Louw and Kaneko *EURASIP Journal on Advances in Signal Processing* 2013, **2013**:181
http://asp.eurasipjournals.com/content/2013/1/181

EURASIP Journal on Advances in Signal Processing
a SpringerOpen Journal

RESEARCH Open Access

Suppressing feedback in a distributed video coding system by employing real field codes

Daniel J Louw* and Haruhiko Kaneko

Abstract

Single-view distributed video coding (DVC) is a video compression method that allows for the computational complexity of the system to be shifted from the encoder to the decoder. The reduced encoding complexity makes DVC attractive for use in systems where processing power or energy use at the encoder is constrained, for example, in wireless devices and surveillance systems. One of the biggest challenges in implementing DVC systems is that the required rate must be known at the encoder. The conventional approach is to use a feedback channel from the decoder to control the rate. Feedback channels introduce their own difficulties such as increased latency and buffering requirements, which makes the resultant system unsuitable for some applications. Alternative approaches, which do not employ feedback, suffer from either increased encoder complexity due to performing motion estimation at the encoder, or an inaccurate rate estimate. Inaccurate rate estimates can result in a reduced average rate-distortion performance, as well as unpleasant visual artifacts. In this paper, the authors propose a single-view DVC system that does not require a feedback channel. The consequences of inaccuracies in the rate estimate are addressed by using codes defined over the real field and a decoder employing successive refinement. The result is a codec with performance that is comparable to that of a feedback-based system at low rates without the use of motion estimation at the encoder or a feedback path. The disadvantage of the approach is a reduction in average rate-distortion performance in the high-rate regime for sequences with significant motion.

Keywords: Distributed video coding; Feedback suppression; Real field coding; Compressive sensing; Multi-hypothesis

1 Introduction

In a standard transform-domain distributed video coding (DVC) system, frames are split into key frames (which are similar to I-frames in H.264) and Wyner-Ziv (WZ) frames. The key frames are intra-coded, and the WZ frames are discrete cosine-transformed, quantized and encoded using a systematic error correction code (ECC). The parity symbols from the encoding are then transmitted. At the decoder, advanced motion compensation methods are used to produce an estimate of the frame, also known as the side information (SI), from the intra-coded key frames. Errors in the SI are corrected using the received parity symbols.

The encoder must operate at a high enough rate (transmit enough parity symbols) to ensure that the errors can be corrected. However, calculating this rate at the encoder is fundamentally impossible without calculating the SI used at the decoder. Doing this at the encoder would defeat the purpose of DVC as it would increase the computational complexity of the encoder to the same level as that of conventional video coding. The conventional method used to solve this problem is to introduce a feedback path from the decoder [1]. If the rate estimate is too low, more bits are requested. This requires the video sequence to be decoded in real time and renders the codec unsuitable for any application where the compressed sequence needs to be stored for decompression at a later time. It also introduces latency in the process which can become severe if the decoder is far away from the encoder. Constraining the use of the feedback channel has been shown to alleviate some of these problems [2], though real-time decoding is still required.

The second method used to determine the rate is called suppressed feedback DVC, where a low complexity estimate of the SI is created at the encoder and used to estimate the rate. If the rate estimate is too low, then

*Correspondence: louwdj.eng@gmail.com
Department of Computer Science, Tokyo Institute of Technology, Tokyo 226-8552, Japan

decoding failures can lead to significant distortion in the decoded frame. Conversely, if the rate is overestimated, any extra bits are wasted. Typically, the rate estimate is good for low motion sequences and poor for high or complex motion sequences. In order to achieve robust performance, extra bits may be transmitted to increase the likelihood of successful decoding. However, a finite probability of a decoding failure remains. Increasing the number of extra bits to reduce this probability implies that a larger percentage of transmitted bits will be redundant, yielding reduced rate-distortion (R-D) performance.

In this paper, the authors propose to solve this problem by changing the codec structure so that the output distortion will be a smoother function of the rate. As a result, any extra bits will continue to improve the distortion, and if the rate is underestimated, the image quality will degrade gradually. The rate-distortion characteristics are modified by changing the nature of the quantization region that the Wyner-Ziv code imposes on the signal space. This change is implemented by reversing the order of the quantizer and the low-density parity check (LDPC) encoder and by defining the code over the real field instead of over a finite field.

The results will show that the system reduces the variance of the distortion and improves the perceptual video quality, in comparison with more conventional feedback-free methods without increasing the encoding complexity. The trade-off introduced by the proposed method is a reduction in the average R-D performance as compared to systems with perfect rate knowledge.

This paper is structured as follows: Section 2 provides a short review of the relevant literature. Section 3 discusses real field coding and describes how it alleviates the effects of an imperfect rate estimation. Section 4 describes the overall design of the system. Specific aspects of the encoder and its subsystems are described in Section 5, while the decoder with its subsystems are discussed in Section 6. The complexity of the proposed system is analysed in Section 7 and the performance of the system is analysed in Section 8, with the conclusion following in Section 9.

2 Overview of the previous work

DVC is based on the Wyner-Ziv [3] and Slepian-Wolf (SW)[4] coding theorems. There have been several DVC systems proposed in literature along with many improvements to individual subsystems. The first practical systems were the Stanford system [5] and PRISM (Berkeley) [6]. Since then many improvements have been introduced. Notable milestones include the DISCOVER [1] and VISNET [7] codecs, both of which are based on the architecture of the Stanford system. The authors have also presented a codec based on the Stanford architecture [8,9], of which the SI creation method is used in this paper.

The majority of competitive systems require a feedback path. Methods to remove feedback are based on estimating the required rate at the encoder and attempting to mitigate the visual effects of decoding failures at the decoder. Estimating the rate requires estimating the SI at the encoder, which increases the encoder complexity. Some estimate must be made and thus the increased complexity at the encoder becomes a trade-off point for R-D performance.

There have been some notable proposals for feedback suppressed systems. A pixel-domain feedback suppressed system was developed in [10]. However, the simulation results did not consider the key frames in the R-D performance, making comparisons difficult. In [11] and [12] there were attempts to learn the required rate from the SI estimate using machine learning and neural networks, respectively. In [13] a pixel-domain system, exploiting spatial and temporal correlations along with iterative decoding was presented. A system with a structure not relying on conventional Wyner-Ziv techniques and exploiting overlapped block motion estimation with probabilistic compensation (OBMEPC) and SI dependent correlation channel estimation (SID-CE) was presented in [14]. This system produced excellent results at a reduced encoding complexity. A system similar to [10] but for transform-domain DVC was presented in [15], where a multi-mode SI method was used at the encoder. More advanced and complete attempts at creating and analysing a feedback-free DVC system were presented in [16] and [17]. These systems rely on performing reduced complexity motion estimation at the encoder. Sophisticated techniques related to the correlation noise modelling, improved side information generation, and mitigation of decoding errors allowed authors in [16] to achieve R-D performance approaching that of the DISCOVER codec. A useful addition in [17] is the use of a hash to improve the motion estimation at the decoder. Future research may consider the complexity vs. rate-distortion trade-off of including low complexity motion estimation at the encoder, but in this paper, it is assumed that motion estimation is not desired at the encoder.

Encoding over the real field, quantizing the result and decoding with side information can be shown to be mathematically similar to compressive sensing (CS), if the error signal is assumed to be sparse or compressible. The effects of quantization and the R-D performance of quantized CS were analysed in [18]. In the proposed system, we show how the use a binning quantizer [19] can improve the R-D performance when the rate estimate is accurate. CS has been applied to video compression previously, for example in [20]. However, previous systems are different to the one presented here, since in these systems CS is used to reduce the sampling requirements in the pixel domain and, as such, the discrete cosine transform (DCT) is not used.

3 Real field coding

In this section, real field coding and its effect on Wyner-Ziv coding is described. Wyner-Ziv coding can be understood as a process of creating discontiguous quantization bins over the signal space. This can most easily be seen when the Wyner-Ziv code is implemented as a nested lattice quantizer [21]. A scalar quantizer followed by an error correction code performs the same role, where the syndrome describes a coset of codewords, each of which maps to a quantization cell in the signal space. The expected distortion, $D_{\mathrm{W|Z}}$, of a Wyner-Ziv code can be simplistically expressed as follows:

$$D_{\mathrm{W|Z}} = E[D_{\mathrm{B}}]\,P[\epsilon] + E[D_{\mathrm{Q}}]\,(1 - P[\epsilon]\,), \tag{1}$$

where $P[\epsilon]$ is the probability of a codeword error, D_{B} is the distortion introduced by a codeword error and D_{Q} is the distortion introduced by quantization. When the code is operating at a high enough rate $P[\epsilon] \rightarrow 0$ and the quantizer is the dominant distortion source. However, when the rate is too low, $P[\epsilon]$ can no longer be considered negligible and the distortion of a codeword error significantly affects performance. A real field code creates a contiguous subspace over the signal space. This changes the distortion function to a more continuous and gradual function of the noise, since there is no codeword error-based distortion. If $\mathbf{x} \in \mathbb{R}^N$ is the signal, encoding can be expressed in a similar manner as for conventional error correction codes:

$$\mathbf{y} = \mathbf{G}^T\mathbf{x}, \tag{2}$$

where $\mathbf{G} \in \mathbb{R}^{N\times(N+M)}$ is the generator matrix and $\mathbf{y} \in \mathbb{R}^{(N+M)}$ is the encoded signal. Since only the parity symbols are transmitted, the generator matrix is in systematic form:

$$\mathbf{G} = [\mathbf{I}_n|\mathbf{P}]\,, \tag{3}$$

where $\mathbf{P} \in \mathbb{R}^{N\times M}$ is a parity matrix. Let $\mathbf{s}$ be the parity symbols (also referred to as the syndrome in Wyner-Ziv literature):

$$\begin{aligned} \mathbf{s} &= [y_{N+1}\dots y_{N+M}]^T \qquad (4)\\ &= \mathbf{P}^T\mathbf{x}. \qquad (5) \end{aligned}$$

The parity symbols identify a specific subspace with $(N - M)$ dimensions, in $\mathbb{R}^N$ where $\mathbf{x}$ must lie. For example, if $N = 3$ and $M = 2$, then $\mathbf{s}$ will describe a specific line along which $\mathbf{x}$ lies. Decoding will find the most likely point on this line. In finite field coding, the same is true over the finite-field-based signal space, but when the correct codewords are mapped back to the signal space, they map to non-contiguous quantization regions.

Assuming a spherically symmetrical i.i.d Gaussian distributed error vector, the components of the error not along the line will be removed during decoding. Thus, if the original error variance was σ_e^2, then the decoded error variance, σ_d^2, becomes

$$\sigma_d^2 = \sigma_e^2\frac{N-M}{N}. \tag{6}$$

If the noise is not Gaussian, the decoded variance can be reduced even further. For example, if the noise is K-sparse, which means $||e||_0 = K, K \ll N$, then with $M \geq K + 1$, perfect reconstruction is possible, assuming an l_0-minimization decoder, though the problem is NP-complete [22]. If the noise is Laplace distributed, as is typically the case for DVC, or more generally compressible, then the performance lies between these two cases. Thus, decoding reduces the error variance in all cases and catastrophic decoding failures do not occur. Therefore, while there is no guarantee of a specific R-D performance due to a lack of knowledge about the channel, the method will always reduce the distortion in the SI, and there will be no catastrophic decoding failures due to noise variance underestimation.

Before the real field encoded parity symbols can be transmitted or stored, they must be quantized. We thus calculate:

$$\mathbf{s}^{\mathrm{Q}} = Q(\mathbf{s}). \tag{7}$$

The quantized parity symbols describe a cell in $\mathbb{R}^N$ that is N-dimensional, but bounded in M of those dimensions. The size of the bounded dimensions depends on the quantization bin size. There will thus be an additional quantization noise component added to the distortion as described in (6). Quantizer design is considered in Sections 5.3 and 5.4. From this point, we will refer to $\mathbf{P}^T$ as $\mathbf{H}$ to more closely align with notation used in the Wyner-Ziv coding literature.

4 Description of the proposed system

Figure 1 shows a block diagram of the proposed system. Incoming frames are split into key frames, which are H.264 intra-coded and WZ frames. Let the original pixel-domain WZ frame, at time t in the group-of-pictures (GOP), be referred to as $f[t]$. $f[t]$ (i,j) refers to the pixel at position (i,j). The GOP size (N_{G}) refers to the number of WZ frames in between the key frames plus one. The last key frame from the previous GOP will be referred to as $f[0]$ and the key frame at the end of the current GOP as $f[N_{\mathrm{G}}]$.

4.1 Encoder

The WZ frame is transformed with the block-based DCT to yield:

$$F[t] = \mathrm{DCT}(\,f[t]\,). \tag{8}$$

The bth subband of the frame will be referred to as $F_b[t]$, and the coefficient corresponding to the (v, w)th subblock

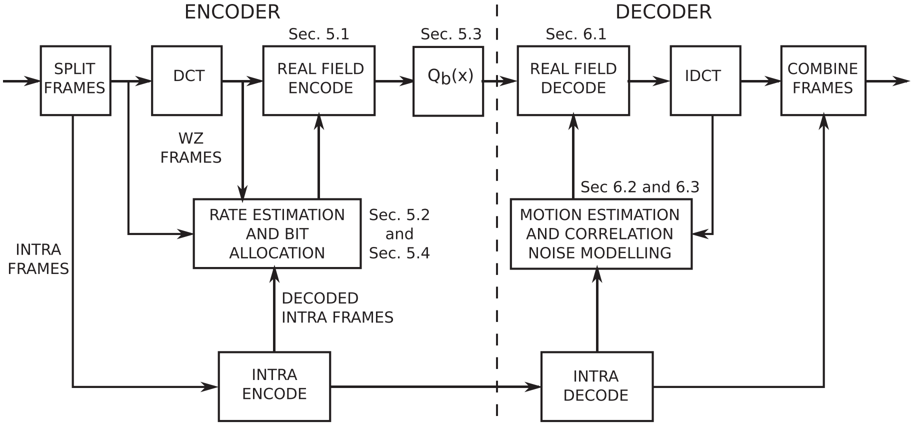

Figure 1 System diagram.

will be referred to as $F_b[t]\,(v,w)$. For notational brevity, we will now drop the time indices and assume that each WZ frame in the GOP is treated in a similar manner. The number of bits assigned to each subband is calculated using a rate estimation algorithm (Section 5.2) aiming for a specific distortion. The system attempts to achieve the same distortion as a conventional DVC system using a pre-defined quantization matrix. For this paper, eight quantization profiles taken from [23] are used in conjunction with a 4 × 4 DCT. We also consider quarter common intermediate format (QCIF) resolution (176 × 144) video sequences, which result in $N = 1,584$ elements per subband. After estimating the required rate for each subband according to the quantization profile, the bit budget ($\Re$) is allocated to a specific code size (M_b) and bits per quantized symbol (q_b) as described in Section 5.4. Let $\mathbf{x}_b = \text{vect}(F_b)$, be a $(N\times 1)$ vectorised version of F_b, then $\mathbf{x}_b$ is encoded:

$$\mathbf{s}_b = \mathbf{H}_b \mathbf{x}_b, \tag{9}$$

where $\mathbf{H}_b$ is the $M_b \times N$ parity check matrix used for the bth subband. Each encoded subband is then quantized with a symmetric uniform quantizer (Section 5.3). The quantized subband will be referred to as follows:

$$\mathbf{s}_b^{\mathrm{Q}} = Q_b(\mathbf{s}_b), \tag{10}$$

where $Q_b(\cdot)$ is the quantizer for the bth subband. If q_b represents the number of bits assigned to $Q_b(\cdot)$, then $\mathbf{s}_b^{\mathrm{Q}}$ will have elements in $\{0, 1, \ldots, L_b - 1\}$, where $L_b = 2^{q_b}$. Each quantized subband is then sent to the decoder along with the intra-coded key frames. Due to the lack of a feedback path, frames do not need to be buffered after encoding. The parameters required to describe the quantizer, as discussed in Section 5.3, must also be transmitted.

4.2 Decoder

At the decoder, the key frames and possibly other decoded frames are used to create an estimate of f. In the multi-hypothesis case more than one estimate of f is created. The hth hypothesis will be denoted by $\hat{f}_h$. For this system we use four hypotheses as described in [8]. Each hypothesis is transformed to produce $\hat{F}_h$ and each subband is vectorised to yield $\mathbf{y}_{hb}$:

$$\hat{F}_h = \text{DCT}(\hat{f}_h) \tag{11}$$

$$\mathbf{y}_{hb} = \text{vect}(\hat{F}_{hb}), \tag{12}$$

where $\hat{F}_{hb}$ is the bth subband of $\hat{F}_h$. If the error for each hypothesis is given as

$$r_h = f - \hat{f}_h, \tag{13}$$

we calculate an estimate $\hat{r}_h$. The method used to calculate the estimated error will differ for each hypothesis and depend on the method used to produce the hypothesis. From the error estimates, noise parameters are calculated at a coefficient level using the method developed in [23]. In order to improve the relative quality of the noise estimation for the different hypotheses, a motion-adaptive scaling factor is used to scale the Laplace noise parameters of the key frame and motion-based hypotheses as described in [8].

For each subband, the error estimates and the N_{H} hypotheses are combined with the received parity information and decoded using a Gaussian BP (GBP) algorithm which attempts to solve the equation:

$$\hat{x}_n = \underset{x_n \in \mathbb{R}}{\arg\max} f_{X|\mathbf{Y},\mathbf{S}^{\mathrm{Q}}}(x_n|\mathbf{Y}, \mathbf{s}^{\mathrm{Q}}), \tag{14}$$

where n indicates the nth coefficient in a subband with N coefficients and the subband indices are dropped for notational brevity. $\mathbf{Y}$ is an $N_{\mathrm{H}} \times N$ matrix with entries y_{hn} where the hth row represents the N received symbols of the hth hypothesis. As opposed to standard DVC, no reconstruction is required. The decoded bands are recompiled to produce $\hat{F}$, and then the pixel-domain output frame $\hat{f}$ is calculated:

$$\hat{f} = \mathrm{IDCT}(\hat{F}). \tag{15}$$

After decoding, the process is repeated in a method similar to successive refinement [24-27]. Using the decoded frame $\hat{f}$, a more accurate motion estimation and compensation algorithm is used to produce higher quality SI. New correlation noise estimates must then be produced for the SI, after which the GBP algorithm is used to decode the improved SI. This process may be iterated a number of times, though two iterations are typically sufficient. If the original decoding introduced an error, then the iterative process may worsen performance as the motion-compensated interpolation (MCI) process may produce worse quality SI. We will now discuss the subsections in more detail.

5 Encoder

In this section we discuss the specifics of the different parts of the system. Both the time and subbands indices have been dropped from most of the equations, since the same process is applied to all the subbands for each of the WZ coded frames.

5.1 Real field code design

Coding over the real field as opposed to a finite field has not been considered much due to the prevalence of digital systems. Some early analog codes were based on Bose-Chaudhuri-Hocquenghem (BCH) codes and the discrete Fourier transform (DFT) [28]. Large codes, analogous to finite field LDPC codes, were considered recently in the context of CS [29]. While exact design strategies are still being developed, it has been shown that codes that work well as binary LDPC codes also work well over the real field.

LDPC code design can be described as designing the decoding graph and choosing the connection weights. In this paper, we use the quasi-cyclic (QC) codes designed in [30] to design the decoding graph (the parity check matrix structure). QC codes were chosen because of their fast encoding algorithms and reduced storage requirements at the encoder, as compared with random codes. For a code with an $M \times N$ parity matrix $\mathbf{H}$, the connection weights (non-zero elements) were selected at random from a Gaussian distribution such that the expected total energy of the encoded signal is the same as that of the original signal $M\sigma_s^2 = N\sigma_x^2$.

The codes from [30] are girth-6 codes that are constructed using Galois fields. Depending on the number of parity symbols, M, we use a different construction field so as to ensure that the code remains girth-6 while being as dense as possible. If the construction field is the Galois field, $GF(2^{q_\mathrm{c}})$, then the value for q_c is chosen as

$$q_\mathrm{c} = \begin{cases} 5 & \text{if } 70 \leq M \leq 120 \\ 6 & \text{if } 120 < M \leq 250 \\ 7 & \text{if } M > 250. \end{cases} \tag{16}$$

M corresponds to the number of transmitted symbols. These values are sufficient for the video resolution considered in this paper, where $N = 1,584$. If the higher resolution frames are encoded, then N and M might increase. Codes should be designed with the required size in mind. We perform decoding using a version of GBP described in Section 6.1.

To demonstrate the performance of the real field codes, we evaluate the codes using random Laplace distributed data and noise. Figure 2 shows the ratio of the noise variance after decoding, σ_d^2, to the initial noise variance, σ_e^2, as a function of M for a range of signal-to-noise ratio (SNR) values. The number of quantization bits was fixed at a large value ($q = 12$) to remove the effect of the quantization noise from the analysis. The number of decoding iterations was fixed at 10. The figure shows that the decrease in the noise depends on the initial SNR. As M increases the slope decreases indicating a decreased effectiveness. This is however a function of the decoding algorithm and the number of decoding iterations. The performance at large M can be improved by increasing the number of iterations. This improvement would, however, come at the cost of a greater decoding complexity.

5.2 Rate estimation

The proposed system operates at an estimate of the theoretical Wyner-Ziv rate. This rate is calculated assuming a conventional system with a mid-rise symmetrical quantizer for the DC band and a dead-zone uniform quantizer for the AC bands. The bits assigned to each band for each rate-distortion point, are taken from the quantization profiles in [23]. While these quantizers are used to calculate the bit budget assigned to each subband, a different quantizer is used after encoding. This is because we are quantizing the encoded signal and not the original signal. The code size M_b and number of quantization bits q_b are the design parameters that affect performance where $\mathfrak{R} = M_b q_b$ is the bit budget.

We estimate the rate required to reconstruct the source $\mathbf{x}$ quantized with bin size Δ. If $\mathbf{x}$ is the signal, $\mathbf{e}$ is the error and $\mathbf{y}$ is the side information, then the system model for a given subband is

$$\mathbf{y} = \mathbf{x} + \mathbf{e}, \tag{17}$$

where the elements of $\mathbf{x}$ and $\mathbf{e}$ are Laplace distributed and drawn i.i.d from $X \sim \mathcal{L}(0, \alpha_x)$ and $E \sim \mathcal{L}(0, \alpha_e)$ respectively [31]. $\mathbf{x}$ is quantized to $\mathbf{x}^\mathrm{Q}$ with a q-bit

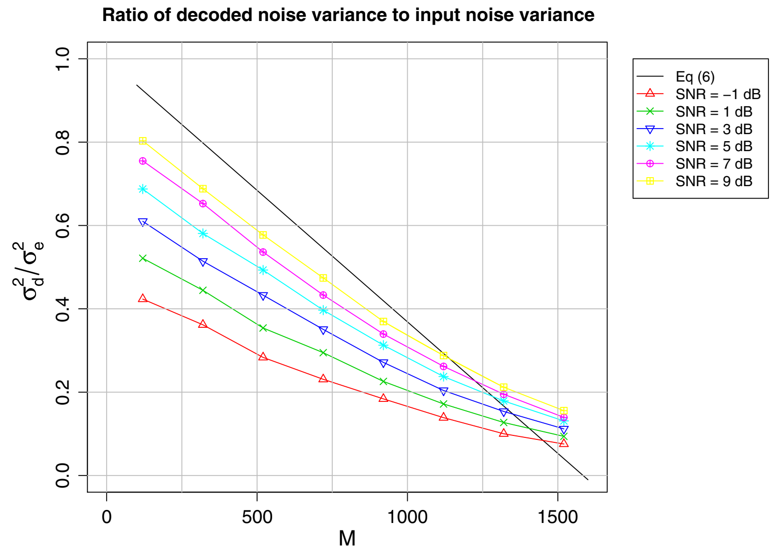

Figure 2 Plot of ratio of the noise variance after decoding to the initial noise variance. Plot of ratio of the noise variance after decoding to the initial noise variance as a function of *M* for several SNR values. The signal and the noise are both Laplace distributed and ten decoding iterations were performed.

quantizer, where the value for q is taken from [23]. The required transmission rate is lower bounded by the entropy remaining in the quantized source given the side information [4]. The per symbol rate can thus be expressed as

$$R \geq H(X^{\mathrm{Q}}|Y^{\mathrm{Q}}), \tag{18}$$

where $H(X^{\mathrm{Q}}|Y^{\mathrm{Q}})$ is the conditional entropy of quantized source, X^{Q}, given the quantized side information Y^{Q}. The final bit budget per subband will be $\mathfrak{R} = NR$.

5.2.1 *Correlation channel model*

In order to estimate the rate, we require an estimate of the signal variance and the noise variance. We use the decoded key frames, $\hat{f}[0]$ and $\hat{f}[N_{\mathrm{G}}]$, which are automatically available at the encoder due to the intra-coding algorithm, to create an interpolated frame without motion estimation:

$$\hat{f}[t] = \tfrac{1}{2}\left(\hat{f}[0] + \hat{f}[N_{\mathrm{G}}]\right). \tag{19}$$

An estimate of the residual, $\hat{r}$, is then used to calculate the required variance estimates:

$$\begin{aligned} \hat{r}[t] &= f[t] - \hat{f}[t] && (20)\\ \hat{R}[t] &= \mathrm{DCT}[\hat{r}[t]\,] && (21)\\ \sigma_e^2(b) &= \mathrm{VAR}(\hat{R}_b) && (22)\\ \sigma_x^2(b) &= \mathrm{VAR}(\mathrm{DCT}[f]\,(b)) && (23)\\ \sigma_y^2(b) &= \sigma_x^2(b) + \sigma_e^2(b). && (24) \end{aligned}$$

While the variance estimation method works fairly well for the DC band, the AC bands are not as accurate. To improve the estimate, we estimate the noise variance of band (b) as usual and then average the value with a predicted variance:

$$\hat{\sigma}_e^2(b)^* = \frac{1}{2}\hat{\sigma}_e^2(b) + \frac{1}{2}\hat{\sigma}_e^2(b-1)^* \frac{\hat{\sigma}_x^2(b)}{\hat{\sigma}_x^2(b-1)}. \tag{25}$$

5.2.2 *Conventional bit plane based estimates*

Most DVC systems encode each bit plane of a DCT subband independently using binary codes. As a result, an estimate of the required rate for each bit plane is required. The standard method for approaching this problem is to consider a binary symmetric channel (BSC) for each bit plane [16,32]. The minimum rate for a given bit plane is calculated as

$$\begin{aligned} R_b &\geq H_b(\epsilon) && (26)\\ &= -\epsilon \log \epsilon - (1-\epsilon)\log(1-\epsilon), && (27) \end{aligned}$$

where ϵ is the crossover probability of the virtual BSC. The crossover probability is defined as the probability that $x_b \neq \hat{x}_b$, where:

$$\hat{x}_b = \arg\max_{i=0,1} Pr(x_b = i|y, x_{b-1}, \ldots, x_1), \tag{28}$$

and x_b is the bth bit plane of X^{Q}. In practice, the bit plane entropy is calculated as [16,33]

$$\begin{aligned} p_n(i) &= \frac{Pr(x_b = i, x_{b-1}, \ldots, x_1|y)}{Pr(x_{b-1}, \ldots, x_1|y)} && (29)\\ H_b(p_n) &= -p_n(0)\log p_n(0) - p_n(1)\log p_n(1), && (30) \end{aligned}$$

where n is the nth symbol in the subband. The final entropy is then calculated as

$$H_b = \frac{1}{N} \sum_{\mathrm{WZ}^j} H_b(p_n), \tag{31}$$

with WZ^j representing all the symbols in the jth subband. This method requires N entropy calculations for each bit plane of each subband.

5.2.3 Symbol-wise conditional entropy estimation

The proposed system operates at a symbol level, but the bit-plane-based method could still be used by adding the entropy estimates for each bit plane in a symbol. However, there are some advantages to estimate the entropy directly. The definition of symbol-wise conditional entropy is as follows:

$$H(X^{\mathrm{Q}}|Y^{\mathrm{Q}}) = -\sum_{X^{\mathrm{Q}}} \sum_{Y^{\mathrm{Q}}} P(X^{\mathrm{Q}}, Y^{\mathrm{Q}}) \log P(X^{\mathrm{Q}}|Y^{\mathrm{Q}}) \tag{32}$$

$$= -\sum_{X^{\mathrm{Q}}} \sum_{Y^{\mathrm{Q}}} P(X^{\mathrm{Q}}, Y^{\mathrm{Q}}) \log \frac{P(X^{\mathrm{Q}}, Y^{\mathrm{Q}})}{P(Y^{\mathrm{Q}})}. \tag{33}$$

This method was used for a pixel-domain system [34] that assumed a uniform distribution for X. In general, estimating $H(X^{\mathrm{Q}}|Y^{\mathrm{Q}})$ directly from (32), requires evaluating $P(X^{\mathrm{Q}}, Y^{\mathrm{Q}})$ L^2 times, where $L = 2^q$ and q is the number of bits in the quantizer. Instead, we propose to evaluate a different form of the expression:

$$H(X^{\mathrm{Q}}|Y^{\mathrm{Q}}) = H(X^{\mathrm{Q}}) - I(X^{\mathrm{Q}}; Y^{\mathrm{Q}}) \tag{34}$$

$$= H(X^{\mathrm{Q}}) - H(Y^{\mathrm{Q}}) + H(Y^{\mathrm{Q}}|X^{\mathrm{Q}}), \tag{35}$$

where $I(X^{\mathrm{Q}}; Y^{\mathrm{Q}})$ is the mutual information between X^{Q} and Y^{Q}. The authors previously analysed this approach in [35]. We now make the simplifying assumption:

$$H(Y^{\mathrm{Q}}|X^{\mathrm{Q}}) \approx H(E^{\mathrm{Q}}), \tag{36}$$

which allows the required rate to be calculated as a function of the entropies of three single variables, each of which is simpler to calculate than the conditional entropy. This is a high-rate assumption which will typically be inaccurate for large Δ values.

Assuming that X is Laplace distributed, $H(X^{\mathrm{Q}})$ can be expressed as a summation that depends on the quantizer. When a dead-zone quantizer is used (as for the AC coefficients), we compute the entropy $H^{\mathrm{D}}(X^{\mathrm{Q}})$ as:

$$H^{\mathrm{D}}(X^{\mathrm{Q}}) = -\sum_{i=-\infty}^{\infty} P(i\Delta) \log_2[P(i\Delta)] \tag{37}$$

$$P(i\Delta) = \begin{cases} \int_{i\Delta}^{(i+1)\Delta} f_X(x)dx & \text{if } i > 0 \\ \int_{(i-1)\Delta}^{(i+1)\Delta} f_X(x)dx & \text{if } i = 0 \\ \int_{(i-1)\Delta}^{i\Delta} f_X(x)dx & \text{if } i < 0, \end{cases} \tag{38}$$

yielding:

$$P(i\Delta) = \begin{cases} \frac{1}{2} e^{-i\alpha_x \Delta}(1 - e^{-\alpha_x \Delta}) & \text{if } i > 0 \\ 1 - e^{-\alpha_x \Delta} & \text{if } i = 0 \\ \frac{1}{2} e^{i\alpha_x \Delta}(1 - e^{-\alpha_x \Delta}) & \text{if } i < 0. \end{cases} \tag{39}$$

Calculating the infinite sum and recombining yields :

$$H^{\mathrm{D}}(X^{\mathrm{Q}}) = -\,(1 - e^{-\alpha_x \Delta}) \log(1 - e^{-\alpha_x \Delta}) - e^{-\alpha_x \Delta} \times \left[\log\left(\frac{1 - e^{-\alpha_x \Delta}}{2} \right) - \frac{\alpha_x \Delta \log(e)}{1 - e^{-\alpha_x \Delta}} \right]. \tag{40}$$

For a uniform mid-rise quantizer that is symmetrical around zero, the entropy is denoted by $H^{\mathrm{S}}(X^{\mathrm{Q}})$ and can be calculated in a similar manner to yield:

$$H^{\mathrm{S}}(X^{\mathrm{Q}}) = -\,2 \left[\frac{1}{2}(1 - e^{-\alpha_x \Delta}) \right] \log \left[\frac{1}{2}(1 - e^{-\alpha_x \Delta} \right] - e^{-\alpha_x \Delta} \times \left[\log\left(\frac{1 - e^{-\alpha_x \Delta}}{2} \right) - \frac{\alpha_x \Delta \log(e)}{1 - e^{-\alpha_x \Delta}} \right]. \tag{41}$$

For a uniform mid-tread quantizer the entropy, $H^{\mathrm{M}}(X^{\mathrm{Q}})$, is

$$H^{\mathrm{M}}(X^{\mathrm{Q}}) = -\left(1 - e^{-\frac{\alpha_x \Delta}{2}}\right) \log\left(1 - e^{-\frac{\alpha_x \Delta}{2}}\right) - e^{-\frac{\alpha_x \Delta}{2}} \times \left[\log\left(\frac{e^{\frac{\alpha_x \Delta}{2}} - e^{-\frac{\alpha_x \Delta}{2}}}{2} \right) - \frac{\alpha_x \Delta \log(e)}{(1 - e^{-\alpha_x \Delta})} \right]. \tag{42}$$

These closed-form equations all assume that the quantizer is defined by the bin size and that the range of the variable is theoretically infinite. In reality, there will be a limited number of levels considered. We can thus also calculate $H(X^{\mathrm{Q}})$ by evaluating (37) and (39) over a finite number of levels. However, the difference between these is small enough to be ignored and we choose to use the closed-form solutions provided by the infinite quantizer instead.

If we assume that both the side information and the noise are Laplace distributed, then the equations derived for the source signal, (40), (42) and (41), can be used to calculate $H(Y^{\mathrm{Q}})$ and $H(E^{\mathrm{Q}})$ as well. However, many subbands in DVC coding are assigned only a small number of bits, and as a result the high-rate assumption may not hold. To improve the accuracy of the method, we use a slightly different approach for $H(E^{\mathrm{Q}})$. For the DC band, a conventional mid-rise uniform quantizer is typically used on the signal. In this case, we use the equation for the entropy of a mid-tread quantizer, (42), to calculate $H(E^{\mathrm{Q}})$. For the AC band, a dead-zone quantizer is typically used on the signal. When X is in the dead zone, $P(Y^{\mathrm{Q}}|X^{\mathrm{Q}})$ can be approximated by $P(E^{\mathrm{Q}})$, assuming a dead-zone quantizer for the error. However, when X is not in the dead zone, $P(Y^{\mathrm{Q}}|X^{\mathrm{Q}})$ is better approximated by $P(E^{\mathrm{Q}})$, assuming a mid-tread quantizer. Thus, for the AC bands, we estimate $H(E^{\mathrm{Q}})$ as follows:

$$H(E^{\mathrm{Q}}) = P(X^{\mathrm{Q}} = 0)H^{\mathrm{D}}(E^{\mathrm{Q}}) + P(X^{\mathrm{Q}} \neq 0)H^{\mathrm{M}}(E^{\mathrm{Q}}). \quad (43)$$

5.3 Quantizer

The encoded signal must be quantized before transmission. If the bit budget for a subband is $\mathfrak{R}$ and the length of the code is M_b, then we use a $q_b = \mathfrak{R}/M_b$ bit uniform quantizer with $L = 2^{q_b}$ levels. If the range of the quantizer is $A = 2||\mathbf{s}||_\infty$, then the quantizer function can be described as

$$Q(s) = \left(\lfloor f_c(s)/\Delta \rfloor + \frac{LD}{2}\right) \bmod L, \quad (44)$$

where

$$f_c(s) = \begin{cases} s & \text{if } |s| \leq \beta A \\ \mathrm{sign}(s)\beta A & \text{otherwise,} \end{cases} \quad (45)$$

$$\Delta = \frac{\beta A}{LD}. \quad (46)$$

The scaling parameter β allows for the quantizer to clip some measurements. The divisor D is a parameter that allows binning of the measurements to improve the rate performance. For $D = 1$, the quantizer is a standard uniform quantizer. The Δ, q_b and D parameters for each subband are transmitted along with the encoded sequence. Δ can be represented with 8 bits, q_b with 3 bits and D with 3 bits as well. As a result, the overhead, from transmitting these parameters, is negligible.

5.4 Bit allocation

The rate estimation algorithm provides a specific bit budget, $\mathfrak{R}$, for each subband. A method is required to allocate M_b and q_b , where $q_b M_b = \mathfrak{R}$. The quantizer can also employ binning to increase the effective number of bits per symbol. Employing binning will create the possibility of a decoding failure. We discuss binning here to show that the real field coding method can be adapted to include binning (discontiguous quantization) if desired. If D is the binning divisor, then an extra $q_D = \log_2 D$ bits are effectively gained. D should be chosen to be as large as possible while keeping the probability of a bin error at the decoder to a minimum. The maximum value for D is thus related to the noise variance and the quantizer range. Let $\Delta_{\mathrm{B}} = A\beta/D$ be the size of the bin. If σ_x^2 is the variance of the signal $\mathbf{x}$ and σ_s^2 is the variance of the parity signal $\mathbf{s}$ then

$$\sigma_s^2 = \sigma_x^2 \frac{N}{M_b}, \quad (47)$$

by design of the parity check matrix. Similarly, if σ_e^2 is the variance of the correlation noise, and $\mathbf{z} = \mathbf{He}$, then the variance of the noise on the parity sequence, σ_z^2, is

$$\sigma_z^2 = \sigma_e^2 \frac{N}{M_b}. \quad (48)$$

The bin size can be described as

$$\Delta_{\mathrm{B}} = \frac{c\sigma_z}{D} \quad (49)$$

$$= \frac{c\sigma_x}{D}\sqrt{\frac{N}{M_b}}, \quad (50)$$

where c is a constant such that $c\sigma_x$ equals the range of the quantizer. The probability that the noise on the encoded signal, $\mathbf{z}$, will yield a bin error can be approximated and upper bounded by ϵ:

$$\epsilon > \mathrm{P(bin\ error)} \quad (51)$$

$$= P\left(z > \frac{\Delta_{\mathrm{B}}}{2}\right) \quad (52)$$

$$= \exp\left(-\alpha\frac{\Delta_{\mathrm{B}}}{2}\right) \quad (53)$$

$$= \exp\left(-\alpha\frac{c\sigma_x}{2D}\sqrt{\frac{N}{M_b}}\right). \quad (54)$$

D is solved as follows:

$$D < -\frac{c\sigma_x}{\sqrt{2}\ln(\epsilon)\sigma_e\sqrt{\frac{N}{M_b}}}\sqrt{\frac{N}{M_b}} \quad (55)$$

$$= \frac{c}{\sqrt{2}\ln(1/\epsilon)}\sqrt{\mathrm{SNR}}. \quad (56)$$

For a typical choice of $c = 6$ and $\epsilon = 0.05$, this yields

$$D < 1.42\sqrt{\mathrm{SNR}}. \quad (57)$$

The bit allocation should be chosen to minimise the expected distortion. In general, for a bounded quantization cell, the area in the cell and the resultant distortion can be halved by adding an additional quantization bit per symbol or by doubling the number of constraints (M_b). This would indicate that the number of quantization bits per symbol (q_b) should be maximised. However, this is only true once the quantization region is bounded. This indicates that the allocation decision function should be piecewise defined. The number of constraints should be increased until the region is bounded at which point the number of bits per constraint should be increased.

Since we are aiming for a specific distortion level, we can estimate the number of errors larger than the allowable distortion. Let this number be K:

$$K = NP(e > \epsilon_{\text{aim}}). \tag{58}$$

If we treat the dimensionality of the error signal as K, then we need at least K dimensions to create a bounded quantization region for the error. We thus set the minimum value for M_b and the maximum value of q_b as follows:

$$M_b \geq K + 1 \tag{59}$$

$$q_b \leq \frac{\Re}{K+1}. \tag{60}$$

This is only an approximation as the noise model at the encoder is not exact and the decoder is suboptimal. M_b should be larger to improve the performance if the noise is underestimated. Our approach is to calculate M_b and q_b as above. If the effective number of bits $q_{\text{eff}} = q_b + q_D$, we then limit q_b as

$$q_b \leq q_{\text{aim}} + 2 - q_D. \tag{61}$$

We also require that $q_{\text{eff}} \geq 2$. After adjusting q_b to meet the above criteria, M_b is recalculated.

6 Decoder

6.1 Belief propagation over real fields

Belief propagation (BP) is a decoding algorithm that calculates an estimate of a marginal probability by exploiting factorization to reduce computational complexity. BP can be performed over any arbitrary field. The update equations require the computation of the product and the convolution of the marginal probability density functions (pdfs) being passed in the graph. When the coding is performed over the binary field, the pdf becomes a probability mass function (PMF) and can be represented using a single number. In the case of the real field, a full pdf is required. Calculating the convolution of a set of arbitrary functions is difficult. Thus, we use a version of relaxed or Gaussian BP [36], which assumes that the messages internal to the graph are Gaussian distributed. This allows for simple closed-form solutions of the update equations that rely only on the mean and the variance of the messages, reducing computational complexity. It also means that only two values need to be passed along each edge of the graph, reducing memory requirements.

Let μ_{ij} be the messages from variable node i to check node j and ν_{ji} be the message from check node j to variable node i. Let $f_X(x_i)$ be the prior information of variable node i, and $f_{\mathbf{Y}|X}(\mathbf{y}_i|x_i)$ be the multi-hypothesis side information. The standard update equation at the variable node is

$$\mu_{ij}(x_i) \propto f_X(x_i) f_{\mathbf{Y}|X}(\mathbf{y}_i|x_i) \prod_{k,k\neq j} \nu_{ki}\left(\frac{1}{H_{ji}} x_i\right), \tag{62}$$

while the update at the check node is

$$\nu_{ji}(x_i) \propto f_S(s_j) \bigoplus_{k,k\neq i} \mu_{kj}(H_{jk}x_k), \tag{63}$$

where $\bigoplus$ indicates convolution, H_{ji} is the connection weight from the parity check matrix, and $f_S(s_j)$ is the pdf of the jth parity symbol. As mentioned, due to computational complexity, we make the simplifying assumption that these messages are Gaussian distributed. Thus, each message is represented with only a mean and a variance:

$$\mu_{ij}^G(x_i) = \left(E[\mu_{ij}(x_i)]\,, V[\mu_{ij}(x_i)]\right) \tag{64}$$

$$\nu_{ji}^G(x_i) = \left(E[\nu_{ji}(x_i)]\,, V[\nu_{ji}(x_i)]\right), \tag{65}$$

where (64) and (65) can be calculated using the erf() function. As a result of the Gaussian assumption for the marginals, the updated equations are simplified. A product of Gaussian pdfs is a Gaussian pdf:

$$\prod_{i=1}^{I} G(\mu_i, \sigma_i^2) = SG(\mu_p, \sigma_p^2) \tag{66}$$

$$\sigma_p^2 = \left(\sum_{i=1}^{I} \frac{1}{\sigma_i^2}\right)^{-1} \tag{67}$$

$$\mu_p = \frac{\sum_{i=1}^{I} \mu_i \sigma_i^{-2}}{\sum_{i=1}^{I} \sigma_i^{-2}}, \tag{68}$$

where S is an irrelevant scaling factor, and I is the number of pdfs in the product. The convolution of Gaussian pdfs is also Gaussian:

$$\bigoplus_{i=1}^{I} G(\mu_i, \sigma_i^2) = G(\mu_c, \sigma_c^2) \tag{69}$$

$$\mu_c = \sum_{i=1}^{I} \mu_i \tag{70}$$

$$\sigma_c^2 = \sum_{i=1}^{I} \sigma_i^2. \tag{71}$$

Similarly, the connection weights are easily taken into account:

$$\mu_{ij}^G(H_{ji}x_i) = \left(H_{ji}E[\mu_{ij}(x_i)]\,, H_{ji}^2 V[\mu_{ij}(x_i)]\right) \tag{72}$$

$$\nu_{ji}^G\left(\frac{1}{H_{ji}} x_i\right) = \left(\frac{1}{H_{ji}} E[\,\nu_{ji}(x_i)]\,, \frac{1}{H_{ji}^2} V[\nu_{ji}(x_i)]\right). \tag{73}$$

To help with convergence, a damping parameter is added at the variable node. If we are calculating the update for iteration $t+1$, $\mu_{ij}^G(x_i)^{t+1}$, then we combine the result of (64), represented by $\mu_{ij}^G(x_i)$, with the output of the previous iteration $\mu_{ij}^G(x_i)^t$:

$$\mu_{ij}^G(x_i)^{t+1} = \beta \mu_{ij}^G(x_i)^t + (1-\beta)\mu_{ij}^G(x_i), \tag{74}$$

where $\beta = 0.7$ was found to yield good results. When employing binning in the quantizer, the pdf of the parity symbols, $f_S(s_j)$, consists of disjoint sections. If we simply calculate the outputs of the check nodes as in (65), we might end up with inconsistent updates. To solve this, we first calculate the most likely bin at the check node. First, we estimate the value of the syndrome symbol from the incoming messages:

$$\hat{s}_j = E\left[\bigoplus_k \mu_{kj}(H_{jk}x_k)\right]. \tag{75}$$

Then, we select the bin with the highest probability given $\hat{s}_j$:

$$\begin{aligned} s* &= \arg\max_l P(s_l|\hat{s}_j) & (76)\\ &= \arg\min_l |s_l - \hat{s}_j|, & (77)\end{aligned}$$

where $s_l,\ l \in \{1, D\}$ are the centres of the D bins corresponding to the received s_j^q. Now, $f_S(s_j)$ is a uniform pdf over the range corresponding to the most likely bin.

6.2 Side information creation

6.2.1 Initial decoding

A multi-hypothesis SI creation method using four hypotheses as developed in [8] is used. The hypotheses are the two reference frames, $\hat{f}[0]$ and $\hat{f}[N_G]$, a MCI-based frame using small blocks, $\hat{f}_S$, as well as a MCI-based frame using big blocks, $\hat{f}_B$. The method used to perform MCI is a variant of the one described in [37]. First the key frames are mean filtered. The filtered key frames are then used for bidirectional pixel level accuracy block matching (BM) using large blocks and the modified sum of absolute difference (SAD) metric in [37]. These vectors are used as starting points for a BM algorithm using smaller blocks with a fixed search range. The motion vector (MV) field generated with large blocks is quadrupled in density (oversampled by a factor of two in both directions), before being median filtered. By increasing the density, we allow the median filter to produce smoother edges and more precise vectors. The MV field generated with smaller blocks is also quadrupled in density before being median filtered. These two MV fields are then used to create two MCI hypotheses by interpolating from the original key frames.

6.2.2 Successive refinement

After BP decoding, the subbands are recombined and a pixel domain version of the WZ frame is produced. Using this decoded frame, the motion estimation process is repeated to produce better quality side information. This is similar to [24-27] where successive refinement was used to improve reconstruction distortion. Due to the use of real field coding, the effect is slightly different from conventional DVC as improved side information can result in decreased distortion from the GBP decoding process, whereas for finite field coding, there can be no further gain (other than improved reconstruction) after successful decoding of the bit planes.

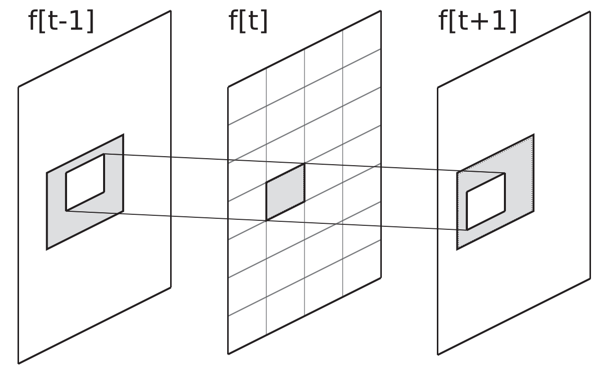

Figure 3 Bidirectional motion estimation. The gray boxes in frames $f[t-1]$ and $f[t+1]$ indicate the search range.

In order to use the decoded frame, the motion estimation method needs to be adjusted. Three different hypothesis frames are created. The first method performs pixel level accuracy bilinear BM motion estimation (ME) using the decoded frame as a hash (Figure 3) to produce $\hat{f}_I$. The matching metric used during ME is

$$M_I(L, M, R) = (1 - 2\lambda_M) \sum |L - R| + \lambda_M \sum |L - M| + \lambda_M \sum |R - M|, \tag{78}$$

where L, M and R represent the subblocks in the left key frame, the reconstructed frame and the right key frame, respectively. λ_M is a scaling factor that may be optimised. For this paper we used $\lambda_M = 0.25$.

The other two hypotheses, $\hat{f}_{L_r}$ and $\hat{f}_{R_r}$, are created by matching the key frames individually with the reconstructed frame (Figure 4). This allows non-linear motion to be tracked across the GOP.

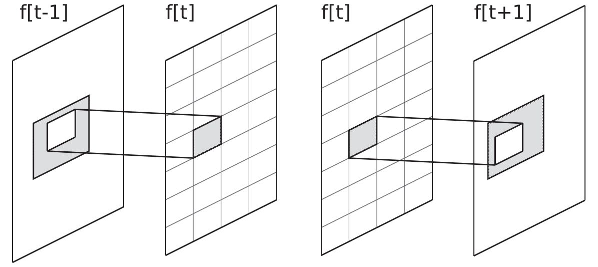

Figure 4 Single directional motion estimation. The gray boxes in frames $f[t-1]$ and $f[t+1]$ indicate the search range.

For these hypotheses, a simple SAD metric is used:

$$M_{\mathrm{L_r}}(L,R) = \sum |L - M| \tag{79}$$

$$M_{\mathrm{R_r}}(L,R) = \sum |R - M|. \tag{80}$$

The metrics are all adjusted by including a motion penalising term [37], which helps to ensure less noisy motion vectors.

6.3 Correlation noise modeling

The system uses a coefficient level noise model [23] which models the correlation noise for a single coefficient of a subband of a hypothesis as being Laplace distributed. The method for calculating this noise parameter (α) requires an estimate of the error in the hypothesis. This error is termed as the residual frame (r). For the multi-hypothesis system, a residual frame estimate is required for each hypothesis.

6.3.1 Residual estimation for the first decoding

For the MCI hypotheses, $\hat{f}_{\mathrm{S}}$ and $\hat{f}_{\mathrm{B}}$, the residual estimates, $\hat{r}_{\mathrm{S}}$ and $\hat{r}_{\mathrm{B}}$ respectively, are the difference between the motion-compensated frames used to perform the interpolation. For frame t in the GOP, this is given by

$$\hat{r}_{\mathrm{B}} = |\hat{f}^{\mathrm{B}}_{\mathrm{L}*} - \hat{f}^{\mathrm{B}}_{\mathrm{R}*}| \tag{81}$$

$$\hat{r}_{\mathrm{S}} = |\hat{f}^{\mathrm{S}}_{\mathrm{L}*} - \hat{f}^{\mathrm{S}}_{\mathrm{R}*}|. \tag{82}$$

Here, $\hat{f}^{\mathrm{B}}_{\mathrm{L}*}$ and $\hat{f}^{\mathrm{B}}_{\mathrm{R}*}$ indicate the motion-compensated reference frames used to interpolate $\hat{f}_{\mathrm{B}}$, while $\hat{f}^{\mathrm{S}}_{\mathrm{L}*}$ and $\hat{f}^{\mathrm{S}}_{\mathrm{R}*}$ indicate the motion-compensated reference frames used to interpolate $\hat{f}_{\mathrm{S}}$. For the reference frame hypotheses, the residual estimates ($\hat{r}_{\mathrm{L}}$ and $\hat{r}_{\mathrm{R}}$) are simply the difference between the two reference frames:

$$\hat{r}_{\mathrm{L}} = \hat{f}[0] - \hat{f}[N_{\mathrm{G}}] \tag{83}$$

$$\hat{r}_{\mathrm{R}} = \hat{f}[0] - \hat{f}[N_{\mathrm{G}}]\,. \tag{84}$$

The correlation noise for each hypothesis is estimated using the method in [23], and motion-adaptive scaling factors are applied as in [8]. All the hypotheses are then combined at a coefficient level using Bayesian fusion:

$$f_{\mathbf{Y}|X}(\mathbf{y}|x) = \prod_{h=1}^{H} f_{Y_h|X}(y_h|x), \tag{85}$$

before being passed to the GBP algorithm for decoding.

6.3.2 Residual estimation for later iterations

After the frame has been decoded the first time, we calculate new side information frames using the methods discussed in the previous section. If $\hat{f}_{\mathrm{I}}$, $\hat{f}_{\mathrm{L_r}}$ and $\hat{f}_{\mathrm{R_r}}$ are the three hypotheses, we estimate a residual frame for each as:

$$\hat{r}_{\mathrm{I}} = (1 - \lambda_{\mathrm{I}})|\hat{f}_{\mathrm{L}*} - \hat{f}_{\mathrm{R}*}| + \lambda_{\mathrm{I}}|\hat{f}_{\mathrm{I}} - M| \tag{86}$$

$$\hat{r}_{\mathrm{L_r}} = (1 - \lambda_{\mathrm{L_r}})|\hat{f}_{\mathrm{L_r}} - \hat{f}_{\mathrm{R_r}}| + \lambda_{\mathrm{L_r}}|\hat{f}_{\mathrm{L_r}} - M| \tag{87}$$

$$\hat{r}_{\mathrm{R_r}} = (1 - \lambda_{\mathrm{R_r}})|\hat{f}_{\mathrm{L_r}} - \hat{f}_{\mathrm{R_r}}| + \lambda_{\mathrm{R_r}}|\hat{f}_{\mathrm{R_r}} - M|. \tag{88}$$

Here, $\hat{f}_{\mathrm{L}*}$ and $\hat{f}_{\mathrm{R}*}$ indicate the motion-compensated key frames used to create $\hat{f}_{\mathrm{I}}$, M is the previous decoded frame, and the λs are tunable parameters. While they may be optimised in the future, we used $\lambda_{\mathrm{I}} = \lambda_{\mathrm{L_r}} = \lambda_{\mathrm{R_r}} = 0.5$.

From the residual estimate, we calculate a correlation noise estimate for each hypothesis. A method similar to [23] is used, but we include the output variance from the belief propagation decoder in calculating the value of the Laplace parameter α. The goal is to identify the areas where the decoder may have been in error, or where further refinement should occur. If $\sigma_b^{2(d)}(u,v)$ is the output variance from the GBP decoder, the new equation is:

$$D_b(u,v) = |\hat{R}_b(u,v) - \mu_b| \tag{89}$$

$$\hat{\alpha}_b(u,v) = \begin{cases} \sqrt{\frac{2}{\lambda\hat{\sigma}_b^2 + (1-\lambda)\sigma_b^{2(d)}(u,v)}}, & [D_b(u,v)]^2 \leq \hat{\sigma}_b^2 \\ \sqrt{\frac{2}{\lambda[D_b(u,v)]^2 + (1-\lambda)\sigma_b^{2(d)}(u,v)}}, & [D_b(u,v)]^2 > \hat{\sigma}_b^2 \end{cases}, \tag{90}$$

where λ is a tunable variable, that we set to one half.

7 Complexity analysis

In this section, we will discuss the complexity of the proposed system relative to conventional DVC as well as H.264 intra coding.

7.1 Encoder complexity

The overall complexity is a function of the GOP size, since it will be a combination of the key frame encoding complexity as well as the WZ encoding complexity. Initially, we consider only the complexity of the WZ frames, since the key frames have the same complexity in H.264 (intra) and in conventional DVC methods.

The complexity of the encoder can be described as the sum of the complexity of the subblocks. The main points where differences between the proposed method and conventional DVC may occur is in the conditional entropy estimation block, the quantizer and the real field encoder. We do not consider the complexity of motion estimation at the encoder and do not compare against systems that employ motion estimation. The complexity of the DCT operation will be the same for all transform domain systems.

Real field encoding compared to finite field coding will have the same number of operations (assuming the same code size), but the type of operation will differ: Real field (floating point) multiplication and addition versus

Table 1 Encoder run-time simulation

	Foreman			Coastguard			Soccer			Hall monitor		
RD	**H.264**	**Prop**	**Disc**	**H.264**	**Prop**	**Disc**	**H.264**	**Prop**	**Disc**	**H.264**	**Prop**	**Disc**
1	112	14	21	120	10	21	104	23	21	121	8	21
2	114	16	21	124	13	21	104	24	21	124	9	21
3	116	16	21	127	13	21	107	24	22	126	9	22
4	128	26	21	136	25	22	114	38	22	133	10	22
5	131	26	22	141	25	21	116	39	22	136	10	23
6	140	35	24	152	39	22	122	48	22	143	11	24
7	148	49	23	157	52	23	132	69	24	147	20	24
8	172	86	24	184	98	25	159	110	25	170	43	26

Run-time comparison of the encoder for the proposed system with the H.264 (intra) method as well as the DISCOVER codec. All results are in milliseconds per frame. Only WZ frames were considered for the DISCOVER codec and the proposed system. 'Disc' refers to the DISCOVER codec and 'Prop' refers to the proposed codec.

Galois field multiplication and addition. The complexity of the different operations will be platform and implementation specific. Typically one would expect finite field operations to be faster than the real field equivalent, however many modern processors are optimised to perform floating point operations and can perform them at high speed.

A similar point can be made for bit-plane-based systems with binary codes. These codes will have a larger number of operations compared to symbol-based coding, but the operations will be simple binary XOR and AND operations which are much faster.

The proposed conditional entropy estimator is faster than the conventional method (see Section 5.2), $O(1)$ vs. $O(q_b N)$ per subband, but the complexity of creating the side information estimate is also at least $O(N)$, so this does not represent a very large saving. The quantizers will also be similar in complexity though the proposed system only quantizes ($M_b < N$) values compared to the N values of a conventional system. Overall, we expect the encoding complexity of the proposed system to be similar to that of conventional DVC without motion estimation at the encoder. Exact comparisons will be implementation specific.

To experimentally evaluate the complexity we performed execution-time experiments. Simulation-based comparisons are difficult and always imperfect, but may still provide some insight into the expected performance of the systems. For these simulations we used the reference implementation of the H.264 intra codec. The proposed codec was implemented in C++. We also compare with the DISCOVER codec.

All the simulations were performed on a computer with a 2.93 GHz Intel Core 2 duo processor (Intel, Sta. Clara, CA, USA) and 2 Gb of RAM running Ubuntu Linux 12.04 LTS. The results can be seen in Table 1. The encoder run time of DISCOVER was analysed in [38]. Without access to the source code, run-time results for the DISCOVER encoder could not be accurately generated. To facilitate a comparison, the run time of DISCOVER (on this computer) was predicted. This was done by calculating the ratio of the per-frame run time of DISCOVER to H.264 (intra) from results presented in [38]. The run time of H.264 (intra), as produced in this simulation,

Table 2 Decoder run-time simulation

	Foreman		Coastguard		Soccer		Hall monitor	
RD	**Prop**	**Disc**	**Prop**	**Disc**	**Prop**	**Disc**	**Prop**	**Disc**
1	2.133	6.528	1.830	4.835	2.910	8.323	1.580	4.984
2	2.164	7.541	1.830	5.780	2.940	8.955	1.590	5.780
3	2.180	8.533	1.850	6.291	2.940	10.869	1.790	6.394
4	4.990	12.933	5.400	8.927	6.430	15.586	2.950	7.964
5	5.080	13.909	5.480	9.138	6.560	16.249	4.070	8.018
6	7.000	17.825	8.180	11.821	8.950	20.301	6.120	9.820
7	10.040	21.682	11.190	14.349	12.570	23.737	6.450	11.251
8	17.200	31.127	18.910	21.704	20.130	29.926	10.880	14.061

Run-time comparison of the decoder of the proposed system with the DISCOVER codec. All results are in seconds per frame. Only WZ frames were considered. 'Disc' refers to the DISCOVER codec and 'Prop' refers to the proposed codec.

Table 3 Key frame quantization parameters for the RD points

RD number	1	2	3	4	5	6	7	8
Hall monitor	37	36	35	33	32	30	29	24
Coastguard	38	37	36	34	33	31	30	26
Foreman	40	39	38	34	33	31	29	25
Soccer	43	42	40	36	35	33	30	25

The same values are used for all GOP sizes.

was then multiplied by the ratio to predict the run time of DISCOVER. While this is not ideal, it still provides a good idea of the relative performance of the different systems.

The table shows the average encoding time for a WZ frame as measured when $N_G = 2$. The overall encoding time for any GOP size can be estimated by combining the running time for the key frames and the WZ frames. From the table, one can see that the proposed system has a reduced run time when measured against the H.264 intra codec. The running time for the proposed system increases as the rate increases (larger RD points), since more subbands are encoded. A rate increase for a given subband also increases the size of the encoding matrix, which leads to a run-time increase.

The proposed system has approximately the same run time as the DISCOVER codec in the low-rate region. However, for the higher RD points, the proposed system has a longer run time than the DISCOVER codec. It should be mentioned that no attempt was made to optimise the encoder implementation, and it is expected that the run time can be significantly reduced in the future.

7.2 Decoder complexity

Decoding complexity is generally not considered a limitation for DVC and is not often considered when evaluating the performance in DVC systems. However, we will briefly discuss the effect of real field coding on the decoder complexity. The reduced complexity GBP algorithm used for decoding is similar to a binary BP decoding algorithm in terms of memory requirements and computational complexity. In GBP each edge requires two values to be transmitted while for binary BP one value is required. The update rules are also simple equations that scale with the check and variable node degree distributions. However, binary-coded systems decode each bit plane independently, thus it has to perform more than one decoding for each subband. The GBP also converges in fewer than ten iterations, which is less than most binary BP algorithms. GBP is thus expected to compare favourably to binary BP in terms of decoding speed. Non-binary BP is typically slower than binary BP and requires more memory. Each edge must transmit the entire PMF (L values). The update equations at the check node are also slow. A faster fast Fourier transform belief propagation (FFT-BP) algorithm still requires two FFT operations for every edge. Typically, FFT-BP is considered to be $O(q_b)$ times slower than binary BP. There are lower complexity non-binary BP algorithms in the literature, but discussing their relative merits is beyond the scope of this article. In general, GBP is expected to be faster than non-binary BP.

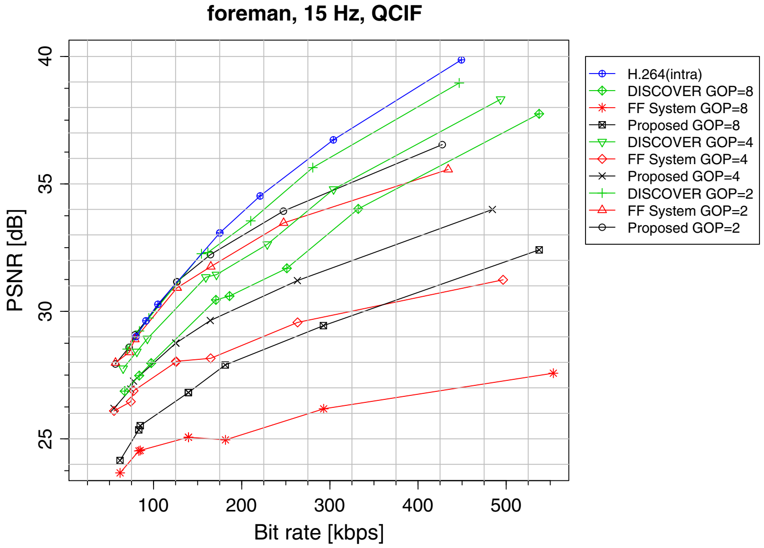

Figure 5 Comparison of the performance of the proposed system on the *foreman* sequence. Performance of the proposed system on the *foreman* sequence compared to the DISCOVER codec, the H.264 intra codec, and the FF system.

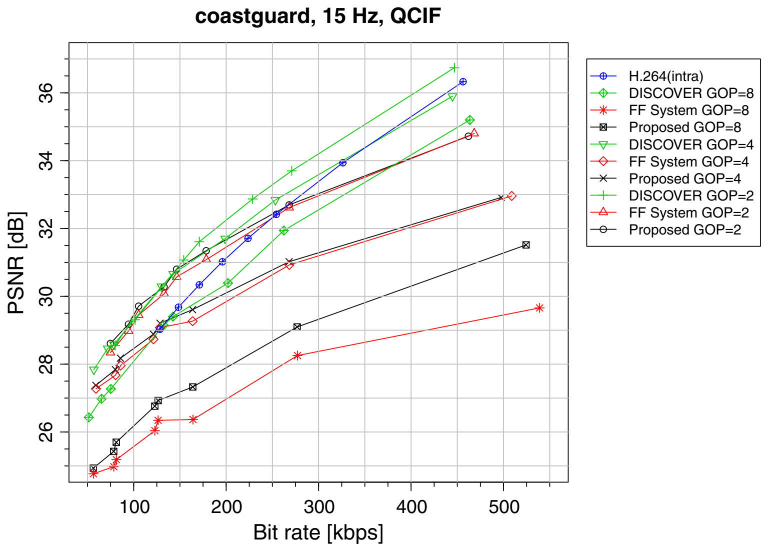

Figure 6 Comparison of the performance of the proposed system on the *coastguard* sequence. Performance of the proposed system on the *coastguard* sequence compared to the DISCOVER codec, the H.264 intra codec, and the FF system.

We include some execution-time experimental results showing the decoding time in Table 2. These results were produced under the same conditions as for the encoder. In contrast to the encoder, a publicly available executable was used to produce the run-time results for the decoder of the DISCOVER codec.

The table shows the average decoding time for a WZ frame as measured when $N_G = 2$. From the table one can see that the proposed system runs faster than the DISCOVER codec in all simulations. While this is not proof that the system will always be faster, it does indicate that real field decoding does not represent a complexity increase.

8 Simulation results and discussion

The system presented in this paper is tested on four standard test sequences: *foreman*, *coastguard*, *soccer* and *hall monitor*. The sequences are all QCIF format with a frame rate of 15 Hz. All the rate-distortion curves show results for the average bit rate and the distortion as calculated

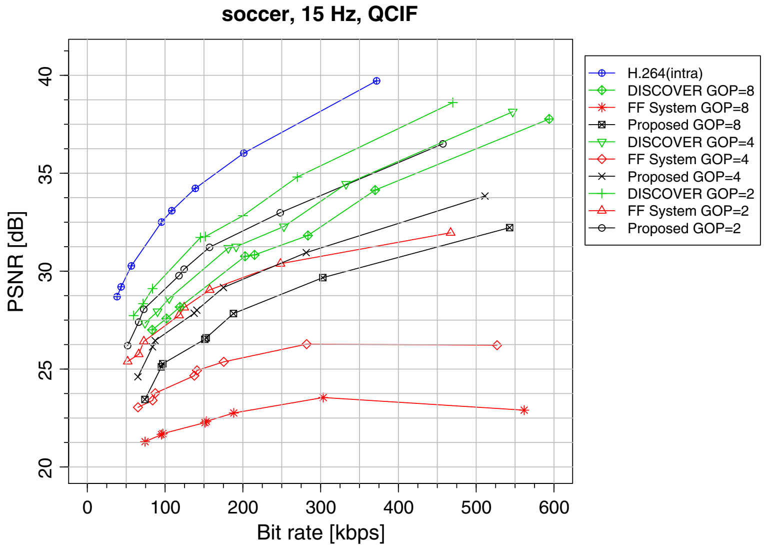

Figure 7 Comparison of the performance of the proposed system on the *soccer* sequence. Performance of the proposed system on the *soccer* sequence compared to the DISCOVER codec, the H.264 intra codec, and the FF system.

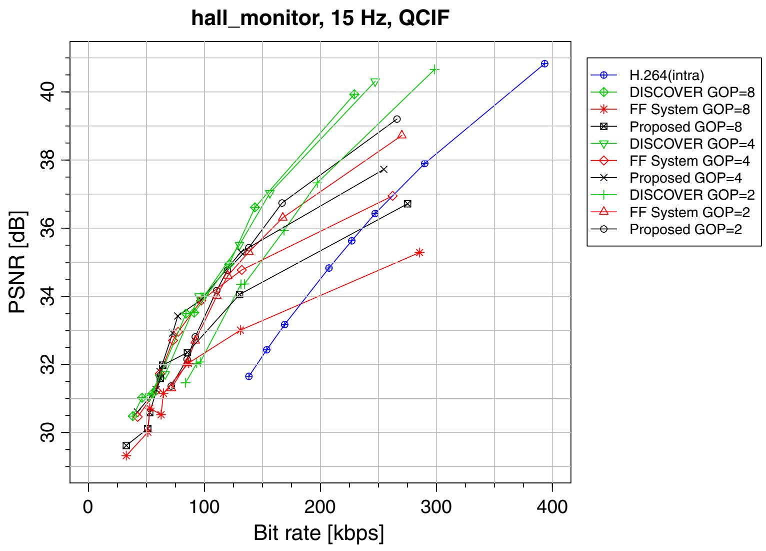

Figure 8 Comparison of the performance of the proposed system on the *hall monitor* sequence. Performance of the proposed system on the *hall monitor* sequence compared to the DISCOVER codec, the H.264 intra codec, and the FF system.

over all the frames in the entire sequence (key and WZ frames). All sequences consist of 150 frames.

The system will be compared against the H.264 intra codec, the DISCOVER codec [1], which is a feedback-based benchmark system in DVC literature, and a system previously developed by the authors [8], adapted for this paper, which will be called the 'FF system' in this section. All the results for the proposed system were obtained without binning, thus $D = 1$, to ensure that there are no decoding failures.

The finite field (FF) system is the same as in [8] and uses the same side information as the proposed system but uses

Table 4 Bjontegaard metric results

GOP size				2			4			8	
Sequence		Type	Disc	Prop	FF	Disc	Prop	FF	Disc	Prop	FF
Foreman	Full	Rate	10.40	13.80	26.05	42.23	106.74	1567.70	91.31	206.78	-
		DSNR	−0.40	−0.89	−1.47	−1.55	−3.60	−5.35	−2.76	−5.70	−8.74
	Mid	Rate	5.13	8.37	18.45	33.87	99.27	−48.09	78.63	277.07	-
		DSNR	−0.25	−0.50	−0.91	−1.43	−3.16	−4.49	−2.65	−5.45	−8.00
Coastguard	Full	Rate	−17.88	−9.88	−3.41	−9.04	37.46	77.84	26.65	193.56	378.17
		DSNR	0.98	0.14	−0.08	0.41	−1.61	−2.03	−0.90	−3.89	−5.20
	Mid	Rate	−19.33	−12.71	−7.42	−12.93	30.60	48.49	18.68	48.83	−91.21
		DSNR	1.07	0.50	0.27	0.58	−1.14	−1.47	−0.75	−3.44	−4.45
Soccer	Full	Rate	90.04	115.72	310.00	167.53	279.95	-	233.58	449.35	-
		DSNR	−2.86	−3.74	−5.89	−4.35	6.36	−9.99	−5.24	−8.06	−12.79
	Mid	Rate	85.63	123.33	321.26	164.38	304.88	-	230.12	571.79	-
		DSNR	−2.69	−3.61	−5.65	−4.07	−6.20	−9.41	−4.93	−7.95	−12.10
Hall	Full	Rate	−28.44	−36.57	−35.37	−42.01	−40.34	−41.54	−41.67	−15.18	−30.27
Monitor		DSNR	2.64	3.04	2.77	3.94	2.26	1.98	3.96	0.63	0.66
	Mid	Rate	−30.52	−39.49	−38.42	−44.77	−45.54	−45.96	−44.75	−28.28	−24.17
		DSNR	2.83	3.68	3.20	4.58	4.24	2.75	5.27	3.23	−0.47

The Bjontegaard metric for rate-distortion performance as measured against the H.264 intra code. 'Disc' refers to the DISCOVER codec, 'Prop' refers to the proposed codec and 'FF' refers to the finite field codec.

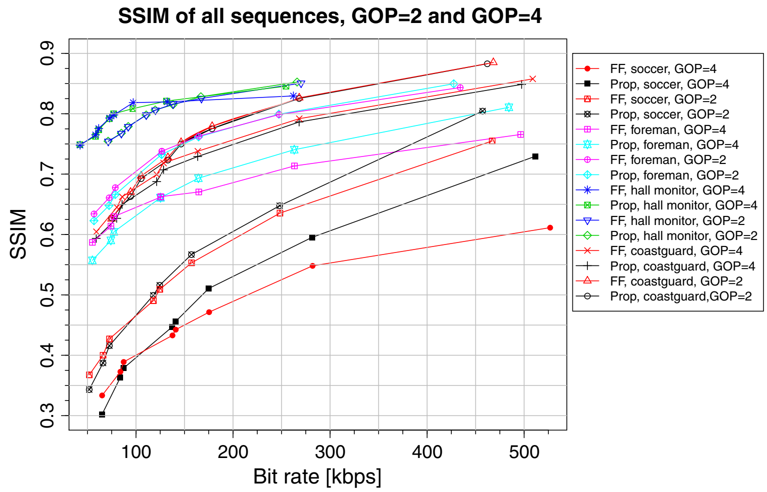

Figure 9 Comparison of the performance of the proposed system on all four sequences. Performance of the proposed system on all four sequences compared to the FF system using the SSIM metric. The results are shown for GOP sizes of 2 and 4.

non-binary codes defined over Galois fields. The modification for this paper is that it also uses the same estimated rate without feedback. The purpose of using the FF system is to highlight the effect of using real field coding while keeping most other aspects, such as the SI, constant. Comparisons with competing systems, while interesting, are more difficult to interpret. Key frames were encoded using the reference implementation of the intra mode of the H.264 codec. The QP parameters used are similar to those used by the DISCOVER codec and are provided in Table 3.

In the performance results for the *foreman*, *coastguard*, *soccer* and *hall monitor*, sequences can be seen in Figures 5, 6, 7 and 8, respectively. The Bjontegaard differential rate and peak signal-to-noise ratio (PSNR) metrics [39,40] were calculated for the proposed system, the FF system and the DISCOVER codec as measured against the H.264 (intra) results. The results can be seen in Table 4.

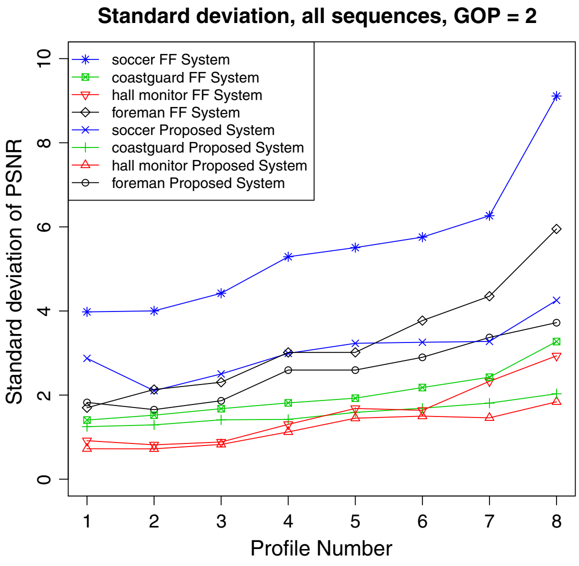

Figure 10 Standard deviation for all sequences, GOP = 2. Standard deviation of the PSNR over the four test sequences for both the proposed system as well as the FF system as a function of the profile number (rate). The same rate is used for both systems and the GOP size is 2.

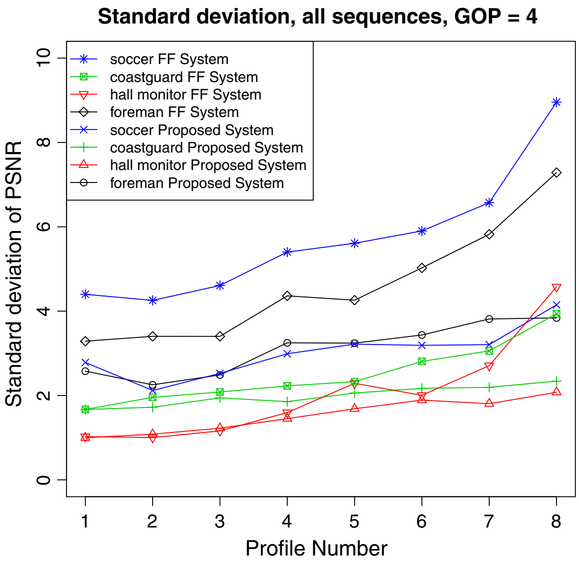

Figure 11 Standard deviation for all sequences, GOP = 4. Standard deviation of the PSNR over the four test sequences for both the proposed system as well as the FF system as a function of the profile number (rate). The same rate is used for both systems and the GOP size is 4.

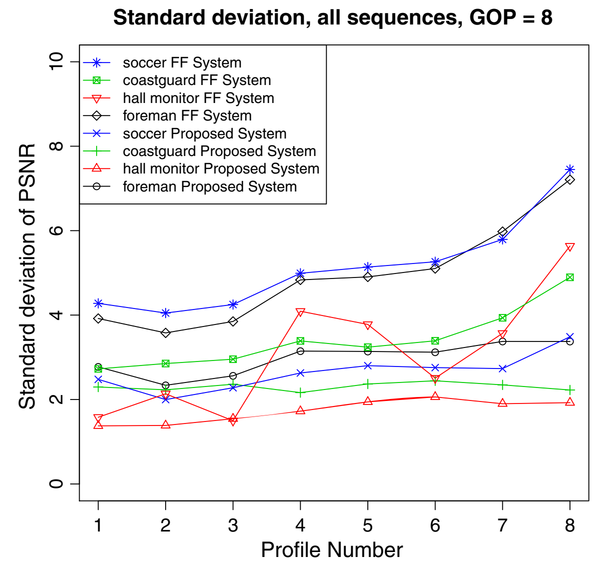

Figure 12 Standard deviation for all sequences, GOP = 8. Standard deviation of the PSNR over the four test sequences for both the proposed system as well as the FF system as a function of the profile number (rate). The same rate is used for both systems and the GOP size is 8.

In the table, 'DSNR' refers to the differential PSNR, and 'Rate' refers to the percentage difference in rate. The 'Type' column indicates whether the full range of RD points was considered or only the mid range.

The same trends are apparent in all the sequences. The proposed system performs comparably with the DISCOVER codec in the low-rate regime, but then deviates and performs less well in the high-rate regime. The FF-based system and the proposed system perform similarly in the *coastguard* sequence for the GOP size equal to 2 and 4. In all other cases, the proposed system has better performance. The performance gain when the GOP size is equal to 2 is small, but increases at higher rates. The performance gain is also larger for GOP sizes equal to 4 and 8. In general, the performance loss of the proposed system as measured against the DISCOVER codec increases with the GOP size.

In order to objectively analyse the perceptual quality of the proposed system, we include two sets of results. First, we include the performance of the proposed system and the FF system on all three sequences as measured using the structural similarity (SSIM) metric in Figure 9, as it is considered to measure perceptual quality better than the PSNR metric for a given frame. From the figure, one can see that the performance is similar for all three sequences when the GOP size is equal to 2. However, when the GOP size increases to 4, the proposed system performs better.

Secondly, to evaluate the perceptual quality of the entire sequence, we analyse the variance of the PSNR for each sequence. Figure 10 shows the standard deviation of the PSNR for the two systems over all profiles and all test sequences for a GOP size of 2. From the figure, it can be seen that in all cases, the standard deviation of the proposed system is less than that of the FF system. Figures 11 and 12 show the standard deviation of the PSNR for GOP sizes of 4 and 8 respectively. A similar trend is visible for these cases as well. A large variance in distortion results in unpleasant flicker in the video sequence. While the proposed system and the FF system have similar average R-D performance for GOP equal to 2, subjectively, the quality of the FF system appears worse because of image flickering. The proposed system on the other hand has a gentler degradation in image quality resulting in less flickering.

As an example, Figure 13 shows the same frame coded at the same rate by the two systems. The estimated rate is obviously lower than required. As a result, the quality in the FF system degraded significantly while the visual quality in the proposed system degraded much less. As a caveat, no attempt was made in the FF system to detect and ameliorate decoding failures. In cases where the estimated rate was high enough, the FF system decoded correctly and produced a higher quality frame than the proposed system.

Though the results are not plotted, the systems presented in [16] and [17] both have better average R-D performance than the proposed system and achieve similar

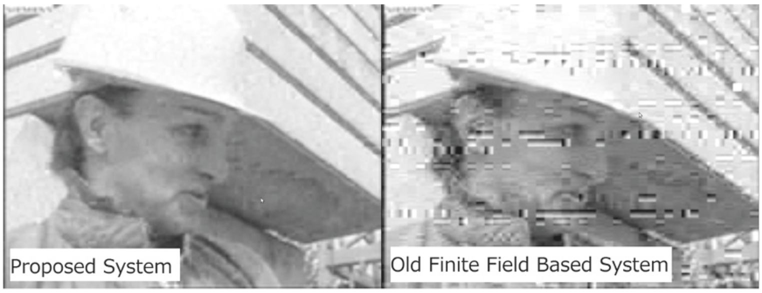

Figure 13 Comparison of the same frame encoded and decoded using FF system and proposed system. Both frames used the same rate.

average R-D performance as the DISCOVER codec. Using low complexity motion estimation at the encoder, they are able to achieve accurate rate estimation. The addition of hashes further improves the performance of these systems. The exact complexity and run-time losses incurred by the motion estimation are not available. The SSIM and PSNR variance results for these systems are also not available.

For the proposed system, it is expected that employing a similar low complexity motion estimation algorithm at the encoder will improve the accuracy of the rate estimation and by extension the average R-D performance of the proposed system as well. Furthermore, accurate rate estimation will allow for binning as described in Section 5.4, which can further improve the average rate-distortion performance. Hashing and more advanced SI creation methods could also be employed.

9 Conclusion

This paper proposed a new approach to feedback suppression in DVC systems that relies on codes defined over the real field. Despite the removal of the feedback path, the encoder complexity was not significantly increased, since no motion estimation was performed at the encoder. The system showed average R-D performance comparable to that of a feedback-based system at low rates. At high rates, there was a reduction in performance compared to feedback-based systems. However, compared to conventional finite-field-based feedback-free systems, the variance in the distortion was reduced. This resulted in improved perceptual visual quality.

Competing interests

The authors declare that they have no competing interests.

Received: 15 April 2013 Accepted: 19 November 2013
Published: 5 December 2013

References

1. X Artigas, J Ascenso, M Dalai, S Klomp, M Ouaret, The DISCOVER codec: architecture, techniques and evaluation, in *Picture Coding Symposium (PCS)* (Lisbon,7–9 November 2007)
2. J Slowack, J Skorupa, N Deligiannis, P Lambert, A Munteanu, R Van De Walle, Distributed video coding with feedback channel constraints. Circuits Syst. Video Technol., IEEE Trans. **22**(7), 1014–1026 (2012)
3. A Wyner, Ziv J, The rate-distortion function for source coding with side information at the decoder. IEEE Trans. Inf. Theory. **22**, 1–10 (1976)
4. D Slepian, J Wolf, Noiseless coding of correlated information sources. IEEE Trans. Inf. Theory. **19**(4), 471–480 (1973)
5. A Aaron, R Zhang, B Girod, Wyner-Ziv coding of motion video, in *Conference Record of the Thirty-Sixth Asilomar Conference on Signals, Systems and Computers,* Pacific Grove, 3–6 November 2002, *Volume 1* (2002), pp 240–244
6. R Puri, K Ramchandran, PRISM: a new robust video coding architecture based on distributed compression principle, in *Annual Allerton Conference on Communication, Control, and Computing (ACCC)* (Urbana, 2–4 October 2002)
7. J Ascenso, C Brites, F Dufaux, A Fernando, T Ebrahim, F Pereira, S Tubaro, The VISNET II DVC codec: architecture, tools and performance, in *European Signal Processing Conference (EUSIPCO)* (Aalborg, 23–27 August 2010)
8. D Louw, H Kaneko, A multi-hypothesis non-binary LDPC code based distributed video coding system, in *Proceedings of the 13th IASTED International Conference on Signal and Image Processing, Volume 74,* Dallas, 14–16 December 2011
9. D Louw, H Kaneko, A system combining extrapolated and interpolated side information for single view multi-hypothesis distributed video coding, in *Proceedings International Symposium on Information Theory and its Applications (ISITA),* Honolulu, 28–31 October 2012, pp. 779–783
10. M Morbée, J Prades-Nebot, A Roca, A Pižurica, W Philips, Improved pixel-based rate allocation for pixel-domain distributed video coders without feedback channel, in *Proceedings of the 9th international conference on Advanced concepts for intelligent vision systems,* ACIVS'07. (Springer, Berlin, Heidelberg, 2007), pp. 663–674
11. J Martinez, G Fernandez-Escribano, WARJWeerakkody Kalva H, WAC Fernando, A Garrido, Feedback free DVC architecture using machine learning, in *15th IEEE International Conference on Image Processing, 2008. ICIP 2008,* San Diego, 12–15 October 2008, pp. 1140–1143
12. I Nickaein, M Rahmati, S Ghidary, A Zohrabi, Feedback-free and hybrid distributed video coding using neural networks, in *2012 11th International Conference on Information Science, Signal Processing and their Applications (ISSPA),* Montreal 3–5 July 2012, pp. 528–532
13. WARJ Weerakkody, WAC Fernando, ABB Adikari, Unidirectional distributed video coding for low cost video encoding. IEEE Trans. Consumer Electron. **53**(2), 788–795 (2007)
14. N Deligiannis, A Munteanu, T Clerckx, J Cornelis, P Schelkens, Overlapped block motion estimation and probabilistic compensation with application in distributed video coding. IEEE Signal Process. Lett. **16**(9), 743–746 (2009)
15. T Sheng, X Zhu, G Hua, H Guo, J Zhou, C Chen, Feedback-free rate-allocation scheme for transform domain Wyner–Ziv video coding. Multimedia Syst. **16**(2), 127–137 (2010)
16. C Brites, F Pereira, An efficient encoder rate control solution for transform domain Wyner-Ziv video coding. IEEE Trans. Circuits Syst. Video Technol. **21**(9), 1278–1292 (2011)
17. F Verbist, N Deligiannis, SM Satti, A Munteanu, P Schelkens, Iterative Wyner-Ziv decoding and successive side-information refinement in feedback channel-free hash-based distributed video coding, in *Proceedings of SPIE 8499, Applications of Digital Image Processing* XXXV, 84990O–1–84990O–10 (2012)
18. V Goyal, A Fletcher, S Rangan, Compressive sampling and lossy compression. Signal Process. Mag. IEEE. **25**(2), 48–56 (2008)
19. U Kamilov, V Goyal, S Rangan, Message-passing de-quantization with applications to compressed sensing. IEEE Trans. Signal Process. **60**(12), 6270–6281 (2012)
20. Y Baig, E Lai, A Punchihewa, Distributed video coding based on compressed sensing, in *Proceedings of the IEEE International Conference on Multimedia and Expo Workshops (ICMEW),* Melbourne, 9–13 July 2012, pp. 325–330
21. Z Liu, S Cheng, A Liveris, Z Xiong, Slepian-wolf coded nested lattice quantization for Wyner-Ziv coding: high-rate performance analysis and code design. IEEE Trans. Inf. Theory. **52**(10), 4358–4379 (2006)
22. D Donoho, Compressed sensing. IEEE Trans. Inf. Theory. **52**(4), 1289–1306 (2006)
23. C Brites, F Pereira, Correlation noise modeling for efficient pixel and transform domain Wyner-Ziv Video coding. IEEE Trans. Circuits Syst. Video Technol. **18**(9), 1177–1190 (2008)
24. R Martins, C Brites, J Ascenso, F Pereira, Refining side information for improved transform domain Wyner-Ziv video coding. IEEE Trans. Circuits Syst. Video Technol. **19**(9), 1327–1341 (2009)
25. AAE Ailah, G Petrazzuoli, J Farah, M Cagnazzo, B Pesquet-Popescu, F Dufaux, Side information improvement in transform-domain distributed video coding, in *Proceedings of the SPIE - Applications of Digital Image Processing XXXVII,* San Diego, 17–21 August 2012
26. N Deligiannis, F Verbist, J Slowack, R Van De Walle, P Schelkens, A Munteanu, Joint successive correlation estimation and side information refinement in distributed video coding, in *2012 Proceedings of the 20th European Signal Processing Conference (EUSIPCO),* Bucharest, 27–31 August 2012, pp. 569–573
27. S Ye, M Ouaret, F Dufaux, T Ebrahimi, Improved side information generation for distributed video coding by exploiting spatial and temporal correlations. EURASIP J. Image Video Process. **2009**, 683510 (2009)

28. T Marshall Jr, Coding of real-number sequences for error correction: a digital signal processing problem. IEEE J. Selected Areas Commun. **2**(2), 381–392 (1984)
29. A Dimakis, R Smarandache, P Vontobel, LDPC codes for compressed sensing. IEEE Trans. Inf. Theory. **58**(5), 3093–3114 (2012)
30. S Song, B Zhou, S Lin, K Abdel-Ghaffar K, A unified approach to the construction of binary and nonbinary quasi-cyclic LDPC codes based on finite fields. IEEE Trans. Commun. **57**, 84–93 (2009)
31. E Lam, J Goodman, A mathematical analysis of the DCT coefficient distributions for images. IEEE Trans. Image Process. **9**(10), 1661–1666 (2000)
32. C Fu, J Kim, Encoder rate control for block-based distributed video coding, in *2010 IEEE International Workshop on Multimedia Signal Processing (MMSP),* Saint Malo, 4–6 October 2010, pp. 333–338
33. S Cheng, Z Xiong, Successive refinement for the Wyner-Ziv problem and layered code design. IEEE Trans. Signal Process. **53**(8), 3269–3281 (2005)
34. C Yaacoub, J Farah, B B Pesquet-Popescu, Feedback channel suppression in distributed video coding with adaptive rate allocation and quantization for multiuser applications. EURASIP J. Wireless Commun. Netw. **2008** (2008). doi:10.1155/2008/427247
35. D Louw, H Kaneko, Efficient conditional entropy estimation for distributed video coding, in *Proceedings of the 30th Picture Coding Symposium (PCS),* San Jose, 8–11 December 2013
36. S Rangan, Estimation with random linear mixing, belief propagation and compressed sensing, in *2010 44th Annual Conference on Information Sciences and Systems (CISS),* Princeton, 17–19 March 2010, pp. 1–6
37. J Ascenso, C Brites, F Pereira, Content adaptive Wyner-ZIV video coding driven by motion activity, in *IEEE International Conference on Image Processing,* Atlanta, 8–11 October 2006, pp. 605–608
38. F Pereira, J Ascenso, C Brites, Studying the GOP size impact on the performance of a feedback channel-based Wyner-Ziv video codec, in *Advances in Image and Video Technology, Volume 4872 of Lecture Notes in Computer Science*. Ed. by D Mery, L Rueda (Springer, Berlin, Heidelberg, 2007), pp. 801–815
39. G Bjontegaard, Calculation of average PSNR differences between RD-curves, in *the 13th VCEG-M33 Meeting,* Austin, April 2001
40. G Sullivan, G Bjontegaard, Recommended simulation common conditions for H.26L coding efficiency experiments on low-resolution progressive-scan source material, in *VCEG-N81, 14th Meeting,* Santa Barbara, September 2001

doi:10.1186/1687-6180-2013-181
Cite this article as: Louw and Kaneko: **Suppressing feedback in a distributed video coding system by employing real field codes.** *EURASIP Journal on Advances in Signal Processing* 2013 **2013**:181.