INVERSE BIT PLANE DECODING ORDER FOR TURBO CODE BASED DISTRIBUTED VIDEO CODING

Yuri Vatis, Student Member IEEE, Sven Klomp, Student Member IEEE, Jörn Ostermann, IEEE Fellow

Institut für Informationsverarbeitung
Leibniz Universität Hannover, Appelstr. 9A, 30167 Hannover, Germany

ABSTRACT

For Turbo Code based Wyner-Ziv codecs, we propose to use an inverse bit plane decoding order. The investigations have shown that the knowledge about MSB's has only a marginal influence on LSB's, which can be regarded as noise, while the influence of LSB's on MSB's tends to be higher. As a results, we obtain a coding gain of up to 0.3 dB, especially for the sequences, which can only be decoded with a coding efficiency lower than H.264/AVC intra. Furthermore, the distribution of the requests for additional parity bits over the back channel is different compared to the conventional decoding order, resulting in a high number of requests for the last bit plane and a low number of requests for the other bit planes. Therefore, the decoder can request several puncturing levels for LSB's in one step, reducing the number of turbo decoding loops. Hence, transmitting the LSB's first, we can obtain a decoding speedup of 30%, thus making DVC more attractive for real-time applications.

Index Terms— Distributed Video Coding, Slepian-Wolf, Wyner-Ziv.

1. INTRODUCTION

Current video coding solutions, such as MPEG or ITU-T H.26x standards, perform well for broadcasting, streaming and other applications, wherein a video is encoded once and decoded several times. The encoder of such a solution exploits the source statistics, whereby the decoder can be kept very simple. For opposite scenarios with many encoders, Distributed Video Coding (DVC) might be more suitable than conventional video coding since the decoder performs the complex task of exploiting the source statistics.

DVC is based on the Slepian-Wolf [1] and Wyner-Ziv [2] theorems. These theorems state that it is possible to compress two statistically dependent signals in a distributed way (separate encoding, joint decoding) using a rate equal to that used in a system, where the signals are encoded and decoded together.

A general block diagram of an unsymmetrical Wyner-Ziv video codec, where the two signals are coded with different bit rates, is shown in Figure 1. At the encoder, the sequence is divided into key frames and Wyner-Ziv frames controlled by the group-of-picture (GOP) size (e.g. at GOP size 4, every fourth frame is coded as key frame). The key frames (here, X_{2i-1} and X_{2i+1} for GOP size 2) are coded with a H.264/AVC intra frame coder and the Wyner-Ziv frames with a distributed coder. Starting from key frames, the decoder generates the side information for the Wyner-Ziv frames. For that purpose, the framework presented in [3] and enhanced in [4] is used. The side information, transformed in the frequency domain, is fed to the turbo decoder, which regards it as corrupted information bits. Encoder and decoder group all transform coefficients at identical positions in a 4x4 transform block, forming so called transform bands. The decoding process is carried out for each band bit plane wise, starting from the MSB's and ending with LSB's. Unless a particular residual error probability is achieved, the decoder requested further puncturing levels of parity bits one by one using a back channel. The reconstruction block takes the quantiser index for inverse quantisation. The representative value is adjusted based on the side information. In this paper we show that in case the side information cannot be estimated properly due to very complex video content, the inverse decoding order results in higher coding gains. The paper is organised as follows: Section 2 describes the state of the art decoding. The proposal is described in Section 3. The speedup improvement is given in Section 4. The results are discussed in Section 5. The paper finishes with conclusions.

2. STATE OF THE ART DECODING IN DISCOVER

The DISCOVER codec (see Section 7) currently uses a Laplacian distribution to model the correlation noise. The error/noise distribution between the corresponding DCT bands of the side information and the original Wyner-Ziv frames is estimated based on the side information. The Laplacian distribution parameter is estimated online at the decoder spatially and temporarily, i.e. for each transform coefficient along the video sequence. Since the side information is generated through a block-based motion compensated algorithm, which may fail for some blocks or regions due to occlusions or other motion estimation errors, significant coding gains are achieved using different α parameter of a Laplacian distribu-

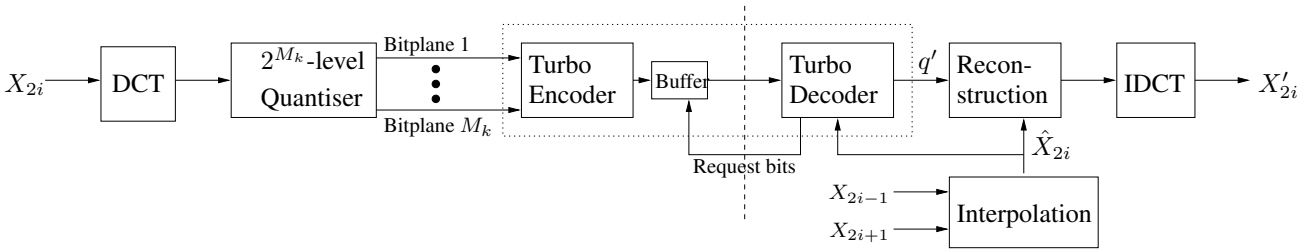

Figure 1: DVC Architecture

tion (Equation 1) for different regions and time instants. For more details, we refer at this stage to [5].

$$p_x(x) = \frac{\alpha}{2} e^{-\alpha|x|} \tag{1}$$

Thus, given for each transform coefficient the corresponding α parameter, the Laplacian distribution centred around the side information value is uniquely defined. Turbo decoding is carried out bit plane wise, starting with the MSB's of each band and finishing with LSB's of this band. A simplified example of such decoding is depicted in Figure 2, assuming that the original transform coefficient is quantised with 3 bits only. First, the Laplacian distribution is assumed to have an α parameter and centred around the side information value. Then probabilities for the MSB having the values 0 or 1 for each coefficient of the band b are computed. In our example, this corresponds to the integral of the probability density function (pdf) from 0 to 3 and from 4 to 7, respectively. These probabilities are fed to the turbo decoder as soft values. Via the feedback channel the turbo decoder can now request further parity bits for the currently processed bit plane, if necessary. In our example the decoder has the 1 as output. This means, the particular transform coefficient is in the right half, $t_b \in [4;7]$. To decode the second bit plane, only the area below the relevant part of the probability distribution function is normalised and fed to the turbo decoder. Thus, the decoder has to decide, whether $t_b \in [4;5]$ or $t_b \in [6;7]$. In our example the decoded value is 0, representing the first case. In the last bit plane, the decoder decides between 4 and 5. Finally, the most probale value of the transform coefficient is reconstructed from the decoded quantised value t_b.

3. PROPOSED DECODING ORDER

The main contribution of this paper is a proposal to use an inverse decoding order, which is explained in Figure 3 with the example from the previous Section. First, for each transform coefficient t_b the area is calculated, where LSB is 0: $t_b \in [0] \cup [2] \cup [4] \cup [6]$ and the area, where LSB is 1: $t_b \in [1] \cup [3] \cup [5] \cup [7]$. In our example, the output after turbo decoding is 0. Now, the decoder has to decide, whether $t_b \in [0] \cup [4]$ or $t_b \in [2] \cup [6]$.

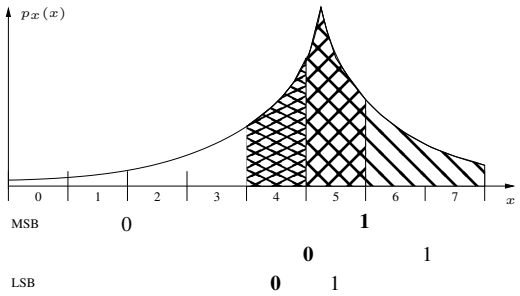

Figure 2: Decoding example starting from the MSB. The distribution is centred on the value of the side information. The dynamic range of a coefficient is quantised into 8 intervals.

Here, we want to show that in case the assumption of the Laplacian model is correct, the decoding order has no influence on coding efficiency, if one does not consider the residual error probability. Although the separate bit planes are independent of each other during the acquisition process, they become dependent at the decoder side as soon as side information is available. Therefore, the entropy of a particular coefficient consisting of e.g. 3 bit planes can be expressed as

$$H(t_b) \quad = \quad H(x_1, x_2, x_3 | s_i) \tag{2}$$

with x_1 representing MSB of t_b, x_4 representing LSB of t_b and s_i representing current state described by the estimated side information. If the decoding starts with the MSB, Equation 2 can be split into

$$H(t_b) = H(x_1|s_i) + H(x_2|x_1, s_i) + H(x_3|x_2, x_1, s_i) \tag{3}$$

which is the same as for decoding the LSB first:

$$H(t_b) = H(x_3|s_i) + H(x_2|x_3, s_i) + (x_1|x_2, x_3, s_i) \tag{4}$$

Therefore, the entropy of t_b is independent of the bit plane ordering. However, the turbo decoder does not guarantee that the decoded bit stream is correct, therefore much more important is to find, when the turbo decoder achieves a particular level of fidelity. The results, given in Section 5 will show that the proposed method provides a better coding efficiency.

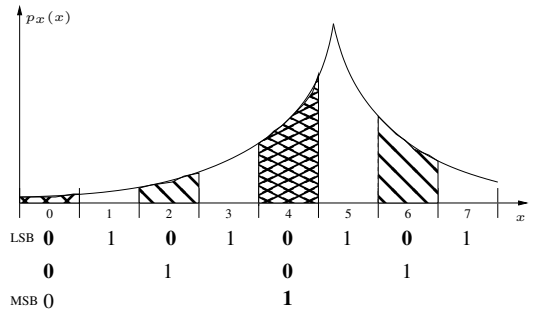

Figure 3: Decoding example starting from the LSB. The distribution is centred on the value of the side information. The dynamic range of a coefficient is quantised into 8 intervals.

4. SPEEDUP IMPROVEMENT

In most publications on DVC, it was always assumed that the encoder has very low complexity, while the decoder might be arbitrarily complex. This however, excludes all real-time applications from DVC. Here, we show that applying an inverse decoding order results in reduced decoding complexity.

Let us consider the case, when decoding from MSB's first, assuming that the side information has a high degree of fidelity. In this case it is difficult to predict, which bit plane will require the highest amount of data, i.e. the highest number of puncturing levels. E.g. in case the side information is exactly in the centre of the entire range, the probabilities for 0 and 1 at MSB are nearly equal, resulting in high number of requested puncturing levels for this bit plane. However, when decoding LSB's first, we force that the probabilities for 0 and 1 are nearly equal independent of the side information. Thus, we artificially transfer a significant part of puncturing levels to the last bit plane. Remembering that the entropy does not depend on the decoding order, we automatically get a reduced number of puncturing levels for all other bit planes. Therefore, the decoding time can be reduced while keeping coding efficiency, when requesting several puncturing levels of parity bits at once for LSB's instead of trying to decode one by one.

In Table 1, the variance and the average number of requested puncturing levels for coarsely quantised DC coefficients are shown for the test sequence *Flowergarden*, QCIF, 15 Hz. This sequence represents the case, when the side information is estimated very accurately. Without analysing coding performance, one can see that the average number of requests is nearly equal (9.69 when starting from MSB and 10.39 when starting from LSB, respectively). However, the distribution of the requests for different bit planes is different. In the second case it is better predictable due to lower variance. Thus, after decoding the LSB, almost no requests follow for the other bit planes due to very accurate side information and consequently narrow pdf.

Also in the case the side information estimation partially fails, the speedup as described above can be realised. Concerning the coding efficiency, we should compare, when the decoder achieves a particular level of residual error probability and not the entropy only. Decoding from MSB's to LSB's results always in two neighbouring intervals, between which the decoder has to decide in the next bit plane. Thus, if the side information is far away from the original value, there is nearly no difference in probabilities between two neighbouring intervals due to assumed Laplacian distribution. On the other hand, decoding LSB's first results always in two intervals for MSB's, the distance between which is exactly half of the entire range. This results in a more significant difference of probabilities and consequently in a lower number of requested puncturing levels, before achieving a particular residual error probability.

	MSB first		*LSB first*	
Bit plane	*Average*	*Variance*	*Average*	*Variance*
1 (MSB)	5.26	1.57	0.00	0.00
2	1.92	1.12	0.02	0.01
3	0.11	0.13	0.48	0.65
4 (LSB)	2.40	1.03	9.89	3.84
Sum	9.69		10.39	

Table 1: Average number of requested puncturing levels and their variances for different bit planes of sequence *Flowergarden*.

In Table 2, the variance and the average number of requests for coarsely quantised DC coefficients is shown for test sequence *Soccer*, representing the case, when side information estimation fails. Here, the total number of requests and thus bit rates differ (58.28 vs. 40.21), resulting in a reduced number of requested puncturing levels and consequently in a reduced data rate as well as reduced decoding complexity. Furthermore, one can see that in case of decoding MSB's first, the percentage of requests for particular bit planes is completely different for different sequences. In case of decoding LSB's first, the tendency is the same as for the test sequence *Flowergarden*. Therefore, further complexity reductions is achieved, when requiring several puncturing levels for LSB's via one request. In order to make the approach generally applicable, the number of puncturing levels for LSB's was chosen considering sequences with reliable side information such as *Flowergarden*, *Coastguard* etc. This was done by approximating the distribution of the requests by the Gaussian distribution and setting the minimum number of requests to 5% of the area of the distribution density function:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2} \tag{5}$$

where σ is the standard deviation of the number of requested puncturing levels for a particular bit plane. In case no motion was detected at all (as often the case e.g. in *Hall&Monitor*), the decoder does not request several puncturing levels at once.

	MSB first		*LSB first*	
Bit plane	*Average*	*Variance*	*Average*	*Variance*
1 (MSB)	15.92	17.99	2.15	11.11
2	25.46	10.50	5.74	21.73
3	7.53	21.90	9.85	26.13
4 (LSB)	9.37	42.65	22.47	17.10
Sum	58.28		40.21	

Table 2: Average number of requested puncturing levels and their variances for different bit planes of sequence *Soccer*.

5. EXPERIMENTAL RESULTS

In our experiments we coded several QCIF, 15 Hz sequences. In Figures 4 and 5, rate distortion performance for two representative test sequences *Flowergarden* and *Soccer* are depicted. As expected for the sequences with inaccurately estimated side information as *Soccer* or *Football*, the coding gains of up to 0.3 dB are achieved. For sequences with accurately estimated side information as e.g. *Flowergarden* or *Hall&Monitor*, the coding gains are smaller.

Concerning the decoding time, an average speedup of 30% is achieved while keeping coding efficiency. By the sequences with the reliable side information, the speed up is mostly achieved due to requesting several puctioring levels at ones, for other sequences it is more due to faster convergency.

The main drawback of the approach is the SNR scalability loss - one needs to obtain all bit planes in order to reconstruct the transform coefficients. Conventionally, the coarse reconstruction can be carried out after decoding of each bit plane.

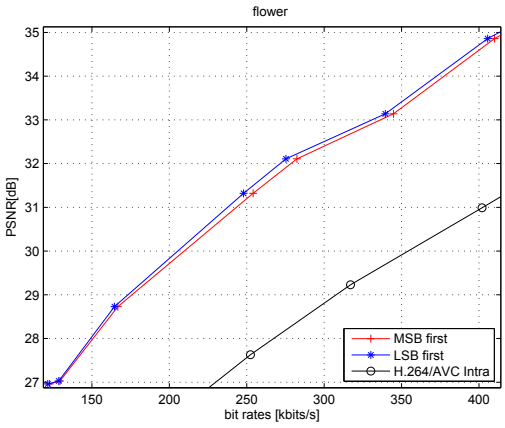

Figure 4: Rate distortion performance, provided by the H.264/AVC Intra (black), by the reference codec (red) and by enhanced codec (blue) for QCIF-sequence *Flowergarden*.

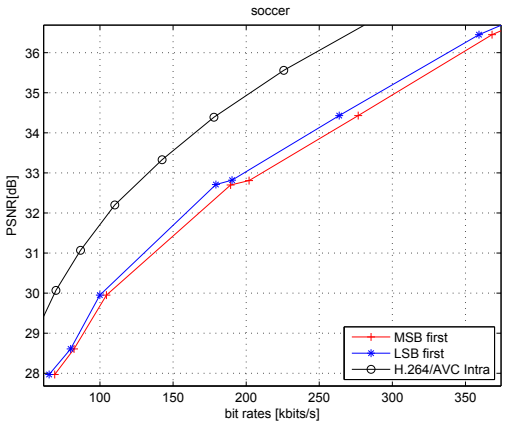

Figure 5: Rate distortion performance, provided by the H.264/AVC Intra (black), by the reference codec (red) and by enhanced codec (blue) for QCIF-sequence *Soccer*.

6. CONCLUSIONS

In this paper, an inverse decoding order is presented. For sequences, where side information estimation fails, which is one of the most important drawbacks of DVC, coding gains of up to 0.3 dB are achieved. Another important improvement is a speedup of averagely 30%, which is an important step to real-time applications using DVC scenarios. The main drawback of this approach is the SNR scalability loss.

7. ACKNOWLEDGEMENTS

We are thankful to the IST and the Discover software development team for having provided the DVC codec software.

8. REFERENCES

[1] J. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. on Inform. Theory*, vol. 19, no. 4, pp. pp. 471–480, July 1973.

[2] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Trans. on Inform. Theory*, vol. 22, no. 1, pp. pp. 1–10, January 1976.

[3] J. Ascenso, C. Brites, and F. Pereira, "Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding," in *5th EURASIP*, Slovak Republic, July 2005.

[4] S. Klomp, Y. Vatis, and J. Ostermann, "Side information interpolation with sub-pel motion compensation for wyner-ziv decoder," in *5th EURASIP*, Portugal, August 2006.

[5] C. Brites, J. Ascenso, and F. Pereira, "Studying temporal correlation noise modeling for pixel based wyner-ziv video coding," in *IEEE International Conference on Image Processing (ICIP 2006)*, Atlanta, USA, October 2006.