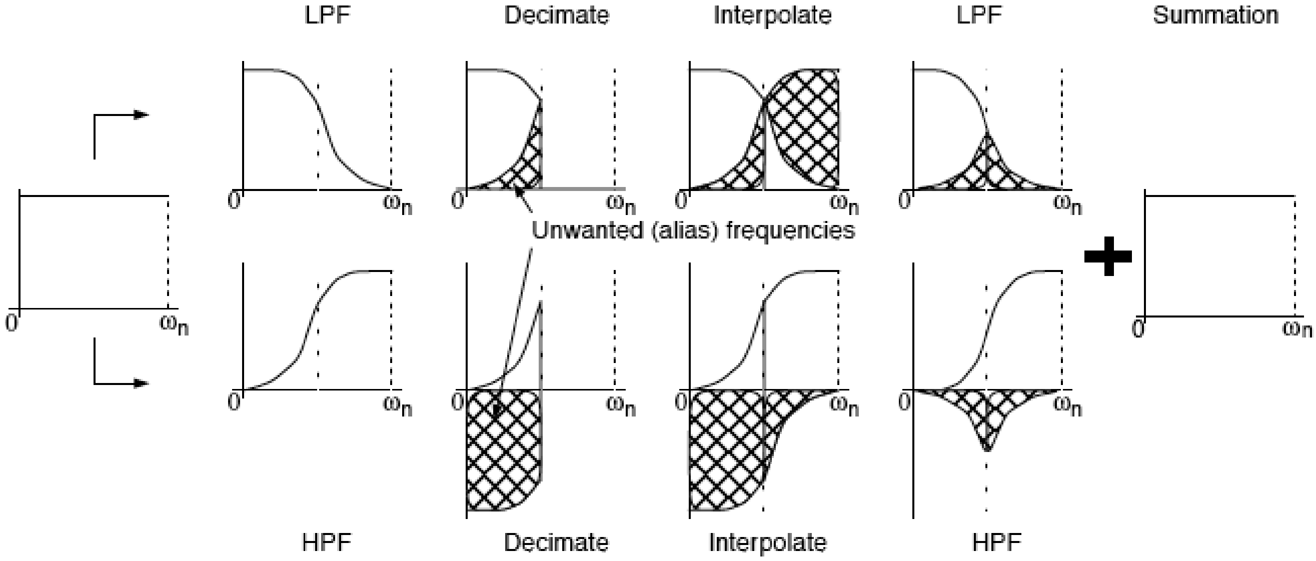

Figure 3: Signal and aliasing in a 2 channel perfect reconstruction filter bank

The two channel filter bank above represents a single stage of a one dimensional wavelet transform. Typically 3, 4 or more stages are required for a practical wavelet codec to achieve sufficient compression. This is achieved by repeatedly splitting the low frequency signal (whilst the high frequency part remains unchanged), which is illustrated in figure 4. Because each step of the wavelet decimation is self-contained, the reconstructed output is still identical to the input (barring quantization errors).

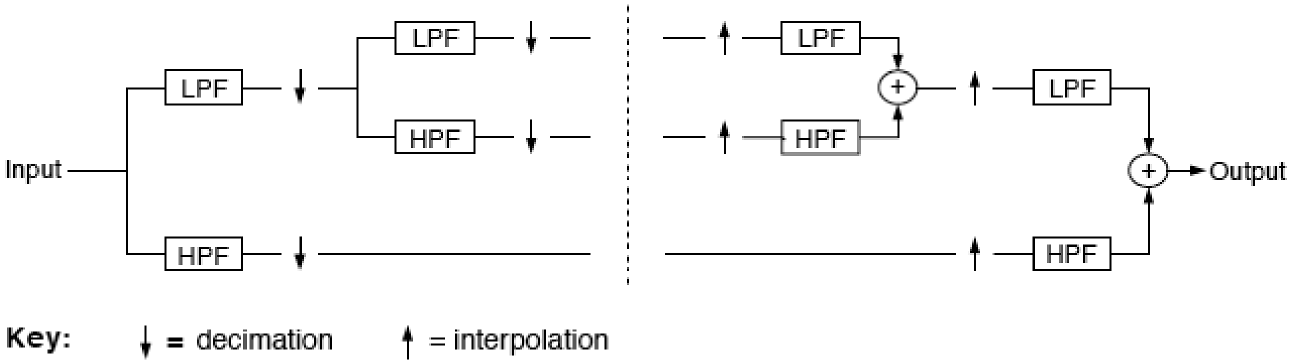

Figure 4: 2 level discrete wavelet transform

For image and video compression we have a two dimensional signal. So the wavelet transform is applied independently in both the vertical and horizontal directions to give a two dimensional discrete wavelet transform. This is “variables separable” processing and is illustrated in figure 5, which shows a two level, two dimensional transform, resulting in 7 wavelet subbands. Here the subbands are shown in conventional order with low frequencies in the top left and high, horizontal and vertical frequencies in the bottom right.

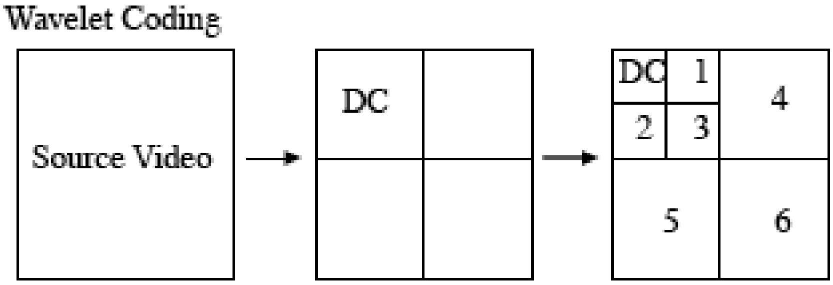

Figure 5: 2 level, 2 dimensional discrete wavelet transform

Figure 6 illustrates a two level wavelet transform applied to a real picture (zero is represented by mid-grey). The lowest “DC” frequencies provide a low resolution representation of the image with the other subband containing higher frequency information on the detail in the picture. Even with a two level transform such as this it is clear that most of the information is contained in the lowest frequencies, which only occupies 1/16th of the picture area. The higher frequencies tend to be of low amplitude. This is because the frequency division provided by the wavelet transform approximately matches the distribution of energy in a natural scene. These features of the transform allow it to form the basis of a compression algorithm.

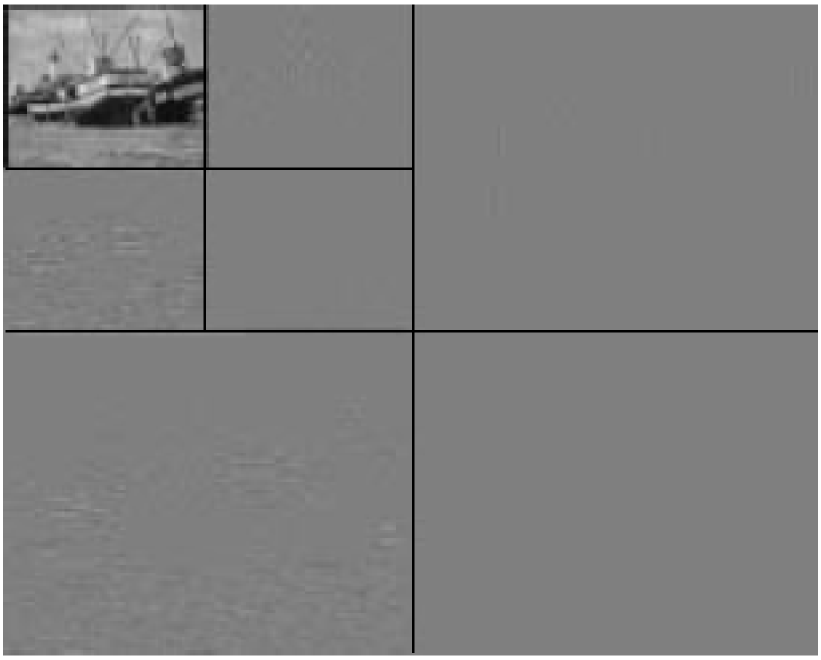

Figure 6: Transform coefficients for a 2 level, 2 dimensional discrete wavelet transform

3.2 Lifting

This section discusses the implementation of the wavelet transform in VC-2. It starts with the polyphase representation of the 2 channel perfect reconstruction filter bank, leading to the efficient lifting implementation used in VC-2.

The basic two channel perfect reconstruction filter, of figure 2, can be modified by placing the 2:1 down samplers before the filters on the analysis side, and the 2:1 upsamplers after the filters on the synthesis side. To do so the z transform representation of filter A, A(z) becomes A($z^{½}$) and similarly for the other filters. When the downsamplers and upsamplers are relocated in this way the two channel filter bank may be re-drawn as shown in figure 7, which is called the polyphase representation of the filter bank.

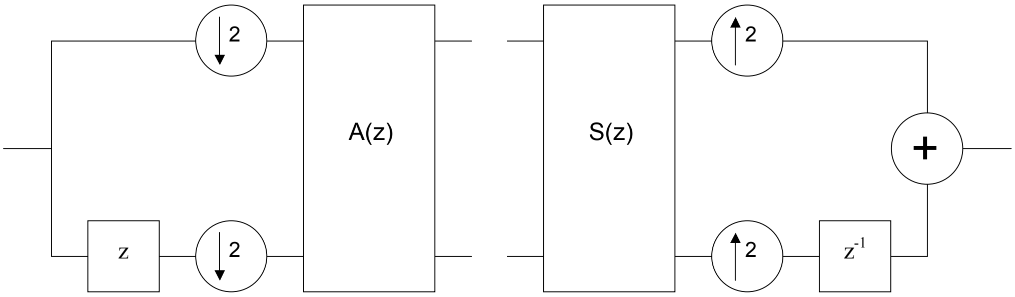

Figure 7: Polyphase representation of perfect reconstruction 2 channel filter bank

In the polyphase representation of the filter bank A(z) now represents a matrix of analysis filter responses and S(z) a matrix of synthesis filter responses. In this representation, linear combinations of filters operate on both even and odd samples to produce new even and odd samples:

$$\begin{pmatrix} x_e^{out}(z) \\ x_o^{out}(z) \end{pmatrix} = A(z)\begin{pmatrix} x_e^{in}(z) \\ x_o^{in}(z) \end{pmatrix}$$ **Equation 1**

Since the filter process is invertible, it can be shown [25] that the analysis and synthesis matrices are related by:

$$A(z) = (S(z^{-1})^{T})^{-1} \qquad \textbf{Equation 2}$$

Hence, in particular, both the analysis and synthesis matrices are invertible. It can also be shown [20] that this means that they are (up to gain factors and delays) factorisable into products of upper and lower triangular matrices as follows:

$$A(z) = \begin{pmatrix} 1 & a_1(z) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ b_1(z) & 1 \end{pmatrix} \begin{pmatrix} 1 & a_2(z) \\ 0 & 1 \end{pmatrix} \ldots.. \qquad \textbf{Equation 3}$$

Each upper or lower-triangular polyphase matrix represents a so-called lifting stage whereby either even coefficients are modified solely by odd coefficients or odd coefficients solely by even coefficients. For example, if

$$\begin{pmatrix} x_e^{out}(z) \\ x_o^{out}(z) \end{pmatrix} = \begin{pmatrix} 1 & a(z) \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x_e^{in}(z) \\ x_o^{in}(z) \end{pmatrix} \qquad \textbf{Equation 4}$$

Then,

$$\begin{aligned} x_e^{out}(z) &= x_e^{in}(z) + a(z)x_o^{in}(z) \\ x_o^{out}(z) &= x_o^{in}(z) \end{aligned} \qquad \textbf{Equation 5}$$

and the filter a(z) has been applied to the odd coefficients and then used to modify the even coefficients. Not only is this computationally efficient by breaking long filters into a number of shorter filters successively applied, but the factorization into such filter stages allows for computations to be done in-place.

VC-2 implements wavelet transforms using a lifting architecture, a simple diagram of which is shown in figure 8. Blocks P & U usually comprise FIR filters[1]. They correspond to the factors of the analysis matrix A(z) of equation 3. The block denoted “P” is the “predict” stage, so called because it predicts the odd samples from the even ones. So the bottom line of the lattice in the diagram represents high frequencies in the signal. The block denoted “U” is the “update” stage because, in some way, it updates the even samples. The need for an update stage can be seen from the algebraic analysis above. On a more intuitive level the function of the update stage is to reduce the degree of aliasing present in the low frequency signal (on the top line of the lattice). The diagram only shows a single predict and update stage, but there may be more than one of each (corresponding to a factorisation into more than one upper and lower triangular matrix in equation 3). Furthermore the order of the predict and update may be reversed (provided this is done for both analysis and synthesis). Indeed having the update stage before the predict stage can result in less aliasing in the low pass signal (see below).

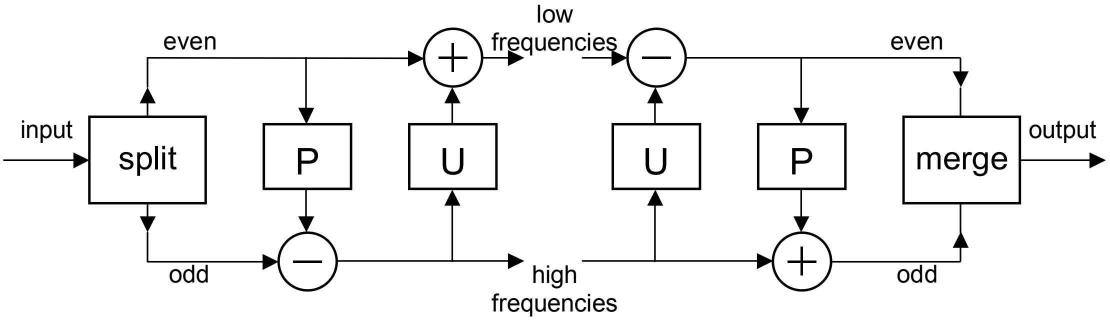

Figure 8: Lifting implementation of perfect reconstruction 2 channel filter bank

[1] Note that in this simple diagram the P & U FIR filters may be non-causal. In practice non-causal FIR filters are replaced by causal one (by the simple expedient of delaying the output of the non-causal filter). This in turn means that there have to be compensating delays elsewhere in the diagram. However the principles of lifting remain the same.

It is straightforward to see that the output is an exact ("perfect") reconstruction of the input, because the synthesis (right hand side) simply undoes the processing performed by the analysis (left hand side). The mathematical analysis above presents lifting from the perspective of linear signal analysis. But lifting is more general than this. The predict and update stages may include non-linear processing and the system still achieves perfect reconstruction [26][27][28].

3.3 Integer Arithmetic

VC-2 is implemented using integer arithmetic. The predict and update blocks in the lifting implementation of the wavelet transform are implemented as FIR filters using integer arithmetic. These filters are only approximately linear because they include rounding as part of their implementation. Nevertheless in a lifting implementation we still have perfect reconstruction in spite of rounding. Thus the integer wavelet transform in VC-2 provides a perfect reconstruction integer to integer transform. This is in contrast to the (small) arithmetic errors in fixed point implementations of DCT used in many codecs ([29][30]) and in contrast to the Daubechies 9/7 wavelet kernel used in JPEG2000 (which specifies coefficient/scaling factors as floating point numbers). A perfect reconstruction integer transform is essential for lossless coding. So JPEG2000 uses the Le Gall 5,3 wavelet kernel (which is a perfect reconstruction integer transform), not the Daubechies 9/7 kernel, for lossless coding. VC-2 LD however, being fixed bit rate, is not designed for lossless coding (although the other profiles of VC-2 do support it). From a more practical perspective having a perfect reconstruction integer transform simplifies the debugging and testing of implementations, particularly in hardware.

3.4 Arithmetic Precision

Knowing that the compression transform is perfect reconstruction makes it easy to ignore the dynamic range of the transform coefficients. After all, it seems that irrespective of their dynamic range the reconstructed output will still be the same as the input. But this assumption, which is often made in wavelet codecs, is only true in the absence of quantisation. Despite perfect reconstruction it is important to maintain the correct dynamic range for the transform coefficients. VC-2 maintains the correct dynamic range by design but other codecs do not necessarily do so, resulting in visible artefacts even at high bit rates. These artefacts do not significantly affect PSNR (a failure of PSNR as a quality metric) but are easily seen as temporal brightness variations over large areas. Subjectively this appears as an annoying flicker or flutter, which is present even at low compression factors and high bit rates.

Low frequency ("DC") wavelet subbands need more dynamic range than higher frequency subbands, and also more dynamic range than the original input signal. VC-2 provides additional dynamic range by specifying a left bit shift (i.e. gain) to the low frequency coefficients (including the original image) before applying each level of the wavelet transform. To understand why, consider the bandwidth of the low frequency subband. At each level of the wavelet transform the bandwidth of the low frequencies is reduced by a factor of (approximately) 2 both horizontally and vertically. Reducing the noise bandwidth by a (total) factor of 4 in this way increases the dynamic range by 1 bit. If the filter itself does not have sufficient gain[2] a left shift is applied before performing the transform in order to increase the dynamic range.

Without sufficient dynamic range for low frequency coefficients, low frequency information is moved into the high frequency coefficients (as additional aliasing) during wavelet analysis. Since the transform itself has perfect reconstruction, and the low frequency components have insufficient dynamic range to support the data, the signal is forced to go via the high frequency channel. This only causes problems when quantisation is applied. Low level, high frequency, signals (precisely those corresponding to DC aliasing) tend to be lost after quantisation. This results in a DC error after synthesis. Since the DC error varies from frame to frame this is perceived as flutter or flicker on reconstructed pictures.

Other codecs suffer from a lack of dynamic range for DC coefficients. For example MPEG-2

[2] In VC-2 the "Fidelity" wavelet kernel provides sufficient gain so that no additional shift is required to increase the dynamic range of low frequency components.

[31][32] supports a maximum precision for Intra DC coefficients of 11 bits [31, Table 6-13 Intra DC Precision], and typically fewer bits are actually used. In MPEG-2 an 8x8 DCT is used as the transform. Hence the noise bandwidth of the DC coefficients is reduced by a factor of 8 in each dimension, requiring an additional 3 bits of precision compared to the original video. So, in principle, MPEG-2 only has sufficient Intra DC coefficient precision to support 8 bit video. Production video usually has 10 bit precision (compared to 8 bit for end user distribution) Consequently MPEG-2 (a.k.a. IMX) does not provide sufficient DC precision. This is a problem making MPEG-2 less suitable as an intermediate codec, when processes such as colour grading or even simply adjusting the contrast are applied, because it makes blocking artefacts much more apparent.

3.5 Boundary Handling (or Edge Extension)

A discrete wavelet transform is constructed from a set of iterated perfect reconstruction filter banks. The filters have a finite length. So, when filtering a pixel at the edge of a picture the filters extend beyond the picture edge. So, what should be done at the edge of the picture? Whatever is done the perfect reconstruction nature of the wavelet transform needs to be maintained. The VC-2 codec uses a non-conventional twist on boundary handling to simplify the implementation and improve the signal processing.

The conventional answer to boundary handling, used by JPEG2000, is to use symmetric extension. Consider extending the whole picture in the first instance (but considering only one dimension). If the start of the signal (e.g. the left hand edge of a picture) consists of pixels:

A B C D E F G …….

this can be symmetrically extended, to the left, as:

……. G F E D C B A B C D E F G …….

Using this type of extension, and odd filter lengths (such as in the LeGall 5,3 and the Daubechies 9,7 wavelet kernels), it can be shown that perfect reconstruction is maintained even at the edge of the picture.

In practice discrete wavelet transforms are implemented using iterated lifting stages. So whatever form of edge extension is used must be implementable within individual lifting stages. Since lifting filters operate on a subset of either even or odd samples the edge extension for the even samples must require only even samples and vice versa. For the edge extension above, known as “whole sample” symmetric extension, the even sample sequence is extended using only even samples and vice versa for the odd samples. It can be shown that doing the edge extension prior to calculating the wavelet transform is equivalent to doing the extension during the lifting steps themselves [33].

In contrast to other wavelet codecs VC-2 uses an unconventional approach to handling picture boundaries. Instead of symmetric extension it simply extends, at each lifting stage, with the nearest available sample. This not only simplifies the implementation but also improves compression. It is less easy to see how this corresponds to extending the whole picture. But we can get a feel for what happens by considering a first lifting stage, in one dimension, as above. Starting with the same signal as above:

A B C D E F G …….

In VC-2 this is extended, in the first lifting stage, as:

……. A B A B A B A B C D E F G …….

Note that even samples are extended with the even sample nearest the edge and odd samples with the odd sample nearest the edge. This method of edge extension, applied at each lifting stage, still results in perfect reconstruction. As Brislawn says in [33], “Invertibility of lifting steps is independent of analysis bank boundary handling, provided the same boundary handling is performed in the synthesis bank.”

The simple boundary handling approach adopted in VC-2 would be expected to improve compression at the edge of pictures compared to symmetric extension. Natural images tend to be predominantly low pass in nature. We may therefore model their edge as a Taylor series with only a few significant coefficients. The simplest case is of a constant value at the edge of the picture. In this case both symmetric extension and VC-2 extension is the same; the edge is just extended with the constant value. The next simplest case is when the pixel values vary linearly with distance from the edge. Now symmetric extension yields a significant cusp at the edge of the picture. When this is analysed by the wavelet filter banks it will inevitably produce significant high frequency coefficients corresponding to the cusp. In the same circumstances the nearest neighbour edge extension of VC-2 only produces a low level frequency component at half sampling frequency. The lower amplitude of high frequency coefficients produced by nearest neighbour edge extension would be expected to yield greater compression efficiency. Of course the difference between the two approaches is small or non-existent for short wavelet filter kernels. However for a relatively long kernel, such as the Daubechies 9,7 kernel, the difference is potentially significant.

3.6 Processing Order (of the Wavelet Transform)

The order in which processing is performed is often ignored but it affects the decoded output. Wavelet transforms were developed from the theory of linear filtering, and so are usually considered to be linear transforms. For linear transforms the processing order does not matter. But integer wavelet transforms are not quite linear due to arithmetic rounding and truncation. So performing the processing in a different order would result in a slightly different output. In particular a defined order is essential to achieve lossless coding. Therefore the VC-2 standard specifies the wavelet transform processing order.

What is the best processing order? One possibility is to perform every level of the vertical wavelet transform first, followed by every level of the horizontal wavelet transform, or vice versa. Alternatively you could perform both vertical and horizontal processing at each level before moving on to the next level. The question is, what is best?

In VC-2 the decision was made to process each level separately and, within each level, vertical synthesis is performed first in the decoder. This implies that horizontal analysis is performed first in the encoder, which is the reverse of the decoder. This order of processing was chosen in order to minimise resource requirements in the decoder. This depends on the assumption that the order in which video is stored is raster scanned order, which is by far the most common order. With data in this order, and by performing vertical synthesis first, a typical software system will cache whole lines of video during the vertical synthesis. The horizontal synthesis can then be performed on these cached lines minimising memory accesses. Similar advantages can be gained in a hardware implementation. To some extent the order of processing is arbitrary; however the order chosen for VC-2 offers the maximum potential for efficient implementation. For a relatively long kernel, such as the Daubechies 9,7 kernel, the difference is potentially significant.

3.7 DC Prediction

Spatial prediction of the DC subband wavelet coefficients is performed to improve compression efficiency. Each DC coefficient is predicted as the mean of the three, previously quantised values to its left, top-left and above it. Since the quantised values used for prediction are also available at the decoder this prediction can be undone, losslessly, at the decoder.

DC coefficient prediction may be regarded as a tool to support ultra low latency coding. To achieve the lowest latency only a two or three level transform would be used. Furthermore a short filter kernel, typically the Haar kernel, would be used. These coding parameters do not yield the best compression efficiency. So DC prediction is applied to improve compression efficiency. However, without the constraint of low latency a higher level wavelet transform might be used to achieve the same result.

3.8 Slices

Wavelet codecs typically process whole subbands at a time. Processing whole subbands requires that the whole picture be processed before any coded data can be output. This limits the minimum latency that can be achieved by the codec. For example, with a 50Hz frame or field rate (standard in Europe), the minimum latency is 20ms for encoding and the same for decoding. So the encode/decode latency cannot be less than 40ms. After allowing time for data buffering and processing, encode/decode latency is more typically of the order of 100ms. Latency of this order is highly undesirable, or impractical, for some video production applications. So some method is required to achieve lower latency.

One way to reduce latency is to divide the picture into blocks or tiles and process these separately. But splitting into blocks (block transform codecs) or tiles (JPEG200) risks introducing “blocking” artefacts, and avoiding these was one motivation for using wavelet transforms in the first place.

The VC-2 LD codec takes an alternative approach to dividing the picture and reducing latency. Instead of splitting the original picture into regions, the coefficients themselves are divided after performing the wavelet transform (see figure 1 above, diagram of sequence of operations). Each set of wavelet coefficients, called “slices” in VC-2, loosely correspond to regions of the original image. All slices contain the same number of coefficients. Each slice contains one or more DC coefficients corresponding to a rectangular region in the DC subband. Slices also contain the higher frequency wavelet coefficients corresponding to their DC coefficients. So each slice contains complete trees of wavelet coefficients. For example with a 3 level wavelet transform, and a slice containing a single DC coefficient, the slice contains a total of 64 wavelet coefficients for each colour component (1 DC coefficient and 63 higher frequency coefficients).

Figure 9 illustrates the way in which the wavelet transform is divided into slices. A two level wavelet transform is performed on a 16x16 pixel picture. The wavelet transform on the left is then split into 16 slices, each containing a single DC coefficient.

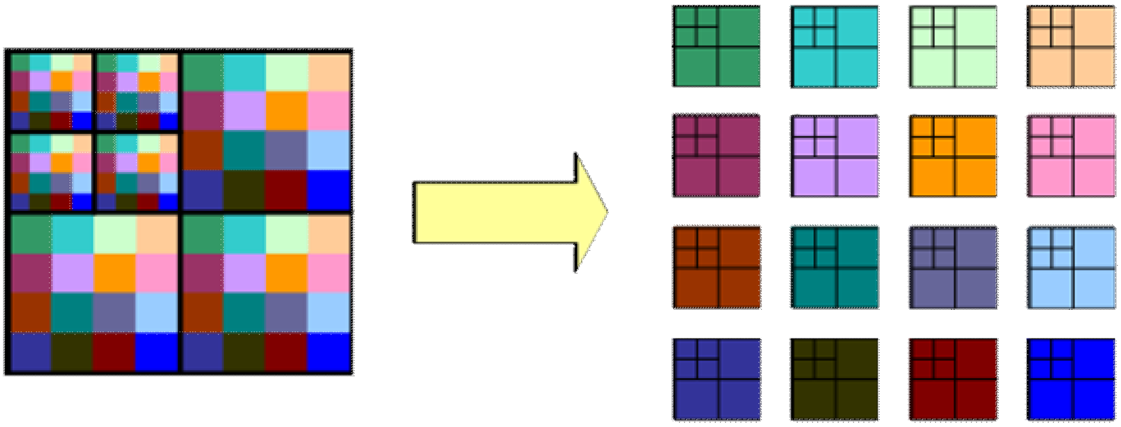

Figure 9: Partitioning a wavelet transform into slices

Partitioning the transform coefficients into slices is fundamentally different to partitioning pixels into blocks or tiles. Conceptually the coefficients in VC-2 LD slices are calculated from the whole image. In practice, since the filter lengths used in the wavelet transforms are of finite length, the slices corresponding to the top of the picture may be coded and transmitted before the bottom of the picture has arrived at the encoder. In this way, using small slices and short wavelet kernels, latencies of only a few milliseconds can be achieved.

3.9 Quantisation Matrix

To reduce complexity VC-2 does not include scaling factors in its lifting implementation, and consequently each subband has a different gain. The differing subband gains are compensated by the use of a quantisation matrix. Conceptually a quantisation matrix would scale the coefficients before they are quantised. However the implementation in VC-2 LD is to adjust the quantiser (described below) rather than to directly scale the coefficients.

A custom quantisation matrix may be coded in the VC-2 byte stream. This could, for example, allow for the relative sensitivity of the human visual system to different spatial frequencies, and so provide optimum subjective quantisation. However, in practice, VC-2 LD is used at relative low compression ratios to achieve high quality decoded pictures. For these applications good results are achieved by choosing the quantisation matrix that maximises the PSNR of the decoded picture.

To simplify its use, so that a quantisation matrix does not always have to be encoded in the byte stream, the VC-2 LD specification includes default quantisation matrices. These are specified for each wavelet kernel and up to 4 level transforms.

The default quantization matrices are designed to counteract the differential power gain of the various wavelet filters, so that quantization noise from each subband is weighted equally in terms of its contribution to noise power when transformed back into the picture domain. Let α and β represent the RMS noise gain factors of the low-pass and high-pass wavelet filters used in wavelet decomposition. From figure 2, B and D are the low-pass and high-pass synthesis filters. The gain factors are given by:

$$\alpha = \left(\sum_n B(n)^2 \right)^{1/2} \text{ and } \beta = \left(\sum_n D(n)^2 \right)^{1/2} \qquad \textbf{Equation 6}$$

Note that the wavelet transform is specified in terms of lifting stages. But to find α and β we need the equivalent FIR filter responses. These can be found from the lifting implementation by removing the up and down samplers in figure 2 above and evaluating the impulse response.

In a single level of wavelet decomposition, quantization noise in each of the four subbands is, therefore, weighted by the factors shown in figure 10.

LL - α^2	HL - $\alpha\beta$
LH - $\alpha\beta$	HH - β^2

Figure 10: Quantisation noise weightings in a single level of wavelet transform

For higher levels of decomposition, these subband weighting factors iterate in the same manner as the wavelet transform itself. For example, with a 1-level decomposition, the first level LL band, with weight α^2 is further decomposed to give four more bands with weights as for the 1-level decomposition, but multiplied by α^2. This yields the weights shown in figure 11 below.

Note also that these factors must also take into account the shift factors that are used to add accuracy bits prior to each wavelet decomposition stage. For a filter shift of d, α and β are each multiplied by $2^{-d/2}$.

The differential power gains for the subbands are compensated by adjusting the quantisation factor appropriately, as described in the next section.

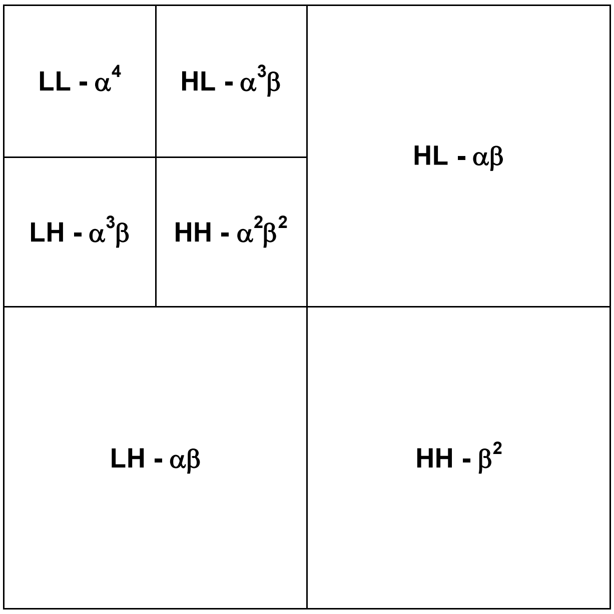

Figure 11: Quantisation noise weightings in a two level wavelet transform

3.10Quantisation

Quantisation in VC-2 is performed with a dead zone quantiser. The quantised value, x_q, of a wavelet coefficient x, is given by:

$$x_q = \begin{cases} 4x/qf & \text{for } x \geq 0 \\ -\left(|4x|/qf\right) & \text{for } x < 0 \end{cases} \qquad \textbf{Equation 7}$$

where qf is the “quantisation factor” and, since both x and qf are integers, the division is performed as an integer division. The quantisation factor is constrained to be:

$$qf = 4 \cdot 2^{(qi/4)} \qquad \textbf{Equation 8}$$

Where qi is an integer called the quantisation index, and the quantisation factor is a rounded integer approximation to the equation 8 calculated using real numbers. For details refer to the standard, which has been defined purely in terms of integer operations to avoid ambiguity.

These two equations, taken together, mean that:

$$x_q \approx x / 2^{qi/4} \qquad \textbf{Equation 9}$$

In the decoder the transform coefficients are reconstructed according to:

$$x_r = qf \cdot x_q + qo \qquad \textbf{Equation 10}$$

where qo is called the quantisation offset. The forward quantisation process involves truncation as part of the integer division by the quantisation factor. So the quantisation offset is added during reconstruction to give a more accurate estimate of the original transform coefficient. The quantisation offset is, approximately, half the quantisation factor. The optimum quantisation offset depends on the probability density function (pdf) of the unquantised coefficients. Since this pdf varies with both compression factor and subband the value of (usually) one half the quantisation factor was chosen for simplicity.

An important consideration in the design of VC-2, as a production or intermediate codec, was to be able to decode and recode with minimal additional impairment. The quantisation offset was carefully designed so that following two signal chains are equivalent.

1. Quantize -> Inverse Quantize -> (Re-)Quantize
2. Quantize

This is so provided the re-quantisation is performed using the same quantiser index. This property allows for coding with no multi-generational loss, provided that the same quantization index can be selected by every encoding stage. And the original quantisation index can be determined, without additional side chain information, by choosing the quantisation index that minimises the quantization error in subsequent stages. If the picture had been previously coded then the optimum quantisation index (which will not necessarily be the smallest index) will yield zero quantization error.

3.11 Quantisation using the Quantisation Matrix

VC-2 LD includes a single quantisation index for each slice. This quantisation index controls quantisation for all colour components and all subbands. As discussed above the quantisation indices used for an individual subband must be adjusted to allow for the different gain of each subband. This section describes how that is done.

One option might be to add an offset to the slice quantisation index to get the subband quantisation index. But if VC-2 LD implemented this strategy it would result in some subbands having a minimum quantisation index greater than zero (i.e. they could never be coded with a quantisation index of zero). Consequently those subbands could never be coded losslessly, or near losslessly.

Instead VC-2 LD subtracts an offset from the slice quantiser index to generate the adjusted subband quantiser index. This is subject to the caveat that no subband quantiser index may be negative. In this way VC-2 LD ensures that slices can be coded losslessly, or near losslessly, if there is sufficient bandwidth to do so.

The offset for a subband is determined from the normalized power gain for that subband. Normalization is performed by dividing by the smallest power gain of all subbands, and in this way ensures that the smallest offset is zero. The actual offset is determined from the normalized subband power gain, w, by computing 4*log2(w) rounded to the nearest integer.

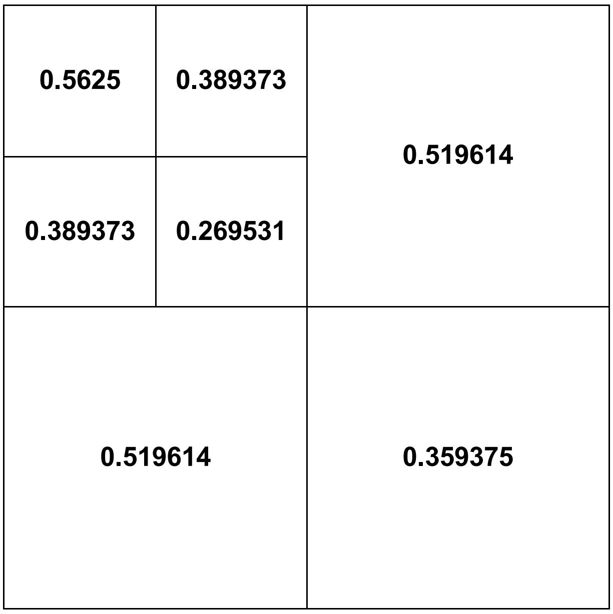

Figure 12: Noise weighting in a two level LeGall wavelet transform (including arithmetic shift)

Consider, for example, a 2 level Le Gall wavelet transform. For the Le Gall kernel the gain factors α and β, described above, are 1.224744871 and 0.847791248 respectively. Allowing for the one bit of shift, applied to increase coefficient precision (giving a further gain of 0.5 per wavelet stage), the relative noise gains of the reconstruction filters are as shown in figure 12. Normalising these by the smallest value (0.269531), calculating 4.log2 of the result and rounding to the nearest integer gives the quantisation matrix for this case shown in figure 13.

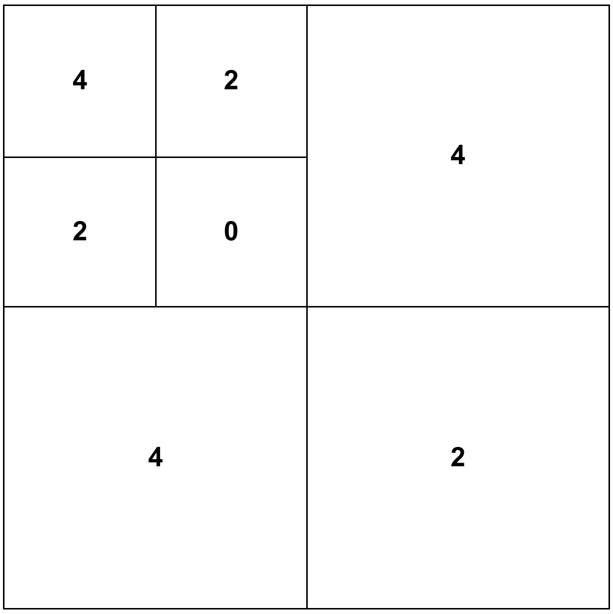

Figure 13: Quantisation matrix for a two level LeGall wavelet transform

During reconstruction the slice quantisation index (which is coded in the compressed stream) is modified by the quantisation matrix before inverse quantisation. For example, with a 2 level Le Gall kernel as above, if the slice quantisation index was 3 then the quantisation index for the DC coefficients would be 0 (3-4 and limited to zero) and the quantisation index applied to the highest frequencies would be 1 (3-2).

3.12 Entropy Coding (Variable Length Coding)

A simple deterministic variable length code (VLC) is used to entropy code quantised wavelet coefficients in VC-2. The code chosen is a rearrangement of exp-Golomb coding in which the codes correspond to non-negative integers. To form the exp-Golomb code of a (binary) number you add 1 and then prepend the number of digits, less 1, zeros to the front. For example, to code the number 3, i.e. 11 binary, you add one, giving 100 binary. This has 3 digits so you prepend two zeros giving the exp-Golomb code 00100. Basically this is a unary code (i.e. a sequence of zero), which tells you the number of (binary) digits, followed by the number itself. Table 1 shows the first few exp-Golomb codes.

Number	Exp-Golomb Code
0	1
1	010
2	011
3	00100
4	00101
5	00110
6	00111
7	0001000

Table 1: Exp-Golomb codes

VC-2 uses an interleaved version of the exp-Golomb code. That is, it interleaves the initial unary code (sequence of zeros) with the number itself. This gives a sequence of pairs of a “follow” bit followed by a “data” bit. The “follow” bits are the initial unary code from the equivalent non-interleaved exp-Golomb code. The sequence is terminated when a “follow” bit equals one. Continuing with the example above the first three digits (including the 1) are the follow bits and the last two digits are the data bits. Note that the data bits are interleaved least significant bit first. To form the interleaved code we simply interleave the follow and data bits. In this case (for the number 3) we get 00001. Note that all interleaved codes are an odd number of bits. Table 2 shows the first few interleaved exp-Golomb codes.

Number	Interleaved Exp-Golomb Code
0	1
1	001
2	011
3	00001
4	00011
5	01001
6	01011
7	0000001

Table 2: Interleaved exp-Golomb codes

Interleaved exp-Golomb codes are used because they lead to a simpler implementation. Interleaved codes are easier to parse and, in software, they may be decoded in a single loop whereas the conventional form requires two loops.

Quantised wavelet coefficients are signed integers. In VC-2 these are coded using a sign and magnitude format. The sign bit is coded after the magnitude value so that it may be omitted for the special case of zero.

This form of deterministic variable length code was chosen for simplicity. Inevitably the lengths of the codes do not accurately match the probability density function (pdf) of the quantised transform coefficients. Hence this VLC does not achieve the maximum possible compression. An alternative would be to use a more efficient entropy code such as Huffman or arithmetic coding. Not only would this be more complex but, more significantly, it is much less flexible. The pdfs of the quantised coefficients change both with subband and quantisation index (and also with wavelet kernels and all the other coding parameters). Consequently different variable length codes would be needed for each set of coding parameters and bit rate (or more complex adaptive entropy coding would be required). It might be possible to embed the VLCs in the coded stream. But this is impractical since it requires the user to determine the correct VLCs to use. The alternative, which is commonly used, e.g. with DNxHD (a.k.a. SMPTE VC-3), is to restrict the coding options, particularly the bit rate, to a few commonly used permutations. Then, whenever a new bit rate or set of coding parameters are required, the standard is extended and a new set of VLCs are provided. By using a single fixed (and simple) variable length code VC-2 allows precisely the same coding to be applied to any image format, any set of coding parameters and for any bit rate, thereby providing a great deal of flexibility.

Arithmetic coding provides substantial coding gains (perhaps 3:1) at high compression ratios. This is due to the high probability of transform coefficients quantised to zero. However the coding gain is much lower, perhaps only 5:4, for low compression ratios and lossless coding because there are many fewer zero coefficients. Bearing in mind that VC-2 LD is aimed at low compression ratios, and that a design objective was a simple codec, arithmetic coding is not an appealing option.

In designing VC-2 LD we wished to retain flexibility, which excluded the use of multiple VLCs for different sets of coding parameters and bit rates. We also wished to avoid the complexity of arithmetic coding (which *is* defined for the main profile of VC 2). The use of the interleaved exp-Golomb code was the compromise that was chosen.

One other form of entropy coding, “early termination” is also used in VC-2 LD, which is described below, in “Coded Slice Structure”. This is broadly analogous to run length coding in a block transform codec.

3.13 Texture Masking & Rate Control

In “busy” regions of the picture it is more difficult to see compression artefacts. This is a well known subjective effect called texture masking [34][35][36][37]. For example, noise is much more visible in plain regions of a picture than in busy regions. VC-2 LD implicitly takes advantage of this effect to mask compression artefacts in busy regions of the picture. Whilst texture masking in VC-2 LD improves the subjective quality of decoded pictures it is another feature not reflected in PSNR measurements that are typically used to assess quality. Texture masking in VC-2 LD is a consequence of its radically different approach to quantisation and rate control compared to the conventional approach used by many codecs. So texture masking requires no additional information to be sent in the coded stream.

Most video codecs use a buffer to smooth the raw variable bit rate from the encoder. When the buffer gets full quantisation is increased to reduce the bit rate and vice versa. In this way negative feedback is applied to control the bit rate, preventing buffer overflow or underflow. Much research has been conducted to find the best rate control algorithms, which can be quite complex. With this conventional model of rate control the level of quantisation applied is fairly constant, i.e. does not vary spatially. Ideally the level of quantisation would be lower in plain areas and higher in busy areas. But with conventional rate control this does not usually happen, i.e. texture masking is not used (although a codec could be specifically enhanced to include it). The limited use of texture masking is, perhaps, a consequence of an over reliance on the use of the PSNR metric.

Conventional rate control is not ideal for production codecs. In addition to the difficulty of taking advantage of texture masking it also leads to additional latency and to underutilisation of channel bandwidth. The additional latency is simply a consequence of needing time for the rate control buffer to become about half full. Even when the bit rate is smoothed using a buffer and rate control there remains some variability in bit rate. Often communication links in programme production, particularly real time links, have a fixed bit rate. With such links it is not possible to exceed the maximum bit rate for even a brief period. So, if a compressed stream has even a slightly variable bit rate, as it typically does, the target bit rate on the encoder must be set lower than the channel bit rate to allow for overshoots. In practice deciding the amount of headroom to leave can be quite tricky. But, whatever the headroom, some potential bandwidth cannot be utilised, resulting in a lower quality picture than is necessary.

The VC-2 LD codec does not require a buffer to smooth the bit rate. This allows it to achieve low latency and take full advantage of the entire available bit rate. It also implicitly results in texture masking. In VC-2 LD each slice is allocated substantially the same number of bytes in the coded stream[3]. This allows the quantisation index for that slice to be quickly determined without requiring feedback control from a buffer. A consequence of this approach is that the quantiser index varies spatially. In busy regions of the picture the quantisation index is higher and it is lower in plain regions of the picture. This implicitly implements texture masking. Whilst this is not, perhaps, the ideal way to achieve texture masking it is, nevertheless, a useful feature of VC 2 LD.

3.14 Coded Slice Structure & "Early Termination"

The VC-2 LD stream itself consists of a header, detailed in the standard, followed by a sequence of coded slices. The format of the coded slice is illustrated in figure 14. Each slice is a whole number of bytes but, internally, is divided into sections that are not aligned with byte boundaries.

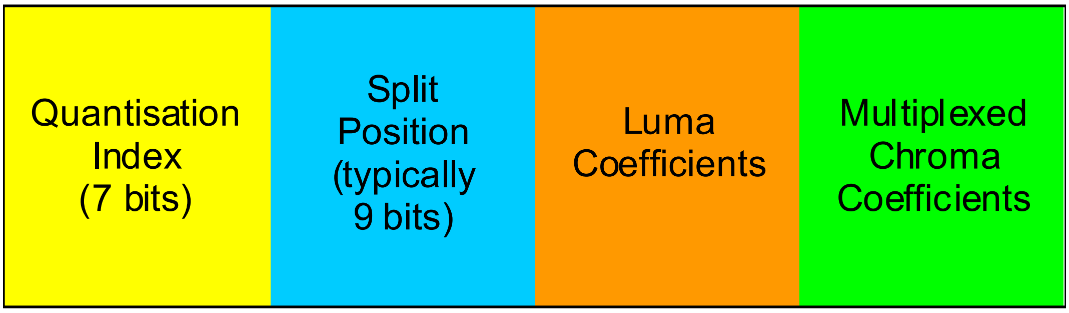
Quantisation Index (7 bits)
Split Position (typically 9 bits)
Luma Coefficients
Multiplexed Chroma Coefficients

Figure 14: Format of VC-2 LD coded slice

The purpose of the slice is to contain the quantised wavelet coefficients. It also needs to contain the slice quantisation index. The luma and chroma coefficients form most of the slice payload. The chroma coefficients are multiplexed together as CbCr pairs.

[3] Actually the number of bytes allocated per slice can vary by one byte, in a deterministic way, to allow the codec to precisely achieve any desired bit rate. For details refer to the standard itself. However this small potential variability in the number of bytes per slice does not affect the arguments above.

A further form of entropy coding is used in addition to the variable length coding described above. The wavelet coefficients are scanned in frequency order from low frequency to high frequency. Natural images tend to be predominantly low pass in nature. So it may happen that the highest frequency subband(s) contain a string of coefficients all quantised to zero. This would result in a trailing sequence of “1”s at the end of the VLC coded coefficients. There is no need to transmit such a sequence of trailing “1”s. If the VLC decoder runs out of bits before decoding all the coefficients it may infer that the missing bits were all “1”. In such a circumstance VLC coding of the coefficients may be terminated early if all the remaining coefficients are zero. This technique, called “early termination” can, on occasion, provide a significant contribution to coding efficiency and is simple to implement in practice. Early termination may be considered to serve a similar function to zigzag scanning and run length coding in conventional block transform codecs.

If the VLC encoder takes advantage of early termination then the VLC decoder cannot infer the number of coded bits. Therefore the number of coded bits must be directly included in the coded slice, as illustrated above. This is only necessary for the luma coefficients because the total length of the slices is known, a priori, from the information transmitted in the header. An alternative implementation might have been to code the chroma components separately, rather than multiplexed together. But since the magnitudes of the chroma coefficients tend to be similar it saves a few bits not to have to send the number of bits needed for the chroma components.

The packaging of quantised wavelet coefficients into each coded slice is straightforward, and the use of early termination improves coding efficiency.

3.15 Codec Algorithm Summary

The VC-2 LD codec has a number of less conventional features that are highlighted in the algorithmic overview above.

VC-2 LD uses the discrete wavelet transform implemented in integer arithmetic using the lifting implementation. Additional bits of precision are used to prevent low frequencies aliasing into high frequency subbands. These additional bits of precision are intended to prevent subjectively disturbing temporal flicker or flutter artefacts that can otherwise result, even at very low compression factors. Unconventional nearest sample boundary extension is applied at each stage of the lifting implementation, rather than the conventional symmetric periodic extension. This not only simplifies the codec implementation but also improves compression efficiency.

Prediction of the DC coefficients (only) is used to maintain compression efficiency even with ultra low latency.

To achieve low latency VC-2 LD partitions the wavelet coefficients into slices, rather than the more conventional technique of partitioning the original picture into blocks or tiles. Partitioning into slices precludes the possibility of introducing subjectively damaging blocking artefacts at higher compression ratios.

The codec uses a simple dead zone quantiser. However the details of the quantiser and inverse quantiser have been carefully designed so that re-quantising with the same quantising index is transparent, supporting the possibility of lossless re-coding. A quantisation matrix is used to equalise the contribution to the decoded quantisation noise from each subband. The quantisation matrix is applied subtractively, rather than the superficially more obvious additive way, to ensure that lossless coding is possible when bit rate is available.

The deterministic (interleaved) exp-Golomb code is used for entropy coding. Although this is not an optimal VLC, in terms of compression efficiency, it has been used to provide simplicity and flexibility. This variable length code is used in conjunction with “early termination, the VC-2 LD equivalent of zigzag scanning and run length coding, to provide greater compression efficiency.

VC-2 LD eschews the use of a buffer to smooth a variable rate stream. This avoids using negative feedback from buffer fullness to control the quantising index. Instead VC-2 LD uses spatially varying quantisation and deterministically constant bit rate coding. The former implicitly implements texture masking of the quantisation residual to improve subjective quality, even though this is not reflected in PSNR measurements. By omitting a buffer VC-2 LD can achieve an extremely low latency.

4 System Features

This section discusses features of the VC-2 LD codec and, particularly, their utility for production and archive applications. It is unusual in that it has a flexible set of coding parameters, which allows it to be configured for a variety of applications. The previous section described the algorithms used in the codec but it did not indicate the range of codec parameters and how they would be used. Whilst these parameters are enumerated in the standard there is little indication of their intended application. Although the experimental results described in this paper relate to the re-use of legacy infrastructure many other applications are also possible. This section highlights the features of the codec which may be of use in other video production applications.

The section starts by considering features of the codec as a whole. These features typically relate to a set of algorithmic features as they are combined in this codec design. Combined they produce a useful characteristic of the codec. Secondly the section discusses the wide range of video formats supported by VC-2. Finally it describes the parameters of the wavelet transform that can be used to trade of different aspects of codec performance to meet the needs of specific applications.

4.1 High Quality

High quality is achieved through the use of low compression ratios (<8:1, and preferably <4:1) or, equivalently, high bit rates. TV programme producers require high quality to allow headroom for video processing, particularly compression, so that the final programme is low noise. Low noise is important as it directly affects the required final distribution bit rate and thereby the cost of distribution. In addition high quality (low noise) is important when programmes are archived for future re-use.

4.2 Ultra Low Latency

Low latency can be very important in producing television programmes, particularly live events such as sport. Often a signal will pass through multiple stages of compression and decompression. If each stage only adds one or two frames of latency (which is sometimes described as “low latency”) then, with repeated coding and decoding, the overall latency quickly build up to a substantial fraction of a second. Such multi-frame delays make it difficult to synchronise audio and video from multiple sources, greatly complicating programme production.

One commercial implementation of VC-2 has a latency of only a few hundred microseconds. In the broadcast world this is sometimes described as “ultra low latency”. In contrast JPEG2000 typically has significantly higher latency (typically >40ms) or requires that the image be divided into tiles. Tiling in JPEG200 re-introduces the possibility of block artefacts which are absent by design in VC-2.

4.3 Low Complexity

The low complexity achieved by VC-2 LD is important in producing low cost, low power, small size, and simple software (and hardware) implementations that are amenable to optimisation.

4.4 Low Concatenated Coding Loss

During the video production process video may be decoded and recoded many times. For example this typically happens whenever video is edited, which may happen several times, and when it is recorded on video tape or transmitted over a compressed link. Each time video is recompressed quality may be lost. This is one reason why high pixel precision is needed: to provide headroom to mitigate cascaded coding losses. VC-2 LD is designed to minimise recoding losses. Figure 15 shows recoding loss for two practical implementations of VC-2 LD.

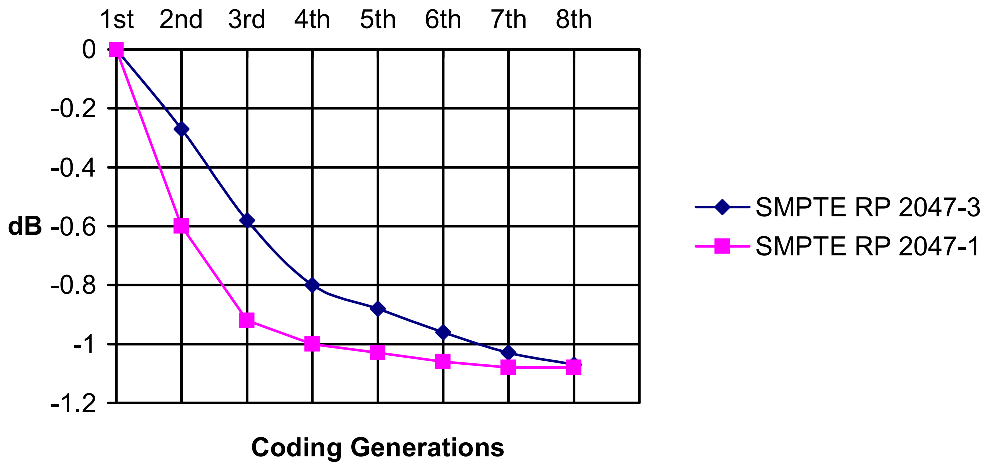

Figure 15: Multi-generation coding loss for a simple encoder

This chart shows the result for a non-optimal hardware implementation. In fact, as described above, with an optimised coder the recoding loss could be reduced further to almost zero.

4.5 Royalty Free Open technology

The issue of patents and royalty payments is a practical issue, rather than technical issue, but none the less important for that. Not only do royalty payments lead directly to increased costs but they also tend to give rise to proprietary implementation to try to avoid patents. Such proprietary implementations not only lead, indirectly, to increased costs (through unnecessary development costs) but also lead to incompatible equipment. TV programme makers don't like proprietary equipment because it reduces competition and, importantly, security of supply. So ideally standards would be patent free. VC-2 is believed to be free from patents and so may be used freely without royalty payments. This has been achieved in part simply as a result of the simplicity of the codec and partly through careful choice of non-patented algorithms. Now that VC-2 is a published international standard any further essential patents are blocked as a result of prior publication.

4.6 Video Format Support

VC-2 is intended to have broad support for the wide range of formats used in video production. It supports unrestricted image size, which means it can be used for next generation TV systems such as Ultra High Definition Television (UHDTV) (specified as 3840x2160 or 7680x4320 in SMPTE 2036-1). It also supports arbitrary frame rates. That is, neither the image size nor the frame rate are restricted to a small set of predefined values, as is typical in other codecs. Instead they can be set by the user. Appropriate values are set by default when a base video format is chosen, to simplify codec use, but these may then be overridden with arbitrary values if desired.

A range of pixel aspect ratios are in common use in video production. Unrestricted pixel aspect ratios may also be coded in the byte stream. This ensures that the decoded signal may be combined, in the correct aspect ratio, with other programme components during production.

An important feature of VC-2 for video production is its support for unrestricted pixel precision (bit depth). Video production is moving towards higher bit depths, and this is supported by VC-2. For distribution codecs 8 bit video precision may be sufficient and many distribution codecs are targeted at this. However in video and movie production much higher precision is sometimes required. Professional video is usually at least 10 bits and the trend is for this to increase to 12 bits and beyond. Higher precision still is required in movie production where 16 bits is not uncommon. Supporting higher bit depths provides greater flexibility, through increased headroom, in post-production processes such as adjusting colour balance. VC-2 is unusual in allowing unrestricted bit depth. This is possible because of its integer implementation and its logarithmically increasing quantiser step sizes.

VC-2 supports a range of chroma resolutions. It is primarily aimed at, although not limited to, coding video luma and colour difference signals (Y′CbCr). It supports the 4:4:4, 4:2:2 and 4:2:0 formats common in video production[4]. R′G′B′, rather than Y′CbCr, is typically used for movie production or for TV dramas shot as a movie. However compression efficiency is improved by performing a colour transformation. In addition to the usual video, Y′CbCr, colour transformations VC-2 also supports YCoCg [38]. This allows it to support R′G′B′ through a reversible colour transformation in the same way as the JPEG2000, JPEG XR and H264 codecs.

An important codec feature for archiving and video production is support for interlaced scanning, which is a vertical temporal quincunx sampling lattice[5]. Interlaced scanning is, essentially, an analogue video compression technique dating from the beginning of analogue television services. Despite the powerful arguments against it, interlacing continues to be supported by the television standards organizations and is still included in digital video transmission formats such as DVB, and ATSC. Moreover most video, including HD, is still produced on an interlaced lattice. So, despite MPEG's newest, High Efficiency Video Codec (HEVC) omitting support for interlace, it is still an important feature in a production codec.

VC-2 supports interlaced sampling. The options for coding interlaced frames are limited in an intra-frame wavelet codec. VC-2 takes the simplest approach of coding fields as independent pictures. The presence of vertical/temporal aliasing does reduce coding efficiency. However this approach allows ultra low latency coding that other, more complex, approaches would preclude.

4.7 Algorithmic Flexibility (Wavelet Features)

VC-2 offers a selection of wavelet kernels for different applications. These range from the simple Haar kernel to the more complex Daubechies 9,7 kernel. The choice of kernels is provided to support a trade off between complexity and other aspects of codec performance such as latency and compression efficiency. All kernels are specified as a lifting implementation using integer arithmetic.

The Haar kernel is least complex but also offers the lowest compression efficiency. It does, however, provide the further advantage that it is the only possible wavelet that supports a block transform implementation, which can simplify (particularly) hardware implementations. The Haar kernel also provides the choice of adding extra precision bits or not. For lossy coding it is generally wise to use additional precision bits. However for lossless coding the issue of low frequencies aliasing to high frequencies is not important and avoiding additional precision bits improves compression efficiency. So the Haar kernel with no shift is intended for lossless coding (and therefore is less relevant to the Low Delay profile described here).

The Fidelity filter was specifically developed for VC-2 using an exhaustive search process to select a filter set with simple integer filter coefficients, limited to 8-bit resolution, and offering good alias rejection in both analysis and synthesis operations. The resulting filter is almost mirror symmetric. It provides good anti-aliasing characteristics, which minimise aliasing by placing the "update" stage before the "predict" stage. This kernel is useful in circumstances where a low resolution "proxy" is required as part of a video production workflow. JPEG2000 offers a similar facility. However without a kernel specifically designed to minimise aliasing the quality of the proxy image is often compromised by the presence of aliasing. The disadvantage of the Fidelity kernel is that it is more complex. It uses higher precision coefficients (although still limited to 8 bits) and more update and prediction stages in the lifting implementation (leading to higher latency). Furthermore its compression performance is (slightly) lower than some other kernels.

[4] In 4:4:4 all components have the same spatial resolution, in 4:2:2 the chroma resolution is reduced by a factor of two horizontally, and in 4:2:0 it is reduced by a factor of two vertically as well.

[5] An alternative view of interlace is that consecutive pictures are vertically subsampled 2:1 to create interleaved "fields", with subsampling on alternate phases for consecutive pictures, and two consecutive fields are grouped to form a video "frame".

The Daubechies 9,7 kernel is the most complex kernel available in VC-2. It uses 4 lifting stages and 13 bit coefficients. It has the highest latency. It is included because this kernel is a popular kernel often used as a benchmark for wavelet codec performance.

In addition to selecting the wavelet kernel the user can also select the wavelet depth. Again this offers a trade off between complexity, latency and compression efficiency. In principle VC-2 offers unconstrained wavelet depth. Initially compression efficiency increases with wavelet depth (at the expense of complexity and latency). In practice, however, there is little gain in increasing wavelet depth beyond a relatively small number of levels. The optimum wavelet depth from a compression efficiency perspective varies with kernel and image content and format.

5 Experimental Proceedure

This paper now describes experiments to characterise the compression performance of VC-2 LD. A wide range of coding parameters are possible, which lead to different trade-offs between compression efficiency, quality, complexity and latency. The approach taken here is to measure rate distortion performance for two codec configurations that are commercially available. The two examples chosen are those specified in SMPTE RP 2047 parts 1 and 3. These relate to the reuse of legacy broadcast infrastructure to carry higher resolution video. Specifically, one set of coding parameters (SMPTE 2074-1) have been used for transport of 1080 line progressive video over a channel designed for interlaced video at a compression ratio of 5:2. The other set (SMPTE 2047-3) is used for transport of HDTV over a standard definition link. The results below show rate distortion curves, that is PSNR versus bit rate, for these sets of coding parameters. They also show the difference between coding interlaced and progressive content using interlaced and progressive coding modes.

These experiments were conducted using high definition reference material available from the European Broadcasting Union (EBU). The test sequences used were the “HIPS” test sequences produced by the EBU [39] and SVT test sequences available from VQEG (and the EBU) [40]. The material was chosen to provide a variety of video content (e.g. “clean” and “noisy” with low and high motion, varying depth of field etc). Since the VC-2 codec is a video production codec the experiments used the 10 bit Y′CbCr 4:2:2 format typical of video production. The source format was 1080 line progressive, although this was filtered for experiments on interlaced coding (see below).

PSNR, defined below, is used as the quality metric.

$$PSNR \equiv 10.\log_{10}\left(\frac{Max^2}{MSE}\right) = 20.\log_{10}\left(\frac{Max}{\sqrt{MSE}}\right) \qquad \textbf{Equation 11}$$

Whilst it is well known that PSNR is not an ideal metric [41], with a number of alternatives [42], it is used because it is simple and sufficient for the purpose. Reiter et al [43] say “Our study has revealed that in a scenario with fixed content and different distortion types (source versus channel distortion), PSNR is actually capable of comparing the relative quality more accurately than more advanced metrics performing better in the more generic scenarios. This should not be interpreted as evidence of a good performance of PSNR, but rather as evidence that even the more advanced metrics have their weak points”. However, with careful use PSNR can yield accurate results. Huynh-Thu & Ghanbari comment in [44]; “We have shown that PSNR can be used as a good indicator of the variation of the video quality when the content and codec are fixed across the test conditions, e.g. in comparing codec optimisation settings for a given video content. On the other hand, we have shown that PSNR is an unreliable video quality metric when different contents are considered in the test conditions.”

Taking into account the limitations of PSNR as a metric this paper provides comparative analyses which use the same test content. In these experiments we have analysed a range of content, not just individual sequences, and provided both the mean and standard deviation of the distortion. We have noticed that, given a wide range of content, PSNR figures tend to exhibit a roughly Gaussian distribution. It is not within the scope of this paper to discuss this further. However, by providing mean and standard deviation of the distortion measurements we hope to minimise the impact of content on the analysis.

Software was written to implement VC-2 LD compression. It searched for the smallest quantiser index for which the quantised data fitted within the space available in a slice. Other algorithms are possible, indeed are necessary for a hardware implementation, but this algorithm was felt to yield the best performance, at least for 1st generation encoding[6].

Since VC-2 is an intra-frame codec, and many of the frames in the video sequence were very similar, the eleven EBU HIPS test sequences were temporally subsampled to yield a total of 220 frames and the eight SVT test sequences[7] were subsampled to a total of 153 frames. This provided a wide range of results from which to measure the mean and standard deviation of the PSNR measurements.

Distortion was measured for 11 compression factors. These varied from 1.4:1 to 8:1, by factors of $2^{¼}$, which covers the useful compression range of VC-2 LD. Most other papers present results with distortion versus bit rate, compression ratio or bits per pixel. This paper presents distortion versus bits per pixels as this is independent of image format, frame rate and bit depth (dynamic range). Nevertheless, for practical reasons, it is sometimes useful to know the compression ratio or the equivalent bit rate for a particular image format. For this reason table 3 shows the equivalent compression ratio and bit rate for 1920x1080 HD video at 25 frames/s using 4:2:2 colour subsampling. This is the standard high definition video signal in European countries.

Bits per pixel	Compression Factor	Bytes per Frame	Bit Rate / Mbit/s
20	1	5184000	1037
14.14	1.41	3665642	733
11.89	1.68	3082425	616
10.00	2	2592000	518
8.41	2.38	2179604	436
7.07	2.83	1832821	367
5.95	3.36	1541212	308
5	4	1296000	259
4.20	4.76	1089802	218
3.54	5.66	916410	183
2.97	6.73	770606	154
2.5	8	648000	129

Table 3: Equivalent measures of data rate

Compression performance was measured using both progressive and interlaced coding modes for the SMPTE 2047-3 set of coding parameters. However, since the source material was originally progressive, this was, perhaps, an unrealistically harsh test of the interlaced coding mode. To provide a more realistic test, the source material was also vertically filtered using an FIR filter with coefficients (¼, ½, ¼). This simulated the vertical filtering that would occur in a practical interlaced video camera. The results for both the original progressive and the filtered “interlaced” sequences are presented.

6 Results & Discussion

Experiments were performed with two sets of coding parameters corresponding to two recommended practices published by the SMPTE.

[6] Alternative coding strategies, which directly minimise the coding error, may be preferable for recoding previously coded content (as is pointed out in the standard). However the results here relate to first generation coding.

[7] In pursuit of using as wide a range of material as practicable, the “SVT” sequences used here included the EBU “Dance” and “Goal” sequences in addition to the usual “Crowd Run, “Ducks Take Off”, “Flags”, “Into Tree”, “Old Town” and “Park Joy” sequences.

A first set of results were obtained based on the coding parameters of SMPTE RP 2047-1 for coding 1080 line progressive, 50 frame/s (or 60 in the USA) HDTV (1080p50), for carriage over a 1080 line, 25 frame/s interlaced link. For that application a compression ratio of 5:2 was used.

Figure 16 shows the mean PSNR figures for the luma and colour difference channels. In this experiment a range of compression factors were used to explore the shape of the rate distortion curve. For this application (SMPTE RP 2047-1) low latency and low complexity are critical. So the compression uses the simplest Haar wavelet kernel (with one bit of shift for additional coefficient precision) in a 2 level wavelet transform. The slice region was 4 lines by 16 pixels, so each slice contained 4 DC luma coefficients and 2 DC chroma coefficients for each colour difference channels (i.e. 128 luma and chroma coefficients in total).

The error bars shown in figure 16 are for the luma channel (Y′). This corresponds, not to measurement error in the usual sense, but rather to the variation in PSNR across the 220 frames (corresponding to 11 sequences) that were measured. Typically the standard deviation of the PSNR was about 2dB, indicating that PSNR could vary substantially, by 10dB or more, from the easiest to the most difficult to code images. This large variation is typical of video codecs in general. Whilst many reports provide representative PSNR curves for a few well known test sequences, it seems more appropriate to report results as mean and standard deviation of the measured PSNR over an ensemble of varied sequences. As will be seen below this appears to provide a reliable characterisation of the codec even when the set of test sequences is changed. In this way the precise choice of test sequences become less important.

It is not obvious how to report codec distortion, even having decided to use PSNR, because each component exhibits different PSNR figures. Figure 16 shows the PSNR curves for all three colour components. Possible distortion metrics based on PSNR include weighted averages of PSNR for Y′, Cb and Cr components, or weighed averages having converted the video back to the (4:4:4) R′G′B′ domain, and a plethora of other possibilities. Whilst there is some variation in the compression efficiency for the 3 components they are actually quite similar, with the variation between components being less than that between sequences. The other results reported in this paper similarly show a close correlation between the PSNR figures for the different colour components. So, for the remainder of this paper, only the PSNR results for the luma component are presented. This is simplest, avoids clutter from reporting too many results, and corresponds to reporting practice in other papers on video compression.

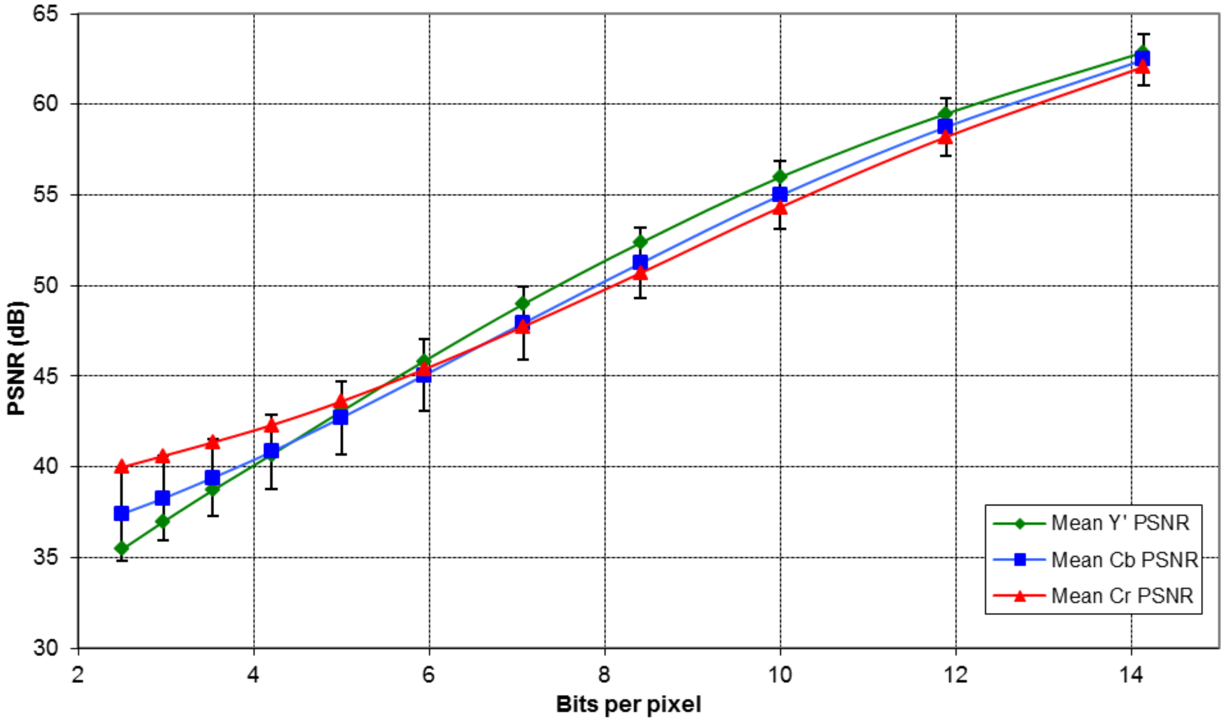

Figure 16: Luma and chroma coding distortion using SMPTE RP 2047-1 coding parameters

Figure 17 shows the compression performance of VC-2 LD with coding parameters intended for compression of 1080 line progressive over a 1080 line interlaced channel (SMPTE 2047-1, described above). The figure shows performance for the two sets of sequences tested (also described above). The PSNR curve for the EBU sequences is the same as in figure 16 but, without the clutter of the results for the colour difference channels, the performance may be more clearly seen. Although there is a difference between these two sets of sequences it is relatively small and less than the variation within each set of sequences. The close correspondence between the two sets of sequences seems to justify the measurement of mean PSNR and its standard deviation, across an ensemble of sequences, as a good measure of codec performance.

Another application, defined in SMPTE 2047-3, is for transport of high definition over a channel intended for uncompressed standard definition video. The PSNR curve is shown in figure 18, (which is for compression of progressive video). The previous example required a lower compression ratio and low latency, so the simplest set of coding parameters were used. This second application requires a higher compression ratio (about 5:1) so coding parameters are chosen to try to achieve better compression. This example uses a 3 level wavelet transform with the LeGall wavelet kernel. This kernel provides better separation between low and high frequencies but requires more latency (although still very low) and a little more computational complexity. The slice region in this example is a little larger corresponding to 8 lines by 16 pixels, so each slice contains 2 DC luma coefficients and only a single DC chroma coefficient for each colour difference signal (i.e. 256 luma and chroma coefficients in total).

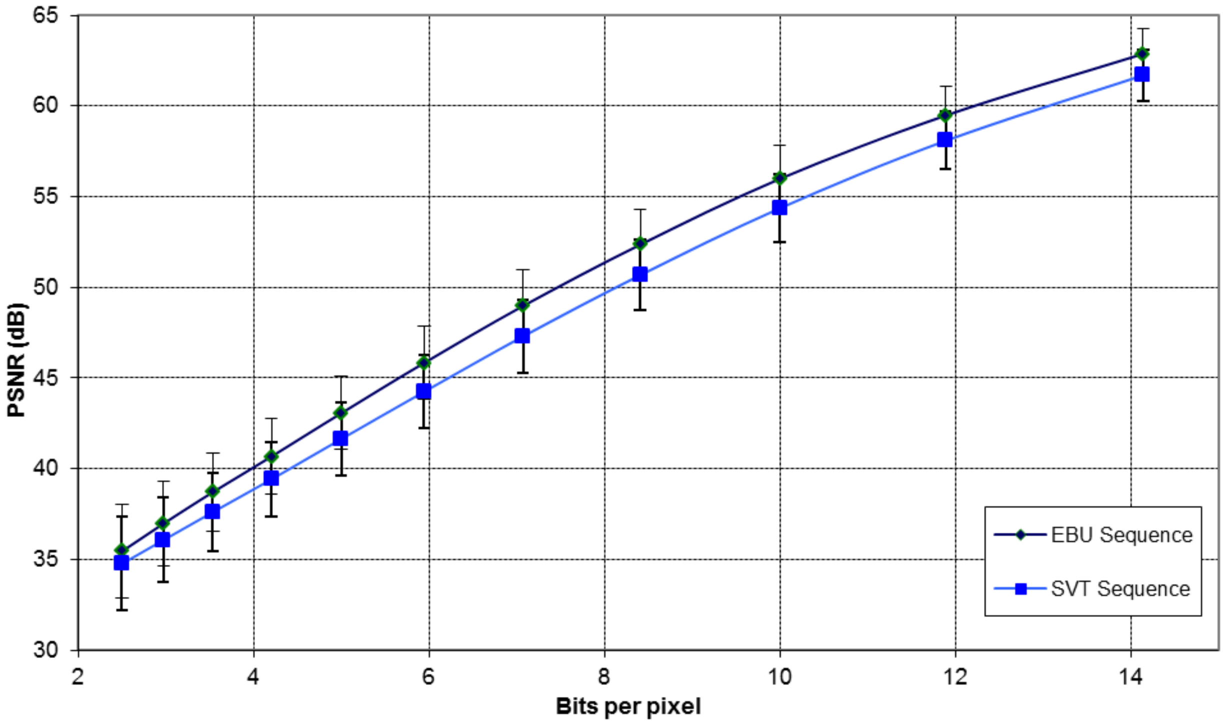

Figure 17: Luma coding distortion using SMPTE RP 2047-1 parameters

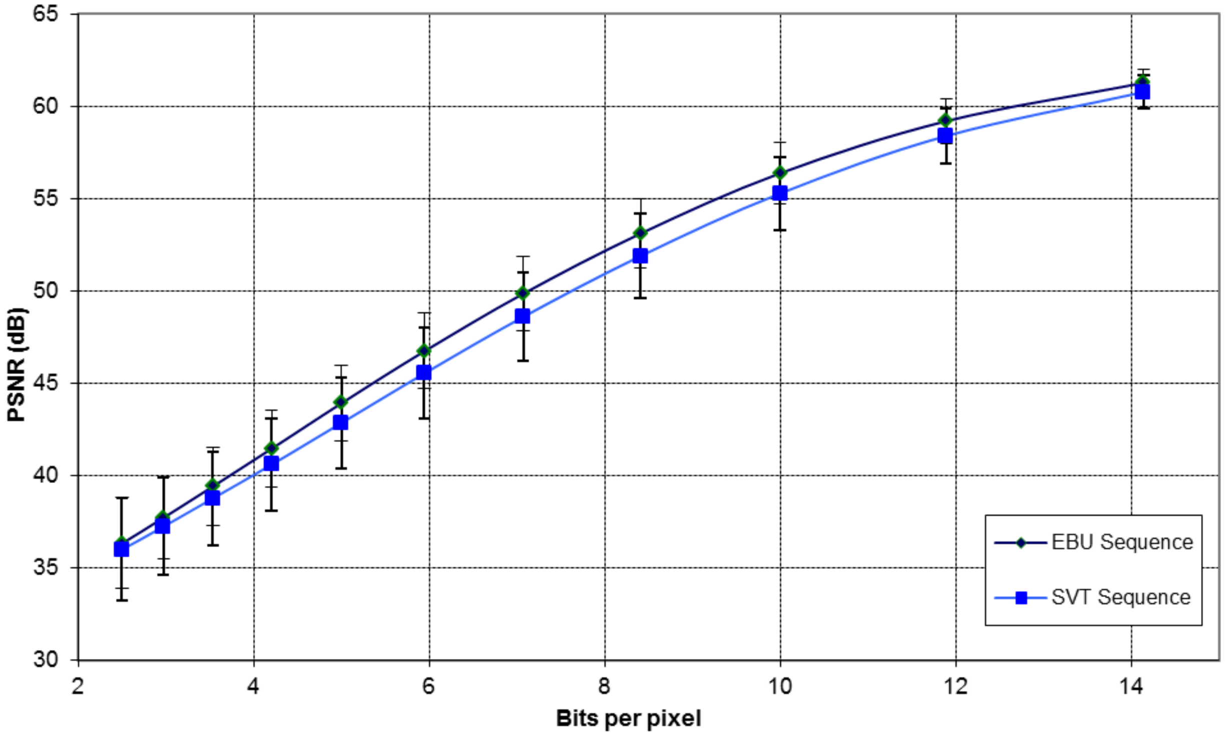

Figure 18: Luma coding distortion of progressive sequences using SMPTE RP 2047-3 parameters

In spite of the difference in the wavelet transform, and the corresponding difference in the slice size, the coding performance of these two sets of coding parameters is surprisingly similar. Figure 19 shows the difference between the mean PSNR (including both EBU and SVT sequences) for the SMPTE RP 2047-1 and 2047-3 sets of coding parameters. With fewer bits per pixel the more complex 3 level LeGall transform produces about 1dB of coding gain. This gain is relatively small, being significantly less than the difference in coding performance between sequences. With more bits per pixel, approaching lossless compression, the simpler 2 level Haar transform actually performed better with a 1dB coding gain at 14 bits/pixel.

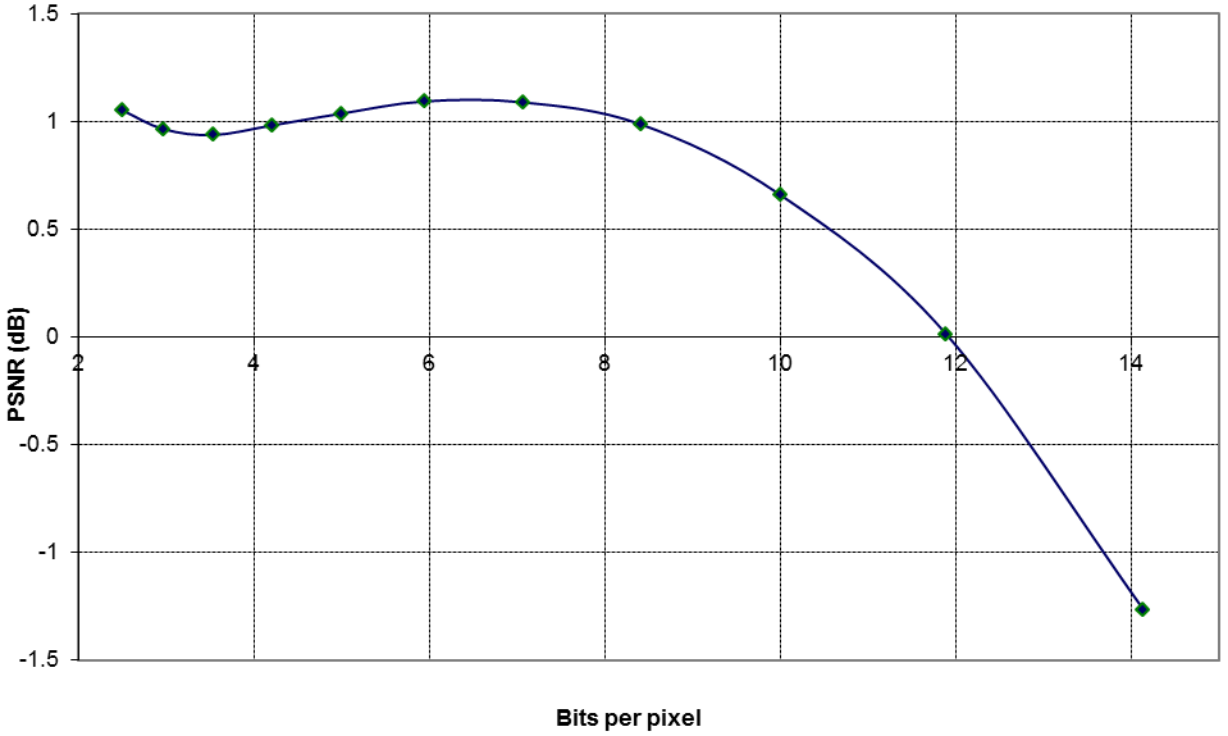

Figure 19: Difference in luma coding performance SMPTE 2047-3 less SMPTE 2047-1

Codec performance for production and intermediate codecs should not be judged solely on PSNR figures. Nevertheless it is useful to provide a comparison with other codecs to see where VC-2 LD fits in. Unfortunately it is extremely difficult to find comparable PSNR figures for a range of proprietary codecs. The most convincing figures the author has found, albeit from a non peer reviewed document, are from [4], which are summarised in table 4. It appears that these used the same progressive SVT test sequences from the EBU. They are broadly in agreement with independent measurements taken by a colleague of the author. Given the difficulty of finding comparable performance figures these should not be taken as definitive, rather they provide an indication of the performance of leading production or intermediate codecs. The figures for bits/pixel are only approximate as the PSNR is specified at a certain bit rate; they are provided to facilitate comparison with the VC-2 LD results presented in this paper.

Codec	Bit Rate (Mbit/s)	Bits/pixel	PSNR
AVC-Intra	100	2	37.4
ProRes 422	147	3	39.1
	220	4	42.7
HQX	145	3	38.9
	224	4	41.8
	400	8	46.4
DNxHD	145	3	38.6
	220	4	41.4

Figure 20: Coding distortion of alternative production codecs

One thing to note is that the PSNR performance of all these codecs is remarkably similar at the same bit rate. At about 220Mbit/s, about 4 bits/pixel, VC-2 LD has broadly similar performance to all these codecs. At low compression ratios and bit rates of about 400Mbit/s, or 8 bits/pixel, VC-2 LD significantly exceeds the performance of Grass Valley's HQX codec (the only PSNR specified at this bit rate). This is a noteworthy achievement for VC-2 LD bearing mind its low complexity. At lower bit rates, around 150Mbit/s, or 3 bits per pixel, VC-2 LD's PSNR is about 2dB less than other codecs. This is probably a consequence of its simple entropy coding.

Most HD signals in the UK are produced either as 25 frames/s progressive (film) or 50 fields/s interlaced. The results in figure 21 show coding performance for progressive scanned images. In practice even progressive HD signals are formatted as "progressive segmented frame" (psf)[8] to be compatible with interlaced video. So, in the applications above, the encoder should really use the interlaced coding mode, in which fields are coded separately, rather than the progressive mode of coding the frame as a whole. When field coding is used, with the SMPTE RP 2047-3 coding parameters, the PSNR curve is as illustrated in figure 21. The effect of coding fields rather than frames is to reduce the PSNR by about 1dB.

[8] In progressive segmented frame the every other line, e.g. an even line, is transmitted first, followed by the alternate lines, e.g. the odd lines, so that the signal is formatted as an interlaced frame would be.

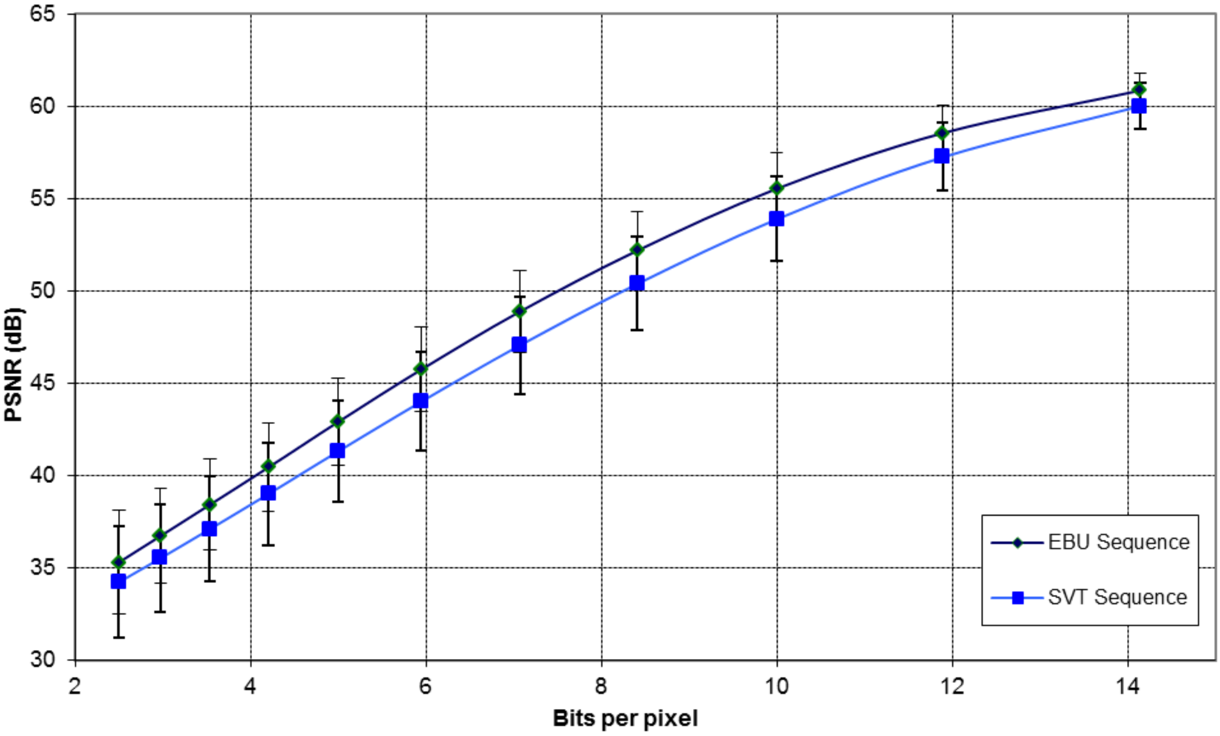

Figure 21: Luma coding distortion of progressive sequences using field coding

In practice true interlaced fields, from an interlaced camera, contain less high vertical frequency energy than results from simply sub-sampling progressive frames. This is because some vertical filtering is applied in the camera. Alternative test sequences were derived by simulating this in-camera filtering, using a vertical FIR filter with coefficients (¼, ½, ¼). The results from compressing this “pseudo” interlaced video are shown in figure 22. Here the PSNR figures are relative to the vertically filtered video, rather than the original progressive video. The distortion from field coding this pseudo interlaced signal is 2 to 3dB better than field coding true progressive video. This is due to the reduction in high vertical frequency energy due to filtering.

Overall the effect of field coding, compared to frame coding, is modest. A small reduction in PSNR, of about 1dB, results from field coding progressive signals. However for true interlaced video this is (more than) compensated for by having less high vertical frequency energy. In either case the change due to coding mode is significantly less than the variation between sequences.

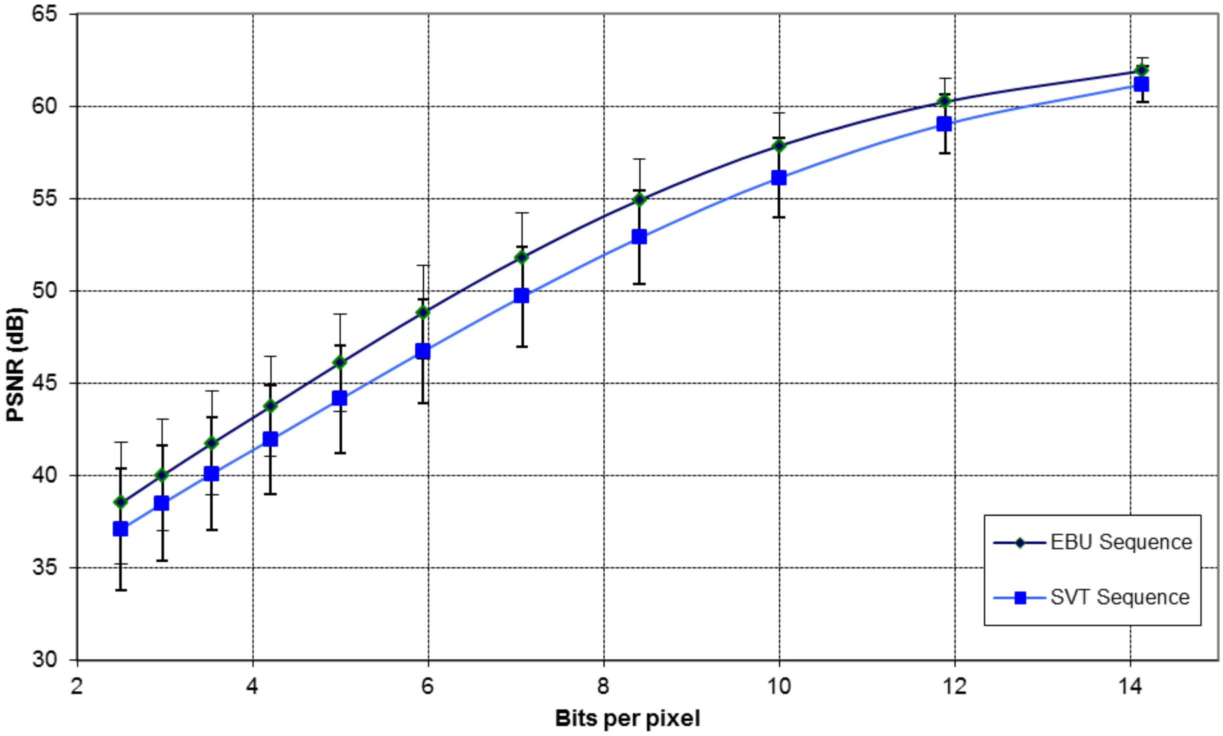

Figure 22: Luma coding distortion of progressive sequences using frame coding

7 Summary & Conclusions

This paper has provided an overview, and some experimental measurements, for the Low Delay profile of the VC-2 codec (SMPTE 2042). It started by discussing the need for “production” or “intermediate” codecs, used in video production, and noting their different requirements to “emission” or “direct to home” codecs. Initially it highlighted the key features, including low complexity, ultra-low latency and strictly constant bit rate, which set VC-2 LD apart. It proceeded to discuss the design principles applied to the codec design. It then moved on to describing the algorithms used and highlighted the unusual and innovative aspects of the codec.

Although the broad features of the codec are familiar from other codecs, many building blocks include their own innovative features that improve performance, increase flexibility or reduce complexity. For example the wavelet transform itself is a reversible integer transform in which care is taken to ensure adequate arithmetic precision. Sufficient arithmetic precision is important to avoid the subjectively disturbing artefact of large area flutter (even though this has a small effect on PSNR). The wavelet transform uses an innovative boundary extension technique that not only simplifies the implementation but also improves coding performance at the edge of pictures. The quantiser matrix is used in an innovative way to ensure that regions of the picture can be losslessly coded when there is sufficient data capacity to do so. Entropy coding uses a simple deterministic variable length code that is optimised for implementation. Although this does not provide the theoretically optimum coding performance it is simple and supports a wide range of bit rates and coding parameters. This flexibility allows VC-2 LD to be used in a wide range of applications.

The algorithmic overview is followed by a discussion of the features of the codec, particularly their utility for production and archive applications. These include support for a wide range of video formats (including a wide range of bit depths), support for interlace, royalty free open technology, and support for low loss multi-generation coding.

The performance of the codec was measured experimentally for two sets of coding parameters used in commercially available hardware. Care was taken to use a wide range of video sequences to ensure that the measurements were representative. In spite of its limitations, discussed at some length, the PSNR metric was used a measure of coding distortion. Typically codec performance is reported for a selection of well known test sequences. It was felt that a more representative measurement was the mean and standard deviation of the PSNR over an ensemble of sequences.

The experimental results support the validity of this approach by producing very similar, albeit not identical, results for two significantly different ensembles of sequences.

Performance of VC-2 LD, purely in terms of compression efficiency measured using PSNR, is broadly comparable with other production and intermediate codecs. At higher compression ratios (8:1) its performance is slightly worse, but at lower compression ratios (2:1) its performance is better. However, as stressed above, subjective performance is not synonymous with PSNR measurements. VC-2 LD provides two features which improve its subjective performance. First care is taken to ensure that sufficient arithmetic precision is used to prevent low frequencies aliasing to high frequencies. This prevents low frequencies being distorted by the quantisation process, and thereby prevents a disturbing low frequency flutter artefact. VC-2 LD also takes advantage of texture masking to improve subjective performance.

In addition to good compression performance VC-2 LD also has the advantages of low complexity and offers ultra low latency, which other codecs cannot match. As a “parameterised” codec it provides a range of coding parameters, enabling it of support a wide range of applications. It is also unusual in providing a precisely constant bit rate by design. This supports its use over fixed bit rate links such as those available in the re-use of legacy infrastructure.

8 References

1. M. Ghanbari. 2011. Standard Codecs, Institution of Engineering and Technology; 3rd edition (4 Mar 2011). ISBN-10: 086341964X
2. Apple ProRes White Paper 2009, http://images.apple.com/support/finalcutpro/docs/Apple-ProRes-White-Paper-July-2009.pdf
3. Apple ProRes, From MultimediaWiki, http://wiki.multimedia.cx/index.php?title=Apple_ProRes
4. Akira Takemoto, Dec 2010. HQX: Grass Valley’s Intermediate Codec White Paper, www.grassvalley.com/docs/WhitePapers/professional/GV-4097M_HQX_Whitepaper.pdf
5. SMPTE ST 2042-1-2009. VC-2 Video Compression.
6. SMPTE ST 2042-2-2009. VC-2 Level Definitions.
7. SMPTE RP 2042-3-2010. VC-2 Conformance Specification.
8. SMPTE RP 2047-1-2009. VC-2 Mezzanine Level Compression of 1080P High Definition Video Sources.
9. SMPTE ST 2047-2-2010. Carriage of VC-2 Compressed Video over HD-SDIs.
10. SMPTE RP 2047-3:2011 VC-2 Level 65 Compression of High Definition Video Sources for Use with a Standard Definition Infrastructure.
11. SMPTE ST 2047-4-2011. Carriage of Level 65 VC-2 Compressed Video over the SDTV SDI.
12. ISO/IEC 15444-1:2004 - Information technology -- JPEG 2000 image coding system: Core coding system.
13. JPEG2000 David Taubman (Editor), Michael Marcellin (Editor), 2002. JPEG2000: Image Compression Fundamentals, Standards and Practice. Springer. ISBN-10: 079237519X, ISBN-13: 978-0792375197.
14. www.mpeg.org
15. Moving Picture Experts Group. Wikipedia http://en.wikipedia.org/wiki/Moving_Picture_Experts_Group
16. Video Coding Experts Group. Wikipedia http://en.wikipedia.org/wiki/Video_Coding_Experts_Group
17. http://www.mpegla.com

18. MPEG LA. Wikipedia. http://en.wikipedia.org/wiki/MPEG_LA

19. Zixiang Xiong, Kannan Ramchandran, Michael T. Orchard, and Ya-Qin Zhang, A Comparative Study of DCT- and Wavelet-Based Image Coding, August 1999, IEEE Transactions On Circuits And Systems For Video Technology, Vol. 9, No. 5, pp692-695

20. Ingrid Daubechies and Wim Sweldens, 1998, Factoring wavelet transforms into lifting steps, Journal of Fourier Analysis and Applications, Volume 4, Number 3, 247-269

21. Martin Vetterli, Jelena Kovacevic and Vivek K Goyal, 2001, Fourier and Wavelet Signal Processing. http://fourierandwavelets.org

22. W. Sweldens. The lifting scheme: A custom-design construction of biorthogonal wavelets. Journal. Applied and Computational Harmonic Analysis, 1996.

23. Wim Sweldens, "Lifting scheme: a new philosophy in biorthogonal wavelet constructions", Proc. SPIE 2569, 68 (1995); http://dx.doi.org/10.1117/12.217619

24. A. Jensen & A. la Cour-Harbo. 2001. Ripples in Mathematics: The Discrete Wavelet Transform. Springer. ISBN-10: 3540416625, ISBN-13: 978-3540416623

25. 28. M. Vetterli and J. Kovacevic. Wavelets and Subband Coding. Signal Processing. PrenticeHall, Englewood Cliffs, NJ, 1995. http://waveletsandsubbandcoding.org/

26. Claypoole, R.; Davis, G.; Sweldens, W.; Baraniuk, R.; 1997 Nonlinear wavelet transforms for image coding. Signals, Systems & Computers, 1997. Conference Record of the Thirty-First Asilomar Conference on, vol. 1, pp. 662-667

27. R. L. Claypoole, R. G. Baraniuk, and R. D. Nowak, 1998. Lifting Construction of Non-Linear Wavelet Transforms. IEEE-SP International Symposium on Time-frequency and Time-scale Analysis. pp 49-52.

28. Claypoole, R.L.; Davis, G.M.; Sweldens, W.; Baraniuk, R.G.; 2003. Nonlinear wavelet transforms for image coding via lifting. Image Processing, IEEE Transactions on, Volume 12. Issue 12, pp 1449-1459.

29. C. Fogg , D. J. LeGall, J. L. Mitchell, W. B. Pennebaker. 1996 MPEG Video Compression Standard. Springer. ISBN-10: 0412087715, ISBN-13: 978-0412087714

30. 1180-1990. IEEE Standard Specifications for the Implementations of 8x8 Inverse Discrete Cosine Transform.

31. MPEG-2 Specification. Generic Coding of Moving Pictures and Associated Audio Information. Part 2 Video ISO/IEC 13818-2: 1995 (E), Recommendation ITU-T H.262 (1995 E).

32. Tudor, P.N, 1995. MPEG-2 video compression. Electronics & Communication Engineering Journal, vol. 7, Issue 6, pp. 257-264.

33. Brislawn, C.M. 2007. Equivalence of Symmetric Pre-Extension and Lifting Step Extension in the JPEG 2000 Standard. ACSSC 2007. Conference Record of the Forty-First Asilomar Conference on Signals, Systems and Computers 4-7 Nov. 2007, pp 2105-2109

34. M. Gaubatz, D. Chandler, and S. S. Hemami, Spatial quantization via local texture masking, in Proc. SPIE HVEI, Jan. 2005, vol. 5666, pp 95-106

35. Gaubatz, M.; Kwan, S.; Chern, B.; Chandler, D.; Hemami, S.S.; 2006 Spatially-Adaptive Wavelet Image Compression Via Structural Masking. Image Processing, 2006 IEEE International Conference on, 8-11 Oct. 2006, pp 1897-1900

36. S. Winkler and S. Süsstrunk, Visibility of noise in natural images, Proc. IS&T/SPIE Electronic Imaging 2004: Human Vision and Electronic Imaging IX, Vol. 5292, pp. 121-129, 2004.

37. Rimac-Drlje, S., Zagar, D., Martinovic, G., 2009 Spatial Masking and Perceived Video Quality in Multimedia Applications, Systems, Signals and Image Processing, 2009. IWSSIP 2009. 16th International Conference on, pp 1-4

38. H.S. Malvar, G.J. Sullivan. YCoCg-R: A Color Space with RGB Reversibility and Low Dynamic Range, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, Document No.JVT-1014r3, July 2003

39. EBU HIPS Test Sequences. http://tech.ebu.ch/Jahia/site/tech/cache/offonce/news/ebu-shoots-1080p50-test-sequences-at-nrk-04may10

40. SVT Test Sequences ftp://vqeg.its.bldrdoc.gov/HDTV/SVT_MultiFormat/

41. Z. Wang & A. C. Bovik, 2009. Mean Squared Error: Love It or Leave It? A New Look at signal fidelity measures. IEEE Signal Processing magazine, vol. 26, no 1, pp. 98-117 January 2009.

42. Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, 2004. Image quality assessment: From error visibility to structural similarity. IEEE Transactions on Image Processing, vol. 13, no. 4, pp. 600-612, Apr. 2004.

43. 46. Comparing apples and oranges: assessment of the relative video quality in the presence of different types of distortions. Ulrich Reiter, Jari Korhonen and Junyong. EURASIP Journal on Image and Video Processing 2011, 2011:8. http://jivp.eurasipjournals.com/content/2011/1/8

44. Scope of validity of PSNR in image/video quality assessment. Huynh-Thu, Q.; Ghanbari, M. The Institution of Engineering and Technology, Electronics Letters, Volume 44, Issue 13, Pages 800-801.