Exploiting the Error-Correcting Capabilities of Low Density Parity Check Codes in Distributed Video Coding using Optical Flow

Rakêt, Lars Lau; Søgaard, Jacob; Salmistraro, Matteo; Luong, Huynh Van; Forchhammer, Søren

Published in:
Proceedings of SPIE, the International Society for Optical Engineering

Link to article, DOI:
10.1117/12.929435

Publication date:
2012

Citation (APA):
Rakêt, L. L., Søgaard, J., Salmistraro, M., Luong, H. V., & Forchhammer, S. (2012). Exploiting the Error-Correcting Capabilities of Low Density Parity Check Codes in Distributed Video Coding using Optical Flow. *Proceedings of SPIE, the International Society for Optical Engineering*, *8499*. https://doi.org/10.1117/12.929435

Exploiting the Error-Correcting Capabilities of Low Density Parity Check Codes in Distributed Video Coding using Optical Flow

Lars Lau Rakêt[a], Jacob Søgaard[b], Matteo Salmistraro[b], Huynh van Luong[b],
Søren Forchhammer[b]

[a]Department of Computer Science, University of Copenhagen, Universitetsparken 1,
2100 Copenhagen, Denmark;
[b]Department of Photonics Engineering, Technical University of Denmark, Ørsteds Plads,
2800 Kgs. Lyngby, Denmark

ABSTRACT

We consider Distributed Video Coding (DVC) in presence of communication errors. First, we present DVC side information generation based on a new method of optical flow driven frame interpolation, where a highly optimized TV-L^1 algorithm is used for the flow calculations and combine three flows. Thereafter methods for exploiting the error-correcting capabilities of the LDPCA code in DVC are investigated. The proposed frame interpolation includes a symmetric flow constraint to the standard forward-backward frame interpolation scheme, which improves quality and handling of large motion. The three flows are combined in one solution. The proposed frame interpolation method consistently outperforms an overlapped block motion compensation scheme and a previous TV-L^1 optical flow frame interpolation method with an average PSNR improvement of 1.3 dB and 2.3 dB respectively. For a GOP size of 2, an average bitrate saving of more than 40% is achieved compared to DISCOVER on Wyner-Ziv frames. In addition we also exploit and investigate the internal error-correcting capabilities of the LDPCA code in order to make it more robust to errors. We investigate how to achieve this goal by only modifying the decoding. One of approaches is to use bit flipping; alternatively one can modify the parity check matrix of the LDPCA. Different schemes known from LDPC codes are considered and evaluated in the LDPCA setting. Results show that the performance depend heavily on the type of channel used and on the quality of the Side Information.

Keywords: Distributed Video Coding, LDPC, Error-Resilience, Side Information Generation, Frame Interpolation

1. INTRODUCTION

The distributed video coding paradigm contrasts ordinary hybrid video coding, by fully or partly exploiting the temporal redundancy of video data at the decoder side. This also means that one has to rethink the components one would normally use. In particular one does not have to worry about coding motion vectors, which makes it possible to consider alternative motion estimation strategies. In addition the use of alternative decoders may give rise to other opportunities. The contribution of this paper is two-fold. First we propose a novel side information generation scheme, which significantly increases the bitrate saving. Secondly we investigate methods for exploiting the error-correcting capabilities of the LDPCA[1] (low-density parity-check accumulate) code in DVC, in the case of transmission errors.

A novel DVC side information generation scheme is proposed. In this new setup three different motion estimates are used to generate a single side information frame. The motion is estimated using standard forward and backward schemes, and in addition we include a symmetric estimate, that has recently been showed to give superior quality for frame interpolation.[2] Together these three estimates are used for generating side information

Further author information:
Søren Forchhammer[b]: E-mail: sofo@fotonik.dtu.dk, Telephone: +45 45253622

for Wyner-Ziv frames, and we demonstrate that results from this procedure outperforms overlapped block motion compensation and optical flow methods[3, 4], resulting in a significant bitrate saving.

Various techniques for error correction, that has been developed for fixed rate LDPC codes, has been implemented and compared, using transmission modeled by a Binary Symmetric Channel (BSC) and Gaussian channel. We have restricted ourselves to methods which do not require alterations of the encoder but only of the decoder. While previous works[5] addressed the problem using rate-adaptive Turbo codes, this is the first study on using LDPCA codes in DVC to also combat transmission errors, to the best of our knowledge.

The rest of the paper is organized as follows: In the next section we will briefly describe the DVC setup used. In Section 3 we will consider our optical flow driven side information generation. Section 4 describes the error-correcting techniques that has been implemented. Results are given in Section 5, and finally conclusions are drawn in the last section.

2. DISTRIBUTED VIDEO CODING

An efficient approach to DVC is Transform Domain Wyner-Ziv (TDWZ) video coding with a feedback channel, which was first proposed by Girod et al.[6] The decoder controls the rate by requests over a feedback channel. The DISCOVER codec[7] improved the performance of the initial TDWZ architecture and it constitutes a well known benchmark . More recently various improvements have been reported. TDWZ video coding with a cross-band noise model was proposed[3] to further improve the coding efficiency by utilizing the cross-band correlation, without changing the encoder.

The architecture of a TDWZ video codec[7] is depicted in Fig. 1. In this system, the sequence of frames is split into key frames and so-called Wyner-Ziv frames. Key frames are intra coded using conventional video coding techniques such as H.264/AVC intra coding. The Wyner-Ziv frames are transformed (4×4 DCT), quantized and decomposed into bitplanes. Each bitplane is fed to a rate-compatible LDPC Accumulate (LDPCA) encoder[1] from most significant bitplane to least significant bitplane. The corresponding error correcting information is stored in a buffer and requested by the decoder through a feedback channel.

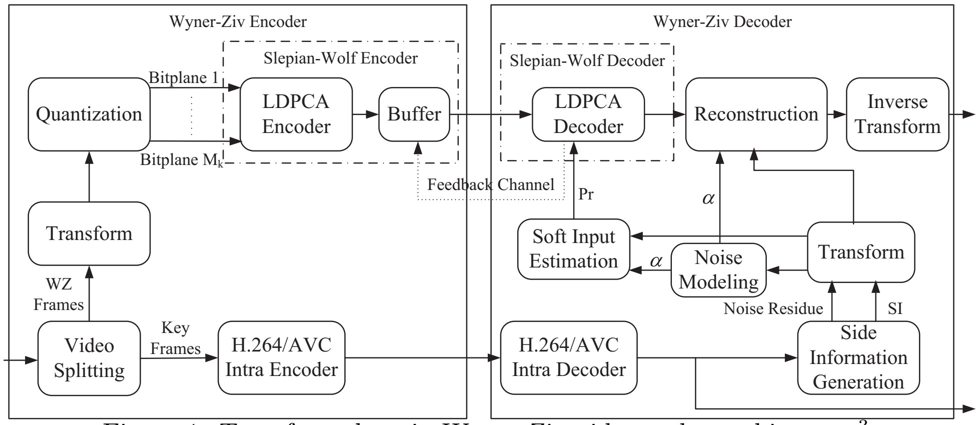


Figure 1: Transform domain Wyner-Ziv video codec architecture[3].

The Wyner-Ziv frame is predicted at the decoder side by using already decoded frames as references. The predicted frame, called the Side Information (SI) frame, is an estimate of the original Wyner-Ziv frame. Given the available SI, soft-input information (conditional probabilities for each bit) within each bitplane is estimated using a noise model. Thereafter the LDPCA decoder starts to decode the bitplanes selected by the quantizer, ordered from most to least significant bitplane, to correct the bit errors. The decoder requests bits from the buffer until the bitplane is decoded. Thereafter CRC bits are sent for confirmation. After all the bitplanes are successfully decoded, the Wyner-Ziv frame can be decoded through combined de-quantization and reconstruction followed by an inverse transform.

In DVC (Fig. 1) there are three different channels, namely the transmission channel, the virtual channel and the feedback channel. Through the transmission channel the parity bits are sent from the encoder to the decoder. The feedback channel is used by the decoder in order to request more bits to the encoder. Finally the virtual channel is used to model and calculate the relation between side information and the actual encoded frame. While the two previous channels are real communication channels, the latter is only a theoretical construction.

3. OPTICAL FLOW DRIVEN SIDE INFORMATION GENERATION

The problem of frame interpolation find uses in a number of fields, e.g. video post processing, restoration of historic material, and, the application we will consider here, video coding. For the two former applications, the goal is often to satisfy a viewer, in which case the main concern often is that the results look good,[8] rather than having good performance in terms of a specific error measure. In distributed video coding, however, it is used to generate side information for decoding and performance in terms of specific error measures are more important than crisp results. In ordinary video coding applications discrete methods like block matching has been used very successfully, and variational motion estimation methods have not gained much ground. One reason for this is that optical flow fields are dense, and thus problematic to code. In distributed video coding, however the source statistics are exploited at the decoder side, eliminating the problem of coding the flow field motion vectors. Such a setup makes it possible to exploit the highly accurate motion estimates of modern optical flow methods[4, 26]. We shall extend own previous work on optical flow in DVC, by including a symmetric flow.

3.1 TV-L^1 Optical Flow

Optical flow estimation concerns the determination of apparent (projected) motion. Given a sequence of temporally indexed images I_t, we want to estimate the optical flow $\boldsymbol{v}$ such that the motion matches the image sequence while still maintaining sufficient regularity. Here we will consider a Total Variation (TV)-L^1 energy for the optical flow estimation, which is given by

$$E(\boldsymbol{v}) = \int \|I_{t+1}(\boldsymbol{x}+\boldsymbol{v}(\boldsymbol{x})) - I_t(\boldsymbol{x})\| \,\mathrm{d}\boldsymbol{x} + \int \|D\boldsymbol{v}(\boldsymbol{x})\| \,\mathrm{d}\boldsymbol{x}, \tag{1}$$

where the first term is a L^1 norm of the difference between I_t and the motion-compensated version of I_{t+1}, and the second term is a total variation regularization, which is to be understood as the integral over the Frobenius norm of the derivative of $\boldsymbol{v}$.[9] The total variation regularization will smooth the estimated motion while still allowing for sharp motion boundaries. In order to efficiently minimize E we introduce two relaxations. First we linearize the data fidelity term $I_{t+1}(\boldsymbol{x}+\boldsymbol{v}) - I_t(\boldsymbol{x}) \approx \rho(\boldsymbol{v})(\boldsymbol{x})$, where ρ is the first order Taylor approximation

$$\rho(\boldsymbol{v})(\boldsymbol{x}) = I_{t+1}(\boldsymbol{x}+\boldsymbol{v}_0) - I_t(\boldsymbol{x}) + (\boldsymbol{v}(\boldsymbol{x}) - \boldsymbol{v}_0)^\top \nabla I_1(\boldsymbol{x}+\boldsymbol{v}_0) \tag{2}$$

with $\boldsymbol{v}_0$ being the current estimate of $\boldsymbol{v}$ around $\boldsymbol{x}$. We further relax E by introducing an auxiliary variable $\boldsymbol{u}$ that splits data fidelity and regularization in two quadratically coupled energies:

$$E_1(\boldsymbol{v}) = \int \lambda\|\rho(\boldsymbol{v})(\boldsymbol{x})\| + \frac{1}{2\theta}\|\boldsymbol{v}(\boldsymbol{x}) - \boldsymbol{u}(\boldsymbol{x})\|^2 \,\mathrm{d}\boldsymbol{x}, \tag{3}$$

$$E_2(\boldsymbol{u}) = \int \frac{1}{2\theta}\|\boldsymbol{v}(\boldsymbol{x}) - \boldsymbol{u}(\boldsymbol{x})\|^2 + \|D\boldsymbol{u}(\boldsymbol{x})\| \,\mathrm{d}\boldsymbol{x}, \tag{4}$$

The above type of relaxation was first proposed by Zach et al.[10], and has since been used in a large number of optical flow algorithms.[11, 12] Its most important advantage is that the two problems can easily be solved pointwise which makes the solution very easy to implement on massively parallel processors like graphics processing units (GPUs). The minimizing solutions (3) and (4) will not be replicated here, but we note that the minimizer of (3) can be found by the method of Zach et al.[10] in the case of grayscale images and in the general case of vector valued images the minimizer is explicitly presented in the work of Rakêt et al.[12]. The regularization energy (4) is minimized by the projection method of Chambolle[9, 13]. We have also applied this to DVC[4, 26], but here we select parameters differently.

In order to improve interpolation quality we use a specialized coarse-to-fine pyramidal implementation of the above algorithm (for more details on standard implementations we refer to the works of Rakêt et al.[2, 12]). We have 70 pyramid levels with a scaling factor of 0.95, where each pyramid level is smoothed with a Gaussian with standard deviation $\frac{\sqrt{2}}{4}$ before downscaling to the coarser level. On each level we do 30 warps of first solving (3) and then solving (4) using 10 iterations of the algorithm of Bresson[9], with $\lambda = 3$ and $\theta = 0.2$, where in order to improve interpolation quality, ρ has been weighted by the gradient magnitude $\|\nabla I_1(\boldsymbol{x} + \boldsymbol{v}_0) + 0.01\|$ (slightly shifted to avoid division by 0) in the minimization of (3)[14]. Additional improvement of interpolation quality was found by applying a 3×3 median filter of the flow after upscaling to the next pyramid level[11].

3.2 Frame Interpolation algorithm and results

We are interested in interpolating an in-between frame $I_{1/2}$ using only the two surrounding frames I_0 and I_1. We first note that the optical flow algorithm presented in the previous section is asymmetric, since the (forward) flow estimated from I_0 to I_1 is not the same as the (backward) flow from I_1 to I_0. In addition the forward flow will have a coordinate system corresponding to the pixels in I_0 and the backward flow follows the coordinate system given by the pixels in I_1, so in order to use these flows to interpolate at pixel positions in $I_{1/2}$ we need to temporally warp the flows[15–17] to match the intermediate frame. This is done by assuming that the intermediate frame follows the estimated motion linearly, and then defining the warped forward flow as the flow from $I_{1/2}$ to I_1, which is approximated by

$$\boldsymbol{v}_f^{1/2}(\mathrm{round}(\boldsymbol{x} + {}^1\!/\!_2\boldsymbol{v}_f(\boldsymbol{x}))) = {}^1\!/\!_2\boldsymbol{v}_f(\boldsymbol{x}), \tag{5}$$

where the round function rounds to nearest pixel. The warped backward flow is estimated similarly. This simple warping procedure does however contain some problems, first multiple flow vectors may hit the same pixel $\mathrm{round}(\boldsymbol{x} + {}^1\!/\!_2\boldsymbol{v}_f(\boldsymbol{x}))$ (typically occlusion), which can be dealt with by choosing the vector with best data fidelity. A more serious problem is the problem of dis-occlusion which causes holes in the warped flow. We will correct this by filling holes using an outside-in strategy, however ideally one would reason about depth and occlusion in the interpolation procedure, which should give slightly better results[16].

With the warped flows, the straightforward approach for interpolation is to interpolate along the flow vectors,

$$I_{1/2}(\boldsymbol{x}) = \frac{1}{2}(I_1(\boldsymbol{x} + \boldsymbol{v}_f^{1/2}(\boldsymbol{x})) + I_0(\boldsymbol{x} + \boldsymbol{v}_b^{1/2}(\boldsymbol{x}))), \tag{6}$$

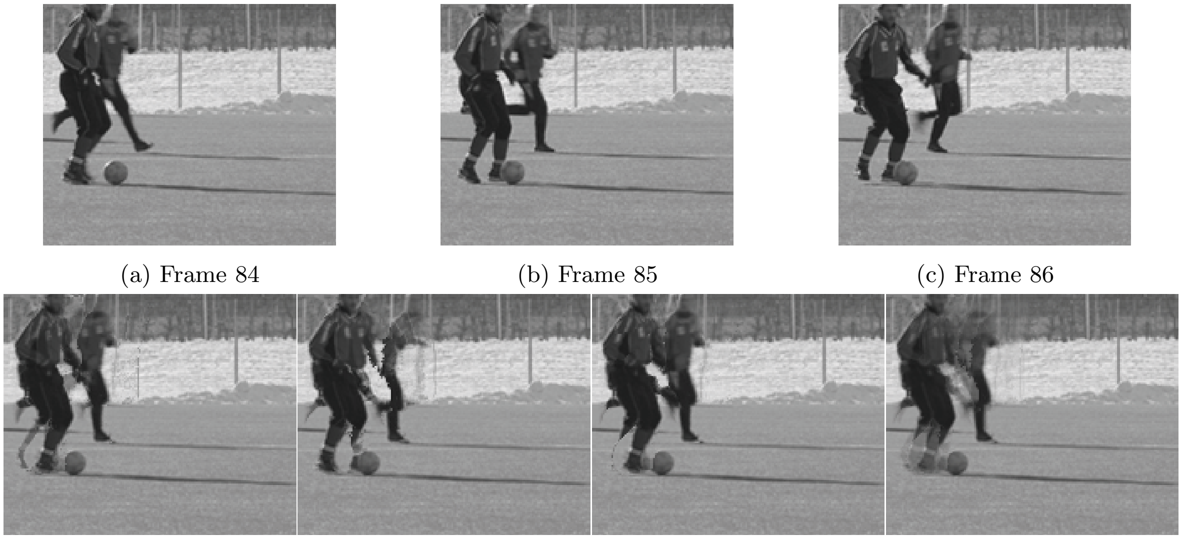
however, since we have discarded occlusion information by filling holes and clearing collisions, the warped forward flow should have been symmetrized, so it can be thought of as a minimizer of $I_1(\boldsymbol{x} + \boldsymbol{v}_f^{1/2}(\boldsymbol{x})) + I_1(\boldsymbol{x} - \boldsymbol{v}_f^{1/2}(\boldsymbol{x}))$, and vice versa for the backward flow. Even though the two computed flows are symmetric around $I_{1/2}$, they will be different since they originated from asymmetric flows. We propose to include a truly symmetric flow estimate which is calculated directly using the pixel positions of the unknown frame $I_{1/2}$, to complement the two asymmetric flows. This flow $\boldsymbol{v}_s$ is calculated using the reparametrization of (3) first suggested by Alvarez et al.[18], and recently analyzed in a frame interpolation setup by Rakêt et al.[2] i.e. replacing the data fidelity term in (3) by

$$I_1(\boldsymbol{x} + \boldsymbol{v}_s(\boldsymbol{x})) + I_1(\boldsymbol{x} - \boldsymbol{v}_s(\boldsymbol{x})) \approx I_1(\boldsymbol{x} + \boldsymbol{v}_0) + I_1(\boldsymbol{x} - \boldsymbol{v}_0) + (\boldsymbol{v}_s(\boldsymbol{x}) - \boldsymbol{v}_0)^\top \left(\nabla I_1(\boldsymbol{x} + \boldsymbol{v}_0) + \nabla I_0(\boldsymbol{x} - \boldsymbol{v}_0)\right). \tag{7}$$

We see that the linearized data fidelity term fits in the setup of Zach et al.[10], and so can be minimized by the formula giving the minimizer of (2). The result will however be different in a number of ways. The motion vectors are now only half size, which makes the method more robust against large deviations. Furthermore the sum of the two gradient terms will make the algorithm more robust to noise, and finally we do not have to do a temporal warping of the flow, in order to use it for interpolation. All in all this produces a more robust flow for interpolation, and combining the symmetric flow with the warped forward and backward flows, we propose to do the interpolation as follows

$$\begin{aligned} I_{1/2}(\boldsymbol{x}) = \frac{1}{6}(&I_1(\boldsymbol{x} + \boldsymbol{v}_f^{1/2}(\boldsymbol{x})) + I_1(\boldsymbol{x} - \boldsymbol{v}_b^{1/2}(\boldsymbol{x})) + I_1(\boldsymbol{x} + \boldsymbol{v}_s(\boldsymbol{x})) \\ + \;\; &I_0(\boldsymbol{x} - \boldsymbol{v}_f^{1/2}(\boldsymbol{x})) + I_0(\boldsymbol{x} + \boldsymbol{v}_b^{1/2}(\boldsymbol{x})) + I_0(\boldsymbol{x} - \boldsymbol{v}_s(\boldsymbol{x}))), \end{aligned} \tag{8}$$

i.e. the interpolation is the average of the two surrounded frames warped to the center using the three different flows. Figure 2 shows the results of the three different types of interpolation, along with the estimate (8). The noise residual frames (in pixel domain) used in the DVC setup are calculated by subtracting the average of the three warped versions of I_0 from the three warped versions of I_1.



(a) Frame 84 (b) Frame 85 (c) Frame 86

(d) Forward interp. PSNR 20.40 (e) Backward interp. PSNR 20.33 (f) Symmetric interp. PSNR 21.45 (g) Average (8) PSNR 21.17

Figure 2: (a)–(c) Frames 84, 85 and 86 of the *Soccer* sequence. (d)–(f) The forward, backward and symmetric parts of (8). (e) The average interpolation (8).

We will evaluate (8) which we will denote 3OF on the test sequences (QCIF, 15 fps) *Coastguard* QP=26, *Foreman* QP=25, *Hall* QP=24 and *Soccer* QP=25, where we interpolate every other frame and compare to the overlapped block motion compensation (OBMC) method of Huang et al.[3] and the TV-L^1 optical flow (OF) method presented by Huang et al.[4]. The results can be found in Table 1 where we see that the proposed method outperforms OBMC and OF on all sequence, with an average increase in PSNR of 1.16 dB over OBMC and 2.14 dB over OF.

Sequence	OBMC[3]	OF[4]	3OF
Coastguard	31.83	30.92	32.59
Foreman	29.26	29.28	30.08
Hall	36.46	32.28	36.91
Soccer	21.30	22.43	23.90

Table 1: Average PSNR across the 74 interpolated frames for the four test sequences.

The SI generated based on the frame interpolation (8) is then used inside the TDWZ decoder together with the OBMC method, for more details please refer to Section 5.

4. ERROR CORRECTION

We now consider the problem of having noise on the transmission of the syndrome bits. We assume that the feedback channel, the transmission channel of the H.264 frames and the transmission of the Cyclic Redundancy Check (CRC) are error free. In LDPCA-based decoders, since the syndromes are error-free they are used to check the results. We relax this condition in order to allow the decoder to accept a result even if the syndrome condition is not satisfied.

4.1 Expanded Code

The most straightforward method to handle errors on the transmission channel is to consider the syndrome bits belonging to the code with parity check matrix H as the last parity bits of another larger code $[H_{m\times n}|I_{m\times m}]$ where $I_{m\times m}$ is the identity matrix. This was proposed by Tan and Li[19] among others. Thus instead of having to fulfill the syndrome conditions $HY = S$, where Y is the side information and S is the syndromes, the new code should fulfill:

$$[H_{m\times n}|I_{m\times m}]\begin{bmatrix} Y_n \\ S_m \end{bmatrix} = 0. \tag{9}$$

This means that instead of only considering Y as a noisy version of the original bitplane X, the received syndromes $\hat{S}$ are also considered as a noisy version of the original syndromes S.

It is well known[20] that there are three major features of the parity check matrix that influence the performance of the message passing algorithm for a LDPC code. The three features are:

1. The weight of each column should be big enough
2. The weight of each row should be small enough
3. The graph of the code should contain no cycles of length four

In a typical DVC setup with a regular LDPCA code the first feature is satisfied for all rates for the original parity check matrix, but when concatenated with the identity matrix a problem arises since each new column only has a weight of one. The second feature is easily satisfied for high rates, but is harder to satisfy for low rates, since the number of rows drops. The third feature is again easily satisfied for high rates but in typical LDPCA codes it is not ensured, even for high rates. For low rates it may be impossible to satisfy. The concatenation with the identity matrix does not change the second and third features. An alternative to item 3 could be that the girth of the corresponding Tanner graph should be big enough. It should be noted that even though these features are well known to influence the performance of a LDPC code we do not have theoretical grounds allowing us to predict the behavior of the modified LDPCA code.

The next two sections will present methods inspired by traditional LDPC codes assuming that the errors on the transmission channel can be considered as a Binary Symmetric Channel (BSC). In Section 4.4, the noise on the transmission channel will be assumed to be Gaussian distributed.

4.2 Bit Flip

Bit flipping methods[21] for LDPC codes are fairly good approximations to the more advanced belief propagation. More advanced variations of this method such as weighted bit flip decoding[22], reliability ratio based weighted bit flip decoding[23] and gradient descent bit flip decoding[24] have also been developed in the recent years.

The main idea behind the methods is that if there is a low enough number of parity checks which fail it might be due to transmission errors. Thus in this case all the syndromes involved in these failed parity checks could be flipped and if the decoding is successful with these new syndromes, it is assumed that the flipping was correct. If the correctness of the decoding is checked by a CRC then it should be noted that each time a sequence of syndromes are flipped the strength of the CRC is in a sense weakened since there is a new risk of decoding into a wrong code word which also satisfies the CRC. Before starting the explanation of the developed methods, it should also be noted that since there are two errors on the syndromes for each error on the accumulated syndromes (unless the errors on the accumulated syndromes are right next to each other) the expected number of errors on the syndromes are approximated by multiplying the expected number of errors on the accumulated syndromes by two.

The first method is the simplest version of this way of thought and it is called “Simple Bit Flip”. Suppose we have received m bits, and P_e is the error probability on the transmission channel and let τ be a small natural number. After running the belief propagation algorithm, if the decoding is not successful, we define with PCF the number of failed parity checks, if $PCF < 2mP_e + \tau$ we flip syndromes involved in failed parity checks and we rerun the belief propagation, after that we again check the syndrome condition and the CRC check. If both are satisfied we accept the word, otherwise we increase the rate.

The second method is inspired by the gradient descent method of Wadayama et al.[24] which outperforms traditional weighted bit flip. In this method the maximal number of expected errors on the syndrome is calculated using the binomial distribution, and after the belief propagation an error function value for each bit is calculated: $E\left(y_i\right) = \lambda y_i \hat{x}_i + \sum_{k \in C} PC_k$ where λ is a weight parameter, y_i is the bit belonging to the SI in bipolar coordinates, $\hat{x}_i$ is the corresponding decoded bit in bipolar coordinates, C is the map of connected parity checks to the current node, PC_k is the value of the parity check in bipolar coordinates. The first term in the error function corresponds to the correlation between the SI word and a codeword while the second term is the sum of the bipolar syndromes. At a given rate, after the first belief propagation, if $PCF \leq m$ where m is the highest number of expected errors with certainty η, we calculate the error term for each bit and with this the reliability of the syndromes. The syndromes having lowest reliability are flipped and the belief propagation is executed again.

4.3 Increased Column Weight (ICW)

In order to improve the aforementioned features various methods have been proposed.[20, 25] We have developed an alternative approach in order to increase the weight of columns with column weight one and disregard cycles of four (since they are present in the original LDPCA code anyway). Our method is outlined in Algorithm 1. It should be noted that the algorithm is only designed for LDPC codes where all columns have a weight above one except for the concatenated identity matrix.

Algorithm 1 Increase Column Weight

```
Let $H_{m \times n}$ be the input parity check matrix and initialize $NM = n + m$ and an all-zero output matrix $H'_{4m \times n+2m}$.
for $i = 1$ to $i = m$ do
    Let the set $O_i$ denote all the positions of 1's in row $i$.
    if any bit in row $i$ is part of a cycle of length four then
        Choose a random element $o_i \in O_i$ which is part of such a cycle.
    else
        Choose a random element $o_i \in O_i$.
    end if
    Let $N = n + i$ and $K = 4(i - 1)$.
    Set the elements indicated by $N$ and $NM + 1$ in row number $K + 1$ of the output matrix to 1.
    Set the elements indicated by $o_i$, $NM + 1$ and $NM + 2$ in row number $K + 2$ of the output matrix to 1.
    Set the elements indicated by $o_i$, $N$ and $NM + 2$ in row number $K + 3$ of the output matrix to 1.
    Set the elements indicated by $o_i \setminus o_i$ and $NM + 2$ in row number $K + 4$ of the output matrix to 1.
    Set $N = N + 2$.
end for
```

4.4 Modifications in Case of Gaussian Errors in Accumulated Syndrome Bits

We assume the noise on the transmission channel to be Gaussian distributed. The error function flip method uses the error probabilities calculated from the soft values of the syndromes. One can calculate the probability of error of the syndromes $P(S_i)$ from the error probability on the accumulated syndromes $P(A_i)$.

The Error Function Flip method is altered to handle soft errors by changing the error function $E(y_i) = -\lambda P(1 - \hat{x}_i | y_i) - \sum_{k \in C} P(S_i)$, where $P(1 - \hat{x}_i | y_i)$ is the probability of the decoded bit to be wrong given the soft value of the received bit and P_e^k is the error probability of a connected syndrome.

The Log-Likelihood Ratio (LLR) values for the syndromes, in this work when using soft errors, are initialized by comparing the magnitude of the current LLR-value (of the accumulated syndrome) and the magnitude of the previous LLR-value (previous syndrome), and then choosing the lower of the two as the magnitude for the current LLR-value of the syndrome. In this way the uncertainty for a syndrome bit is propagated to the next bit to accommodate for the relationship between accumulated syndromes and not accumulated syndromes.

5. RESULTS

5.1 Performance Evaluation for DVC using Optical Flow

This section considers the TDWZ video codec[26] obtained by including the proposed 3OF (Section 3) in our TDWZ codec, which uses a cross-band[3] noise model with clustering[27] techniques in the noise model.

5.1.1 Transform Domain Wyner-Ziv Video using Optical Flow and Clustering

The TDWZ video depicted in Fig. 3 consists of OBMC and the proposed Optical Flow based side information generations (3OF), a noise model (Clustering) using clustering[27], and a cross-band noise model (Cross Band)[3]. The proposed optical flow (3OF) replaces the optical flow of our previous TDWZ codec[26]. The cross band noise model[3] was introduced utilizing cross band correlation based on the previously decoded neighboring bands. The decoder cross band noise model includes a classification module, a bitplane level noise residue refinement, and a modified maximum likelihood estimator to calculate noise parameter. The clustering noise model[27] was utilized to take correlation of DCT coefficients and residues from previously decoded frames into account to estimate the decoding residue more precisely. This noise model estimates the correlation noise by clustering of DCT blocks and using the correlation of neighbor coefficients to refine the Laplacian parameter. Furthermore, the noise model also generates a number of noise residual distributions based on previously decoded frames for adapting of soft side information during decoding.

The architecture of the TDWZ decoder[26] including the proposed 3OF is presented in Fig. 3. The side information generations generate the noise residual frames NR_1, NR_2 and the side information frames, SI_1, SI_2. SI_1 and NR_1 are generated by using OBMC[3] and SI_2 and NR_2 are generated by the proposed 3OF. These are transformed and input to the noise models. For each side information scheme, noise parameters α_{CB} using multiple hypotheses[4] combined with the cross-band[3] and α_{CL} are calculated using the clustering model[27]. Based on the transformed side information frames and the noise parameters, the soft-inputs Pr_{1CB}, Pr_{2CB}, and Pr_{1CL}, Pr_{2CL} are calculated, where Pr_{1CB} and Pr_{2CB} are calculated based on the cross-band noise and multi-hypothesis techniques.[4] Pr_{1CL}, Pr_{2CL} are obtained by applying the clustering model to each side information generation scheme, here OBMC and the proposed 3OF. All soft-inputs are fed into the multiple input LDPCA decoder and the soft-input which converges first is selected for LDPCA decoding. The corresponding selected noise parameter is chosen for reconstruction.

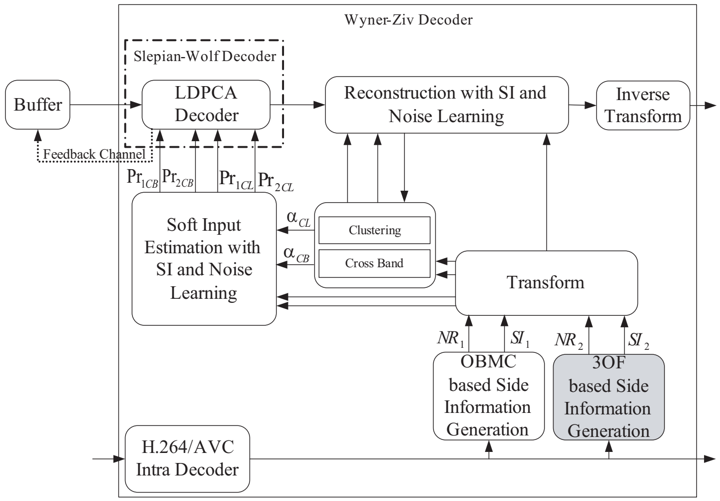


Figure 3: Transform domain Wyner-Ziv video using 3OF Optical Flow.

5.1.2 Performance Evaluation

The rate-distortion (RD) performance of the proposed techniques are evaluated for the test sequences (149 frames of) *Foreman, Hall Monitor, Soccer,* and *Coastguard* with 15Hz frame rate and QCIF format. The GOP size is 2, where odd frames are coded as key frames using H.264/AVC Intra and even frames are coded using Wyner-Ziv coding. Eight RD points are considered corresponding to eight 4 × 4 quantization matrices[7]. The parameters for H.264/AVC Intra are set as by DISCOVER[7] and QP values are set to those used for the key frames in the Wyner-Ziv video coding in the DISCOVER codec[7]. It can be noted that only the luminance component of each frame is evaluated.

Table 2: Bjøntegaard Relative Bit-rate Savings (%) over DISCOVER for WZ Frames

Sequence	Cross-band	Clustering	Multi-hypothesis	TDWZ (3OF)
Foreman	14.0	21.6	27.0	36.0
Hall	8.3	21.0	13.3	26.0
Soccer	26.0	34.5	41.2	63.2
Coast	11.6	21.1	17.4	35.6
Average	15.0	24.6	24.7	40.2

Table 3: Bjøntegaard PSNR Improvement (dB) over DISCOVER for WZ Frames

Sequence	Cross-band	Clustering	Multi-hypothesis	TDWZ (3OF)
Foreman	0.633	0.974	1.177	1.530
Hall	0.370	0.903	0.575	1.095
Soccer	1.305	1.677	1.921	2.782
Coast	0.352	0.637	0.526	1.031
Average	0.665	1.047	1.050	1.610

Tables 2 and 3 report RD performance of the proposed scheme in Section 5.1.1, named TDWZ(3OF). Tables 2 and 3 present the relative average bitrate savings and equivalently the average PSNR improvements (using the Bjøntegaard difference metric[28] and fitting a curve through the 8 RD points measured) over the DISCOVER codec for WZ frames. The results are also compared to the DVC scheme called Cross-band[3]. The TDWZ(3OF) codec based on combining the clustering[27] and multi-hypothesis[4] techniques, which are also individually compared (Clustering[27] and Multi-hypothesis[4]). Compared to DISCOVER, the average bitrate saving for the proposed scheme TDWZ(3OF) is overall (average Bjøntegaard) 40.2% and 16.2% better on WZ frames and all frames, respectively. The performance improvement is 63.2% and 33.6% (or equivalently the average improvement in PSNR is 2.78 dB and 1.56 dB) for WZ frames and overall frames, respectively, for the difficult *Soccer* sequence.

The RD performance of the TDWZ(3OF) codec and H.264/AVC coding is also depicted in Fig. 4 for all frames. The TDWZ(3OF) codec gives a better RD performance than H.264/AVC Intra coding for *Foreman*, *Hall Monitor*, and *Coastguard*, and also better than H.264/AVC No Motion for *Coastguard*. The RD performance of the TDWZ(3OF) codec clearly outperforms those of Cross-band[3] and DISCOVER.

5.2 Error Prone Transmission Channel

In the following sections, results for transmission channels with noise will be presented.

5.2.1 Binary Symmetric Channels

In this section it is assumed that the bit X_i forming the bitplane has equal probability of being 0 or 1 and that the transmission channel and Side Information channel are BSC's. We will refer to the error probability of the SI channel by *crossover probability*. The effect of different parameters will be investigated and the performance of the different methods will be evaluated.

The first two simulations compare the two bit flip methods and the expansion methods and show the influence of λ parameter in the Error Function Flip (EFF) method. The Bit Error Rate (BER) and the rate for different error probabilities on the transmission channel and for two different error probabilities on the SI channel can

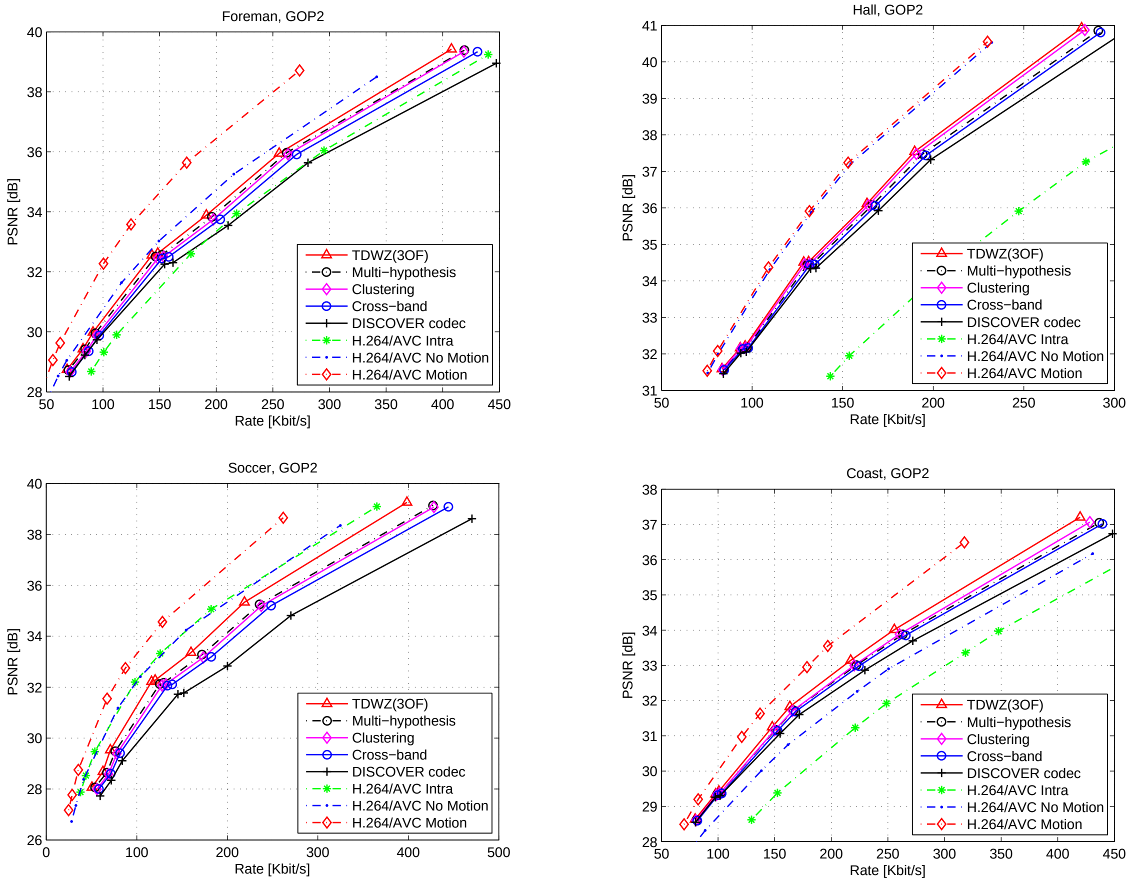


Figure 4: PSNR vs. rate for the proposed TDWZ (3OF) codec for all frames (QCIF, 15Hz, GOP2).

be seen in Fig. 5. It appears that the EFF method has better performance than the Simple Flip for high error probability on the transmission channel. It can also be seen that the λ parameter has a very low impact on the performance of the EFF method, but the best performance is for very low λ parameters which suggests that it is better to disregard the correlation between a received word and a codeword than taking the correlation into account. It is apparent that the expansion methods usually outperform the flipping methods. It also appears that for the good SI the ICW method is performing better than the expansion method[19] with regard to BER. In regards to bitrate the ICW method also outperforms the expansion method when the SI is good, except when there are no errors on the transmission channel.

5.2.2 Gaussian Transmission Channel

In this section the two expansion methods are tested and evaluated in DVC simulations. The simple expansion method is also applied to SI estimated by the 3OF method as described in Section 3. To save computation time, the simulation with 3OF SI has been conducted with SI already calculated in a DVC simulation without channel errors. Thus errors can not propagate down through the bitplanes and the PSNR cannot be calculated. We therefore assume that the PSNR is the same for these simulations as their normal SI counterparts. The simple expansion method is also benchmarked against turbo coding. The noise in the transmission channel is assumed to be Gaussian distributed. Only four different RD points corresponding to four quantization levels are used since they seem to match a concave function in rate-distortion sense.

The rate-distortion plots for the four test sequences appear in Fig. 6 with no errors (NE) on the transmission channel (the punctured lines), with a standard deviation of the Gaussian distribution to match the error probability of $P_e = 0.001$ (dotted lines) and $P_e = 0.01$. From the theoretical point of view we define P_e as the error

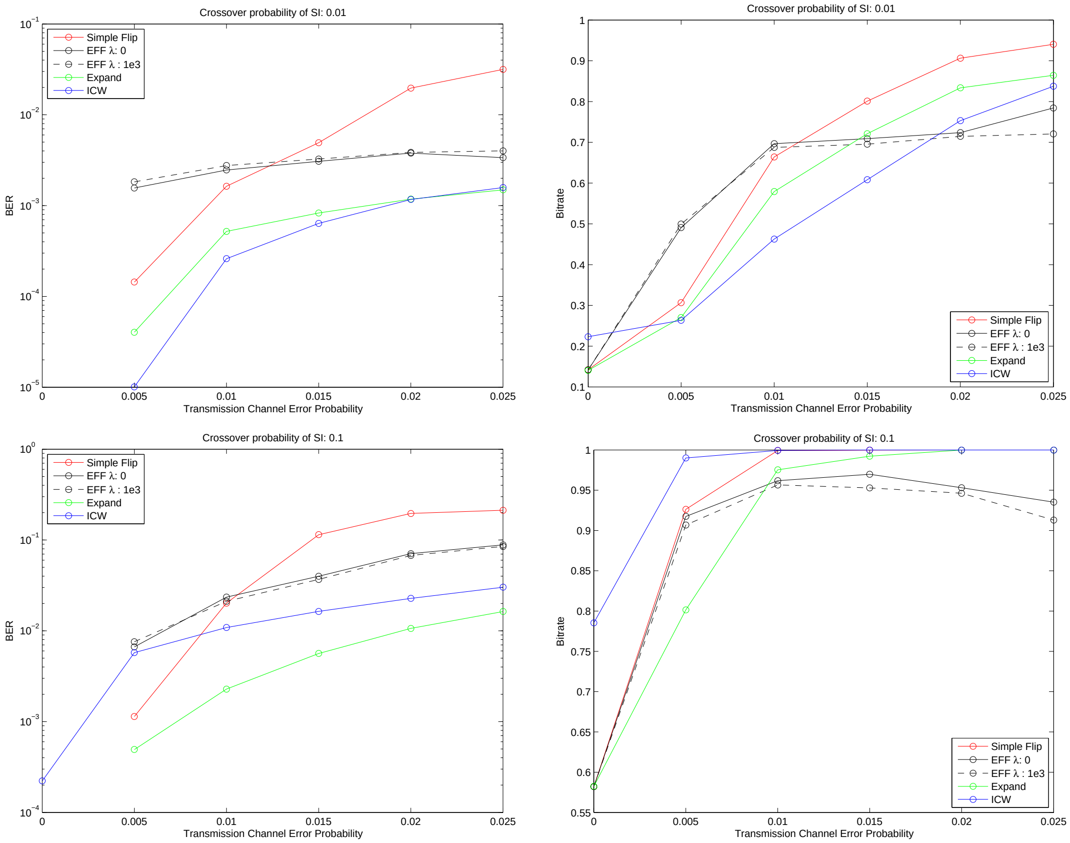


Figure 5: Results for the BSC transmission channel, modeling the SI channel also as BSC.

probability of the Gaussian channel followed by a threshold detector having the threshold at the same distance from the two symbols used. It is evident that the simple expansion method performs best overall, and in most cases the simple expansion with noise is also better than ICW with no noise. It also appears that with 3OF SI the bitrates are lower than a normal SI for the same error rates as expected.

To test the robustness of the LDPC code used versus the robustness of a Turbo code, simulations have been performed with $P_e = 10^{-2}$ (Fig. 7) and with $P_e = 10^{-3}$ (Fig. 8), for the Turbo code 25 iterations are waited before trying the CRC check for the first time. Since the initial simulations showed that the Turbo code depended heavily on the CRC both an 8-bit and a 16-bit CRC has been used in the simulations. From the rate-distortion plots it appears that for an 8-bit CRC the Turbo code has many decoding errors. If the CRC is increased to 16-bit though, the Turbo code has better performance than the LDPCA using the same CRC, which does not improve by the stronger CRC. It has to be noted however that in absence of errors the LDPCA codes outperform Turbo codes. In the presence of errors, 16-bit CRC Turbo coding is better in all the sequences except *Hall* in the case of $P_e = 10^{-2}$. In the case of $P_e = 10^{-3}$ for *Hall* and *Coast* the LDPCA codes outperforms Turbo coding, while on the other two sequences the situation is inverted. The explanation may be that the LDPCA code is built on a Rate 1/2 LDPC code while the Turbo is built on a Rate 1/3 code, thus in high bit rate cases the 1/2 rate LDPC may not provide enough redundancy to correct both errors in the SI and the transmission of syndrome bits.

It can also be noted that in some cases a drop in the PSNR is experienced while increasing the quality level, i.e. increasing the number of bits sent does not improve the PSNR. A possible explanation is that, since increasing the quality is done by increasing the number of sent LSB bitplanes, these new and high error-prone bitplanes increase the number of wrongly decoded bitplanes. Hence skipping a bitplane (i.e. not sending it and using the SI bitplane as substitute) could improve the results, achieving a lower rate and sacrificing PSNR performance in the case of a possible correct decoding. In Table 4, the results are presented for a system in which skipping was

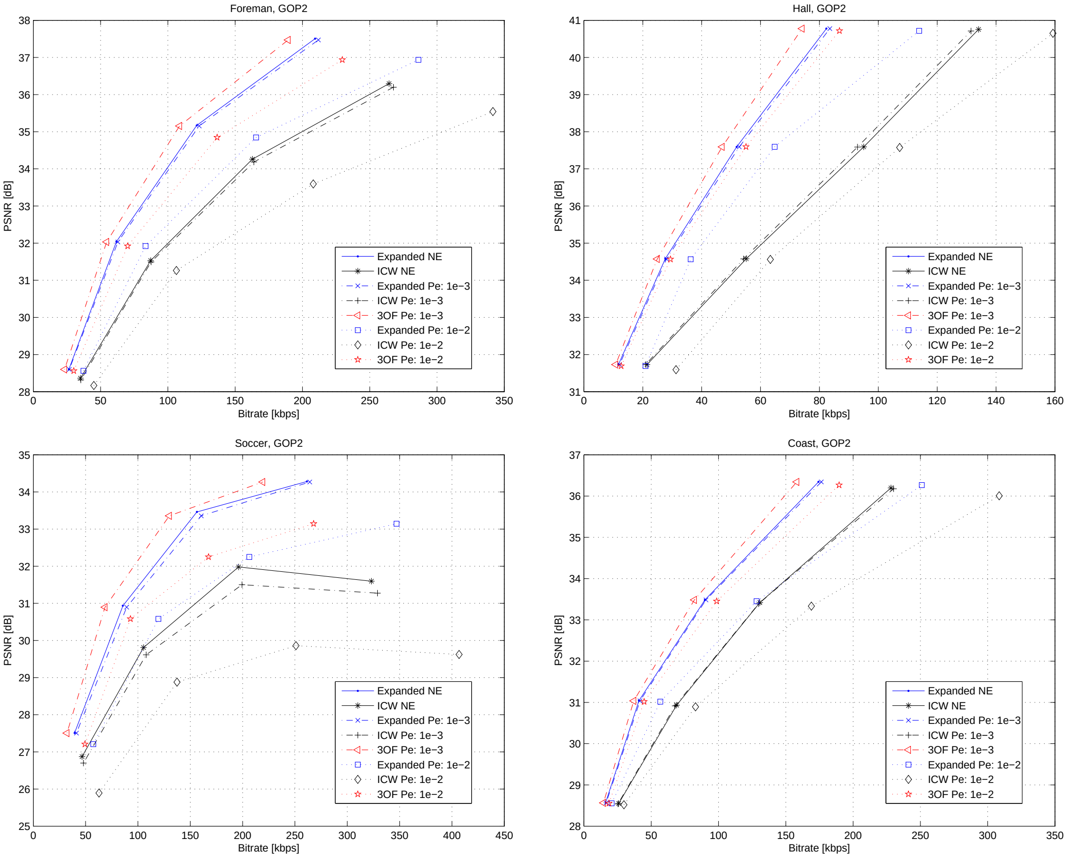


Figure 6: Rate-distortion plots with errors and no errors (NE) on the transmission channel

only allowed for the LSB and it was done only if the estimated conditional entropy is higher than a predefined threshold with $P_e = 10^{-2}$. The results are presented using the Bjøntegaard difference metric between an 8-bit CRC LDCPA-based expanded decoder and the same decoder with the skip strategy implemented. Indeed the skipping improves the performance.

Table 4: Bjøntegaard PSNR and bitrate Improvement over the non-skip decoder for WZ Frames

Sequence	PSNR Difference [dB]	Bit-rate Savings (%)
Foreman	0.61	13.12
Hall	1.09	23.64
Soccer	0.70	13.30
Coast	0.77	15.73

6. CONCLUSION AND DISCUSSION

A new method for side information generation in a DVC setup is presented. The method has been shown to consistently outperform the previously suggested methods, while at the same time being computationally efficient. The novelty of the interpolation method is a setup which includes a symmetric optical flow constraint in the interpolation, and a specialized setup in the motion estimation process, that produces estimates well suited for interpolation purposes. The addition of a symmetric term is not tied to the specific setup, nor the chosen algorithm (TV-L^1), and can easily be incorporated in most motion estimation algorithms, at low cost in terms of computation. A further gain in interpolation accuracy may be obtained from using anisotropic regularization

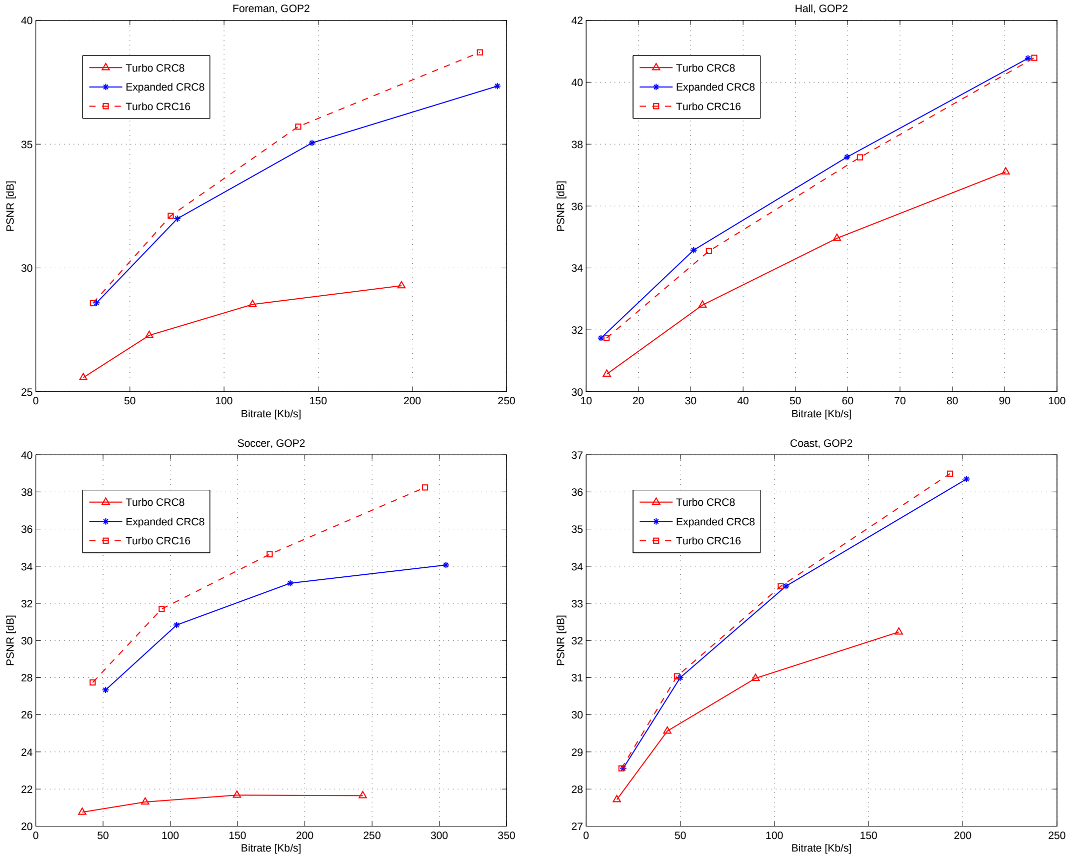


Figure 7: Rate-distortion plots of Turbo and LDPC decoding with 8-bit and 16-bit CRC, with $P_e = 10^{-2}$

instead of TV regularization. In particular, the anisotropic Huber-L^1 algorithm of Werlberger et al.[11] has proved to give good interpolation results[15]. Alternatively one may introduce anisotropy by adaptively adjusting the smoothness weight locally[29], which has recently shown to improve interpolation performance.[15]

In addition we have considered using the Slepian-Wolf decoder to handle transmission errors . Simple bit flip methods are presented to add robustness to the LDPC code in DVC. These methods are simple alternatives to methods where the decoding matrix has to be modified, but the latter shows better performance. Our simulations have shown that there is a difference in performance when assuming a BSC as the transmission channel versus a transmission channel with Gaussian distributed noise. In the BSC case our ICW method outperformed the expanded method when the SI was good, but when the noise in the transmission channel was assumed to be Gaussian distributed the expanded method was the best choice for all of the four test sequences. Our simulations also indicate that the bitrate is still improved when using the 3OF SI and the expanded method with an erroneous transmission channel. Further work with robustness for LDPC in DVC could focus on combining LDPC codes optimized for different intervals of the rate where a PEG-like approach[30] is used to make the LDPC codes rate-adaptive.

The LDPCA code was compared with Turbo coding for DVC. Without errors on parity bits/syndromes LDPCA was the best performing decoder. In the error case, Turbo coding (with a 16 bit CRC) performed best in the high-motion sequences, due to a lower maximum level of redundancy in the investigated LDPCA code. Finally, a proof-of-concept of a decoder-driven skip strategy was presented as a possible remedy to the weakness of the LDPCA code, showing promising results.

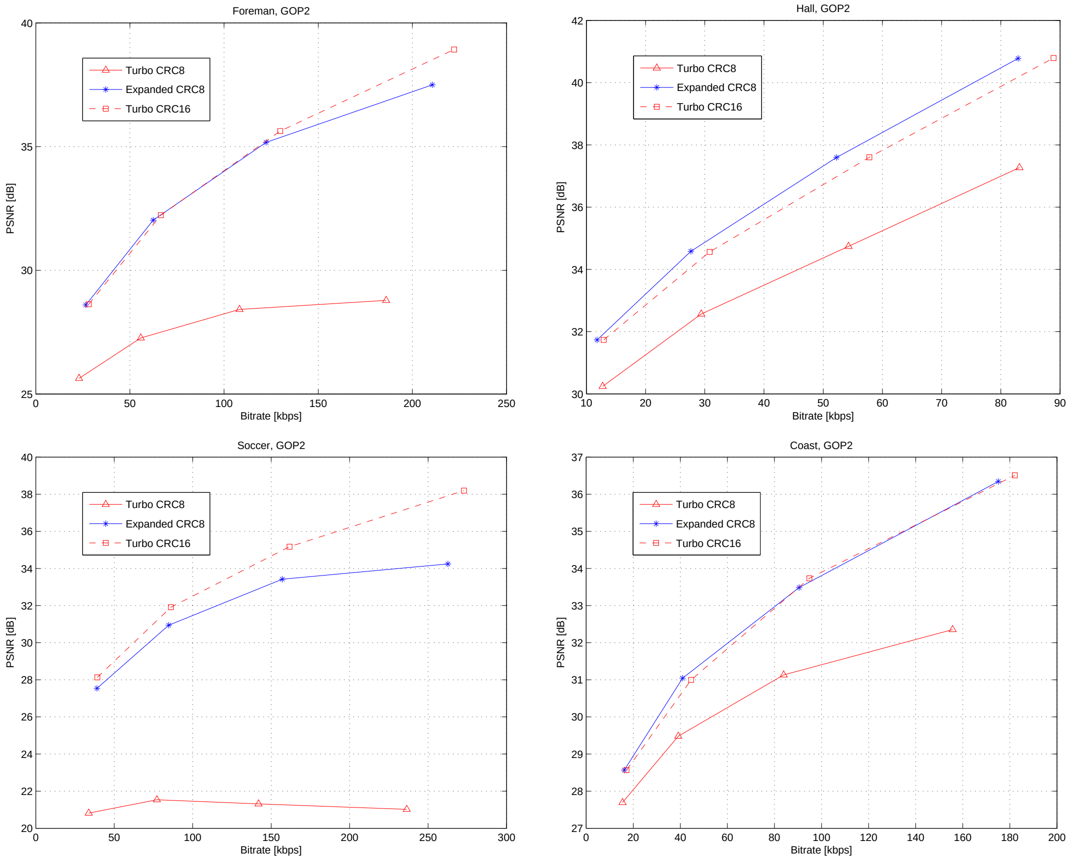


Figure 8: Rate-distortion plots of Turbo and LDPC decoding with 8-bit and 16-bit CRC, with $P_e = 10^{-3}$

REFERENCES

[1] Varodayan, D., Aaron, A., and Girod, B., "Rate-adaptive codes for distributed source coding," *Signal Process.* **86**, 3123–3130 (nov 2006).

[2] Rakêt, L., Roholm, L., Bruhn, A., and Weickert, J., "Motion compensated frame interpolation with a symmetric optical flow constraint," in [*International Symposium on Visual Computing (ISVC)*], (2012).

[3] Huang, X. and Forchhammer, S., "Cross-band noise model refinement for transform domain Wyner-Ziv video coding," *Signal Processing: Image Communication* **27**, 16–30 (2005).

[4] Huang, X., Raket, L., Luong, H. V., Nielsen, M., Lauze, F., and Forchhammer, S., "Multi-hypothesis transform domain wyner-ziv video coding including optical flow," in [*Multimedia Signal Processing (MMSP), 2011 IEEE 13th International Workshop on*], 1 –6 (oct. 2011).

[5] Yasakethu, S. L. P., Weerakkody, W. A. R. J., Fernando, W. A. C., Pereira, F., and Kondoz, A. M., "An improved decoding algorithm for dvc over multipath error prone wireless channels," *IEEE Trans. on Circuits and System for Video Tech.* **19**(10), 1543–1548 (2009).

[6] Girod, B., Aaron, A., Rane, S., and Rebollo-Monedero, D., "Distributed video coding," *Proc. of IEEE (Special issue on advances in video coding and delivery)* **93**(1), 71–83 (2005).

[7] Artigas, X., Ascenso, J., Dalai, M., Klomp, S., Kubasov, D., and Ouaret, M., "The discover codec: Architecture, techniques and evaluation," in [*Proc. Picture Coding Symposium (PCS)*], (2005).

[8] Keller, S., Lauze, F., and Nielsen, M., "Temporal super resolution using variational methods," in [*High-Quality Visual Experience: Creation, Processing and Interactivity of High-Resolution and High-Dimensional Video Signals*], Mrak, M., Grgic, M., and Kunt, M., eds., Springer (2010).

[9] Bresson, X. and Chan, T., "Fast dual minimization of the vectorial total variation norm and application to color image processing," *Inverse Problems and Imaging* **2**(4), 455–484 (2008).

[10] Zach, C., Pock, T., and Bischof, H., "A duality based approach for realtime TV-L^1 optical flow," in [*Pattern Recognition*], Hamprecht, F., Schnörr, C., and Jähne, B., eds., *Lecture Notes in Computer Science* **4713**, 214–223, Springer (2007).

[11] Werlberger, M., Trobin, W., Pock, T., Wedel, A., Cremers, D., and Bischof, H., "Anisotropic Huber-L1 optical flow," in [*British Machine Vision Conference (BMVC)*], (2009).

[12] Rakêt, L., Roholm, L., Nielsen, M., and Lauze, F., "TV-L^1 optical flow for vector valued images," in [*Energy Minimization Methods in Computer Vision and Pattern Recognition*], Boykov, Y., Kahl, F., Lempitsky, V., and Schmidt, F., eds., *Lecture Notes in Computer Science* **6819**, 329–343, Springer (2011).

[13] Chambolle, A., "An algorithm for total variation minimization and applications," *Journal of Mathematical Imaging and Vision* **20**, 89–97 (2004).

[14] Zimmer, H., Bruhn, A., and Weickert, J., "Optic flow in harmony," *International Journal of Computer Vision* **93**, 368–388 (2011).

[15] Baker, S., Scharstein, D., Lewis, J. P., Roth, S., Black, M. J., and Szeliski, R., "A database and evaluation methodology for optical flow," *International Journal of Computer Vision* **31**(1), 1–31 (2011).

[16] Herbst, E., Seitz, S., and Baker, S., "Occlusion reasoning for temporal interpolation using optical flow," Tech. Rep. UW-CSE-09-08-01, Department of Computer Science and Engineering, University of Washington (2009).

[17] Werlberger, M., Pock, T., Unger, M., and Bischof, H., "Optical flow guided TV-L^1 video interpolation and restoration," in [*Energy Minimization Methods in Computer Vision and Pattern Recognition*], Boykov, Y., Kahl, F., Lempitsky, V., and Schmidt, F., eds., *Lecture Notes in Computer Science* **6819**, 273–286, Springer (2011).

[18] Alvarez, L., Castao, C., Garca, M., Krissian, K., Mazorra, L., Salgado, A., and Sánchez, J., "Symmetric optical flow," in [*Computer Aided Systems Theory–EUROCAST 2007*], Daz, R., Pichler, F., and Arencibia, A., eds., *Lecture Notes in Computer Science* **4739**, 676–683, Springer (2007).

[19] Tan, P. and Li, J., "Enhancing the robustness of distributed compression using ideas from channel coding," in [*IEEE GLOBCOM*], (2005).

[20] Heidarzadeh, A. and Lahouti, F., "On robust syndrome-based distributed source coding over noisy channels using LDPC codes," in [*IEEE Int. Conf. on Signal Proc. and Comm.*], (2007).

[21] Gallager, R., "Low density parity check codes," tech. rep., M.I.T. Press (1963).

[22] Kou, Y., Lin, S., and Fossorier, M., "Low density parity check codes based on finite geometries: A rediscovery and new results," *IEEE Trans. on Inform. Theory* **47**(7), 2711–2736 (2001).

[23] Guo, F. and Hanzo, L., "Reliability ratio based weighted bit-flipping decoding for LDPC codes," in [*IEEE Vehicular Technology Conf.*], **61**(1), 709–713 (2005).

[24] Wadayama, T., Nakamura, K., Yagita, M., Funahashi, Y., Usami, S., and Takumi, I., "Gradient descent bit flipping algorithms for decoding LDPC codes," *IEEE Trans. on Comm.* **58**(6), 1610–1614 (2010).

[25] Yedidia, J., Chen, J., and Fossorier, M., "Generating code representations suitable for belief propagation," in [*Proc. 40th Allerton Conf. on Comm., Control and Computing*], (2002).

[26] Luong, H. V., Rakêt, L., Huang, X., and Forchhammer, S., "Side information and noise learning for distributed video coding using optical flow and clustering," *Submitted to IEEE Trans. Image Proc.* (2012).

[27] Luong, H. V. and Forchhammer, S., "Noise residual learning for noise modeling in distributed video coding," in [*Picture Coding Symposium*], (2012).

[28] Bjøntegaard, G., "Calculation of average PSNR differences between RD curves," tech. rep., VCEG Contribution VCEG-M33 (apr 2001).

[29] Rakêt, L., "Local smoothness for global optical flow," in [*International Conference of Image Processing (ICIP)*], (2012).

[30] Jang, M., Kang, J., and Kim, S., "A design of rate-adaptive LDPC codes for distributed source coding using peg algorithm," in [*The 2010 Military Comm. Conf. - Waveforms and Signal Proc. Track*], 277 – 282 (2010).