

Simplifying the Rate Control Scheme for Distributed Video Coding by Flexible Slepian-Wolf Decoding

Ralph Hänsel and Erika Müller

University of Rostock – Institute of Communications Engineering
Richard-Wagner-Strasse 31, 18119 Rostock, Germany
{ralph.haensel,erika.mueller}@uni-rostock.de
<http://www.int.uni-rostock.de>

Abstract. Distributed video coding (DVC) is an important topic for emerging applications, for example visual sensor networks and mobile video streaming. However, the major handicap applying DVC for real world applications is still the mandatory feedback channel. In previous approaches, the rate allocation algorithm was relying on feedback channel data. Commonly bit plane by bit plane decoding is done in a turbo code-based Slepian-Wolf decoder. In this paper inter bit plane decoding replaces the conventional method. The proposed inter bit plane decoding facilitates the rate control, whereas only the sum data rate has to be estimated/requested but not the data rate for each bit plane. This is one step forward to distributed video coding systems free of feedback channels.

1 Introduction

Distributed video coding (DVC) gained much interest in recent years. There is great potential for important application fields, such as low complexity encoders for mobile video streaming, multi-view video coding in visual sensor networks and error robust video transmission. This paper is focused on the low power scenario, although the presented algorithm is not bound to it.

The key idea of distributed video coding is the disjoint encoding and joint decoding of video sources. For example, disjoint encoding is performed for the frames of a video sequence whereas the decoding is done jointly, by exploiting the temporal correlation. This is the basic principle of a low complexity encoder. The computational expensive motion estimation is shifted from the encoder to the decoder. The principle of separate encoding is applied for example on frame level, on camera level (multi-view coding) or on pixel level.

The “intelligent” component of this encoder/decoder structure is the decoder. Because it holds most of the tools for data compression and control. It is the contrary interpretation as for a conventional video coding system (e.g. Mpeg2, Mpeg4, H.264/AVC). In a conventional encoder/decoder structure the encoder is the “intelligent” component, which controls the decoder.

Typically, the “intelligent” component controls the encoder/decoder system. This leads to a problem for the distributed video coding, because the decoder (as the “intelligent” component) needs to control the encoder. For this reason, a feedback channel from decoder to encoder is used. A bidirectional communication between the encoder and decoder is therefore mandatory.

Up to now, the major handicap for distributed video coding is the mandatory bidirectional communication, to become applicable in many real world scenarios with unidirectional communication. (e.g. video storage, unidirectional video streaming). The efficient storage of a DVC bit stream and subsequent decoding is not possible by now, due to the unavailable feedback channel.

In the next section, a brief description of the related work on distributed video coding is given, as well as a detailed description of the Slepian-Wolf decoding. The new algorithm is presented in section 3, which helps to avoid the feedback channel and makes the distributed video coding scheme more practical. The proposed decoding scheme is called “Flexible Slepian-Wolf Decoding”, because it gives the decoder the opportunity to use the data received from the encoder in a more flexible way. The simulation results are presented in section 4 and the conclusions are given at the end of this paper (section 5).

2 Distributed Video Coding - Related Work

A distributed video coding system is based on the theories of D. Slepian, J. Wolf [1] and A. D. Wyner, J. Ziv [2], describing lossless and lossy distributed source coding.

2.1 Low Complexity Wyner-Ziv CoDec

One major benefit of distributed video coding is the possibility to design low complexity encoders by a disjoint encoding and joint decoding of the frames of a video sequence. Hence, the complex motion estimation is not needed on the encoder side, as in conventional video coding. Therefore, the decoder exploits the temporal correlation by motion estimation.

The basic structure of a common distributed video coding system with low encoder complexity ([3], [4]) is shown in figure 1. The input video sequence is split up into two types of frames. One type of frame called key frame (K) is encoded by a conventional intra encoder (H.264intra, JPEG2000). Every second frame is typically encoded as a key frame. The other frames are encoded as Wyner-Ziv frames (X) by distributed video coding techniques. A Wyner-Ziv frame is encoded separately (without the knowledge of other frames), whereas the decoding is done jointly with knowledge of the neighboring key and Wyner-Ziv frames. The key frames are temporally interpolated to calculate the side information (Y). The side information is a lossy version of the original Wyner-Ziv frame (X). The decoded key frames (\hat{K}) and Wyner-Ziv frames (\hat{X}) are finally combined to form the reconstructed video sequence (fig. 1).

As mentioned, the side information is a lossy (corrupted) version of the original Wyner-Ziv frame. Thus, the side information needs to be corrected to get

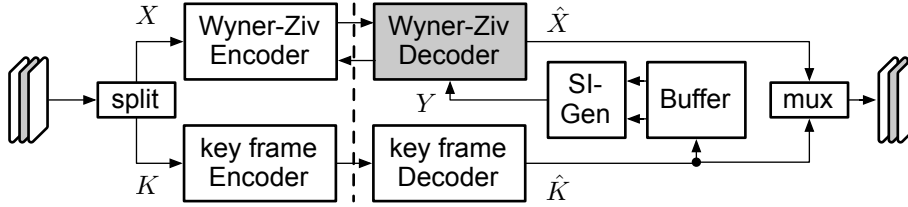


Fig. 1. DVC CoDec scheme with low encoder complexity

the original frame. Which is commonly obtained using channel coding techniques (e.g. [5], [6]). Typically, the Wyner-Ziv encoder generates some parity bits by a channel encoder (convolution code, LDPC - Low Density Parity Check) and transmits them to the decoder. The decoder itself decodes the Wyner-Ziv frames by correcting the side information with the help of the parity bits. The required number of parity bits highly depends on the correlation between the side information and the original Wyner-Ziv frame. For that reason, the decoder must request more and more bits until successful decoding. This method is called decoder rate control. It needs a feedback channel for operation. On the other hand, the encoder can also estimate the necessary data rate and send a fixed number of parity bits (encoder rate control scheme, [7]). Therefore, no feedback channel is needed, but it is not guaranteed that decoding will be successful.

2.2 Slepian-Wolf Encoding and Decoding

A key component of the Wyner-Ziv encoder/decoder is the Slepian-Wolf (SW) encoder/decoder. It is responsible for lossless distributed source coding. The lossy source coding is achieved by introducing a quantizer on the encoder side and a reconstruction on the decoder side. The Wyner-Ziv encoder and decoder as well as the Slepian-Wolf CoDec are shown in figure 2.

A more detailed description of the Wyner-Ziv pixel domain encoder/decoder and the purpose of the SW CoDec is provided in this section. At first, each pixel x_i of the frame X is quantized and consequently transformed to a quantizer

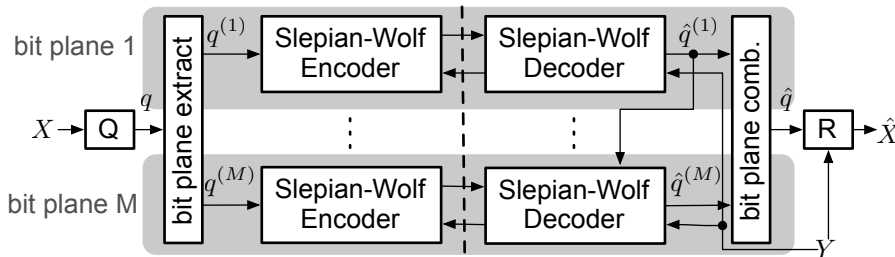


Fig. 2. Wyner-Ziv encoder/decoder

symbol q_i . Each quantizer symbol is split up into its bit planes ($q_i^{(1)}$ - most significant bit (MSB), $q_i^{(M)}$ - least significant bit (LSB)). Whereas each bit plane is fed to a separate Slepian-Wolf (SW) encoder, which generates some parity bits and sends them on request. The bit plane extraction is necessary for coding, due to the fact, that the Slepian-Wolf encoder/decoder is implemented using binary channel codes. For example, turbo codes or LDPC codes are used for Slepian-Wolf coding.

The decoding process is more complex. At first, bit plane 1 is decoded. The side information (Y) and some parity bits are used for decoding. If the decoding is not successful, the decoder requests more bits until it is able to successfully decode the bit plane. In the next step bit plane 2 is decoded in the same way. But the already decoded bit plane is used to improve the decoding performance. Thus, all bit planes are decoded in a sequential way. It is called bit plane by bit plane decoding. Typically, it starts with the MSB and ends with the LSB, but it's not fixed to this order. An inverse bit plane order decoding is proposed in [8].

After successful decoding of all bit planes, the bit planes $\hat{q}_i^{(b)}$ are combined to form the decoded quantization symbol \hat{q}_i . The reconstructed frame \hat{X} is calculated based on the quantizer symbols \hat{q}_i and the side information Y .

The decoder has to request more and more parity bits until successful decoding, which is a problem in this scheme. The requests need to be done for every bit plane in a decoder rate control scheme. On the other hand, the encoder must estimate the data rate for each bit plane b accurately ($R^{(b)}$) rather than only the sum data rate ($R_{sum} = R^{(1)} + R^{(2)} + \dots$) in an encoder rate control scheme. Each bit planes has a fixed data rate as shown in figure 3(a).

The aim of the proposed flexible decoding is a non-fixed data rate for each bit plane, which would greatly facilitate the encoder and decoder rate control.

3 Proposed Inter Bit Plane Decoding

This section points out the proposed inter bit plane decoding algorithm. Furthermore, a model for the quantizer operation, the correlation noise and LLR value calculation are given, which are necessary for flexible inter bit plane decoding.

The aim of the inter bit plane decoding algorithm proposed in this paper is to facilitate the rate control. As mentioned in the related work section, two kinds of rate control schemes are available. A decoder rate control, which needs a feedback channel and an encoder rate control which estimates the necessary data rate at the encoder. In the second case, the rate control has to estimate the data rate for each bit plane accurately, if a bit plane by bit plane decoding scheme is used. Hence, the potential benefit of our inter bit plane decoding is that the encoder rate control has only to estimate the sum data rate. The exact data rate for each bit plane has not to be estimated because the inter bit plane decoding can exchange information between the bit planes. The only constraint is that the data rate for each bit plane is within a specific range, as shown in figure 3(b). The data rate for the first bit plane $R^{(1)}$ can vary between a minimum $R_{min}^{(1)}$

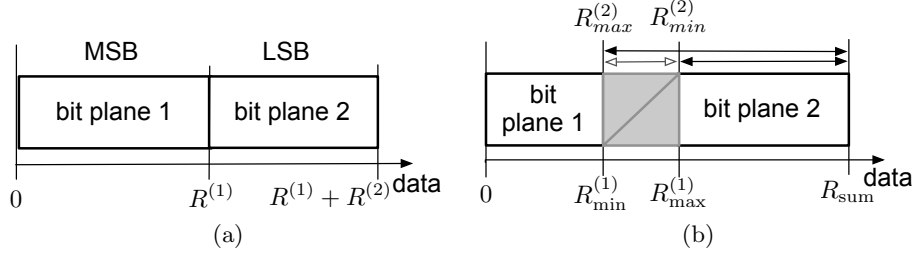


Fig. 3. (a) Fixed and (b) Flexible data rate for each bit plane

and maximum data rate $R_{max}^{(1)}$, whereas the sum data rate R_{sum} must remain constant. The minimum data rate for the MSB $R_{min}^{(1)}$ and the maximum data rate for the MSB $R_{max}^{(1)}$ are given by the bit plane by bit plane decoding. On the one hand, if the MSB is decoded at first than the data rate is at its maximum. On the other hand if the MSB is decoded at last the data rate will be minimum.

The proposed inter bit plane decoding algorithm performs joint decoding of all bit planes. At first, the soft information (LLR value, $L_c(\hat{q}_i^{(b)})$) for every bit plane of the quantization symbol is calculated using the correlation noise model and side information, but independent of the other bit planes. The decoder requests some parity bits from the encoder and performs turbo decoding for every bit plane. The first decoding iteration is typically not successful, because of the not yet performed information exchange. For this reason the soft information is exchanged between the bit planes to reduce the bit rate required for each bit plane. The process of soft information exchange is described below in more detail. Now, turbo decoding and soft information exchange is done again until successful decoding or stopped after 10 iterations. If the decoding is successful the decoder starts over with the reconstruction process. If the decoding was not successful the decoder requests more data from the encoder in a decoder rate control scheme. Inter bit plane decoding is started again with increased amount of data. The decoding fails in case of a encoder rate control scheme, because there is no chance to request more data.

3.1 LLR Calculation and Modeling

The calculation of the soft information (LLR generation, fig. 5) is the key component of our inter bit plane SW decoder. For this reason we present a detailed description of the LLR calculation below.

At the encoder the values of X are scaled to the range $0 \dots 1$ and quantized before Slepian-Wolf coding. This operation is described by the following formula,

$$q_i = \lfloor x_i \cdot 2^M \rfloor = Q(x_i) \tag{1}$$

where x_i is a scaled value of X_i (pixel of input frame), M is the quantization parameter and Q is the quantization operation.

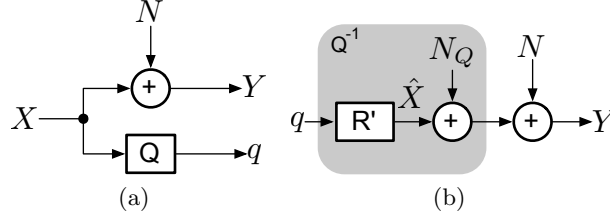


Fig. 4. a - Relation between q and Y , b - Inverse quantizer to compute $p(y_i|q_i)$

The Slepian-Wolf decoder uses the received parity bits to correct the side information Y . The turbo decoder needs the LLR value $L_c(\hat{q}_i^{(b)})$ for every bit plane b of a quantization symbol \hat{q}_i , therefore it is calculated beforehand (eq. 2).

$$L_c(\hat{q}_i^{(b)}) = \ln \frac{p(y_i|q_i^{(b)} = 0, \hat{q}_i^{(k)})}{p(y_i|q_i^{(b)} = 1, \hat{q}_i^{(k)})} \quad \forall k, k \in [1 \dots M], k \neq b \quad (2)$$

The conditional pdf $p(y_i|q_i^{(b)} = 0, \hat{q}_i^{(k)})$ (probability density function) is needed to calculate the LLR value. For this reason a statistical description of the dependency between q_i and y_i is needed. The known relation between the original frame X , the side information Y and the quantizer symbol q is shown in figure 4(a). The side information Y is a noisy version of X and the quantization symbol q the quantised version of X . This setup is transformed into the one shown in figure 4(b) to calculate the dependence between q and Y . The quantizer Q is replaced by its inverse process (grey highlighted).

The quantizer noise N_Q is modeled as equal distribution $p_q(n_{qi})$ (eq. 3) because it is assumed that the original values X_i are equal distributed, too. Furthermore, the correlation noise $p_N(n_i)$ is modeled as Laplacian distributed (eq. 4).

$$p_q(n_{qi}) = \begin{cases} \frac{1}{\Delta} & : |n_{qi}| < \frac{\Delta}{2} \\ 0 & : else \end{cases} \quad (3)$$

$$p_N(n_i) = \frac{1}{2} \lambda e^{-\lambda |n_i|} \quad (4)$$

A mathematical description of the reconstruction process (R' - inverse quantizer model) is given in equation 5 (rescaling is not included). The quantization step size is given by $\Delta = 2^{-M}$. The reconstruction is split up for every bit plane of q in equation 6.

This process can be also described statistically. The corresponding probability density function (pdf) is given in equation 7 and is split up into one pdf for every bit plane b of q_i (see eq. 8), where $\delta(x)$ is the delta function. The pdf $p_R(\hat{x}_i|q_i)$ characterizes the reconstruction of q_i and is calculated by convolution of all pdfs for each bit plane (eq. 9). This is possible due to the fact that the reconstruction of q_i is split up into a sum of separate reconstruction for each bit plane ($q_i^{(b)}$, eq. 6).

Furthermore, equation 10 (adaption of eq. 8) allows inter bit plane decoding. It takes the probabilities for the bit plane b into account which are calculated from

the LLR output values. This allows transfer of information between the decoders of each bit plane. The information transfer is possible before the plane is fully decoded, because the probability for a bit plane $q_i^{(b)}$ is sufficient to calculate the LLR value for the other bit planes.

$$\hat{x}_i = q_i \cdot 2^{-M} + \frac{\Delta}{2} \tag{5}$$

$$\hat{x}_i = q_i^{(1)} \cdot 2^{-1} + \dots + q_i^{(M)} \cdot 2^{-M} + \frac{\Delta}{2} \tag{6}$$

$$p_R(\hat{x}_i|q_i) = 1 \cdot \delta(\hat{x}_i - (q_i 2^{-M} + \frac{\Delta}{2})) \tag{7}$$

$$p_{R,b}(\hat{x}_i|q_i^{(b)}) = 1 \cdot \delta(\hat{x}_i - (q_i^{(b)} 2^{-b})) \tag{8}$$

$$p_R(\hat{x}_i|q_i) = p_{R,1}(\hat{x}_i|q_i^{(1)}) * p_{R,M}(\hat{x}_i|q_i^{(M)}) * \delta(\hat{x}_i - \frac{\Delta}{2}) \tag{9}$$

$$p_{R,b}(\hat{x}_i|q_i^{(b)}) = Pr(q_i^{(b)} = 0) \cdot \delta(\hat{x}_i) + Pr(q_i^{(b)} = 1) \cdot \delta(\hat{x}_i - (q_i 2^{-b})) \tag{10}$$

At least, the dependency between y_i and q_i is given in the following equation. The LLR value is calculated by equation 2 and 12.

$$y_i = R'(q_i) + n_{qi} + n_i \tag{11}$$

$$p(y_i|q_i^{(b)} = 0, \hat{q}_i^{(k)}) = p_R(\hat{x}_i|q_i, q_i^{(b)} = 0) * p_q(n_{qi}) * p_N(n_i) \tag{12}$$

Our proposed LLR value calculation has the advantage that the decoder can do bit plane by bit plane decoding in a non-fixed order as well as inter bit plane decoding. Hence, a bit plane must not be fully decoded to start over with another bit plane. It's possible to switch from one bit plane to another and back (with improved information) before the successful decoding of the bit plane is performed. This one of the main advantages of the proposed inter bit plane decoding.

Our Slepian-Wolf decoder with inter bit plane decoding is shown in figure 5. The LLR values are calculated by equation 2 using the side information Y and the correlation noise parameter λ . The LLR values are passed to a separate decoder (figure 5) for every bit plane $b = 1 \dots M$. Each decoder also obtains the

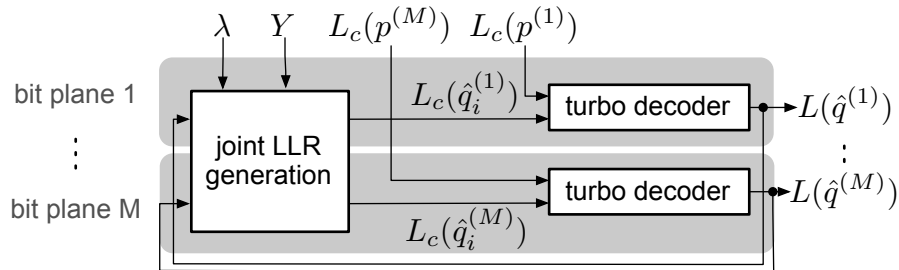


Fig. 5. Slepian-Wolf decoder

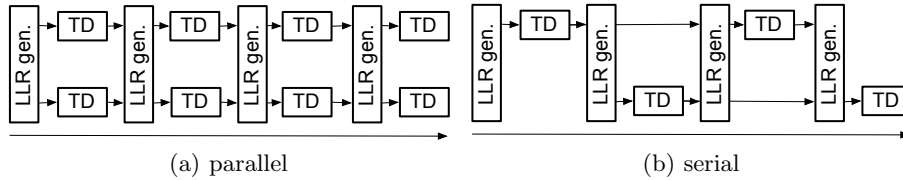


Fig. 6. Joint Slepian-Wolf decoding of the bit planes (TD - turbo decoder)

punctured LLR values $L_c(p^{(b)})$ of the transmitted parity bits. Then the turbo decoder starts decoding and provides the a-posteriori LLR value to the LLR value calculation for the next decoding step. In that way, the decoders for the different bit planes can exchange information.

3.2 Scheduling

The scheduling of the joint decoding for two bit planes is done in a serial or parallel way. In parallel joint decoding (figure 6(a)), the LLR values for each bit plane are calculated and turbo decoding is done before calculating the LLR values again. The serial joint decoding (figure 6(b)) does LLR calculation and afterwards only decodes one bit plane. Before decoding the next bit plane the LLR values are calculated again. The serial decoding is more complex but performs better and is thus used for the proposed joint bit plane decoding.

4 Simulation Results

The key component of our flexible Slepian-Wolf decoding is the inter bit plane decoding. It enables the flexible distribution of the data between the bit planes at the decoder side. The range for the data rate $R^{(b)}$ of a bit plane b is given by $R_{min}^{(b)} \dots R_{max}^{(b)}$. We focus on the 2 bit plane coding here (only MSB and LSB), because the scheduling for more than two bit planes is not solved by now.

The sequences `foreman`, `coastguard`, `hall`, and `mother_daughter` are used for simulation (QCIF, 30fps). A key frame distance of two frames is used (KWK...), where lossless coded key frames are assumed. An uniform quantizer is applied in the Wyner-Ziv CoDec. Furthermore, the Slepian-Wolf encoder is implemented by two parallel concatenated recursive convolutional codes ($G = [1, 33/31]_8$) and a random interleaver. The Slepian-Wolf decoder is implemented by a binary turbo decoder. The data rate is adapted by random puncturing (puncturing period $L_p = 128$) and side information is obtained by temporal interpolation of the neighboring loss less key frames.

4.1 Inter Bit Plane Decoding

The number of parity bits for the MSB and LSB bit planes and the accumulated sum bit rate is shown in figure 7. The data rate for the MSB is linearly increased.

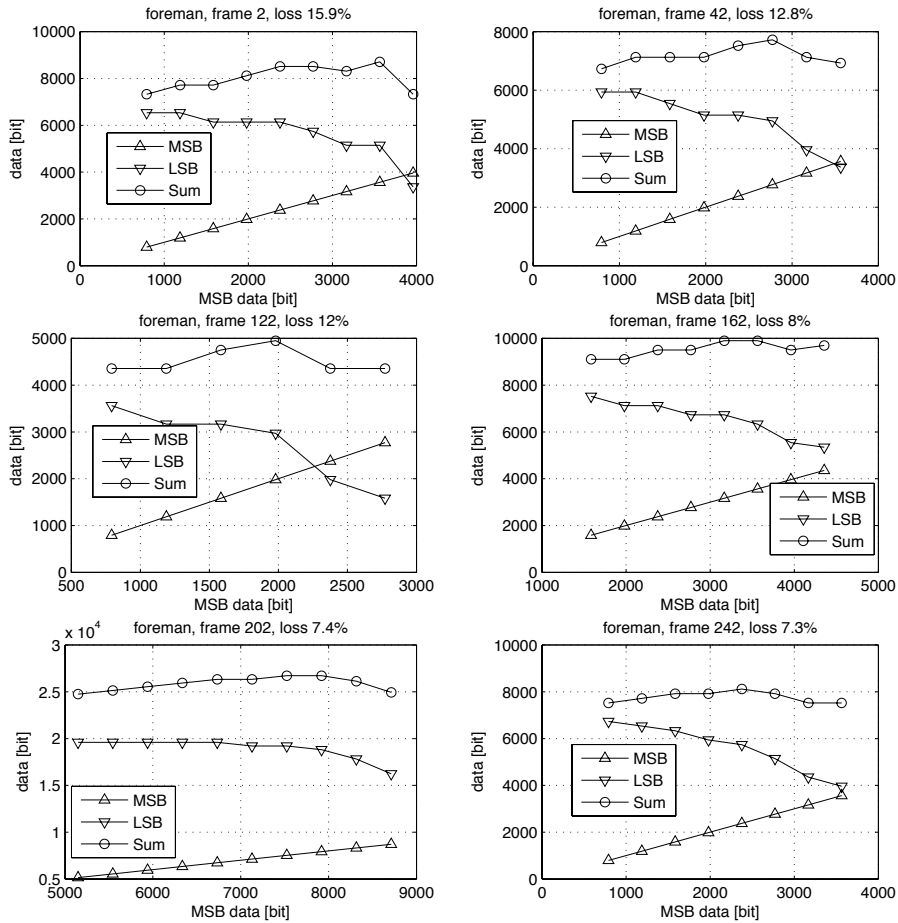


Fig. 7. Inter bit plane decoding: data rate for LSB, MSB, and Sum, including performance loss (foreman, QCIF)

Also the data rate for the LSB is determined for successful decoding. This is done for each data rate of the MSB. As shown in figure 7, the necessary data rate for the LSB is decreased while increasing the data rate for the MSB. The sum data rate is a slight increased between 7% and 16%, which results in a performance loss.

The point for the highest MSB data rate is the same as the bit plane by bit plane (MSB to LSB) decoding in conventional state-of-the-art DVC CoDecs, where the MSB is decoded first. Furthermore, the point for the lowest MSB data rate is equivalent to the bit plane by bit plane decoding in LSB to MSB order, where the LSB is decoded first (fig. 7, frame 42). The data rate point between this borders are only accessible by inter bit plane decoding. For example the data rate distribution at 2000bit MSB data rate (frame 2) cannot be decoded by the conventional bit plane by bit plane decoding, because neither the amount of data for MSB nor for the amount of data for LSB is sufficient to decode this bit plane at first.

The inter bit plane decoding does not influence the reconstruction quality, since the quality remains constant for the same quantization symbol q and identical side information Y . Therefore, the inter bit plane decoding only influences the necessary data rate for decoding (sum data rate).

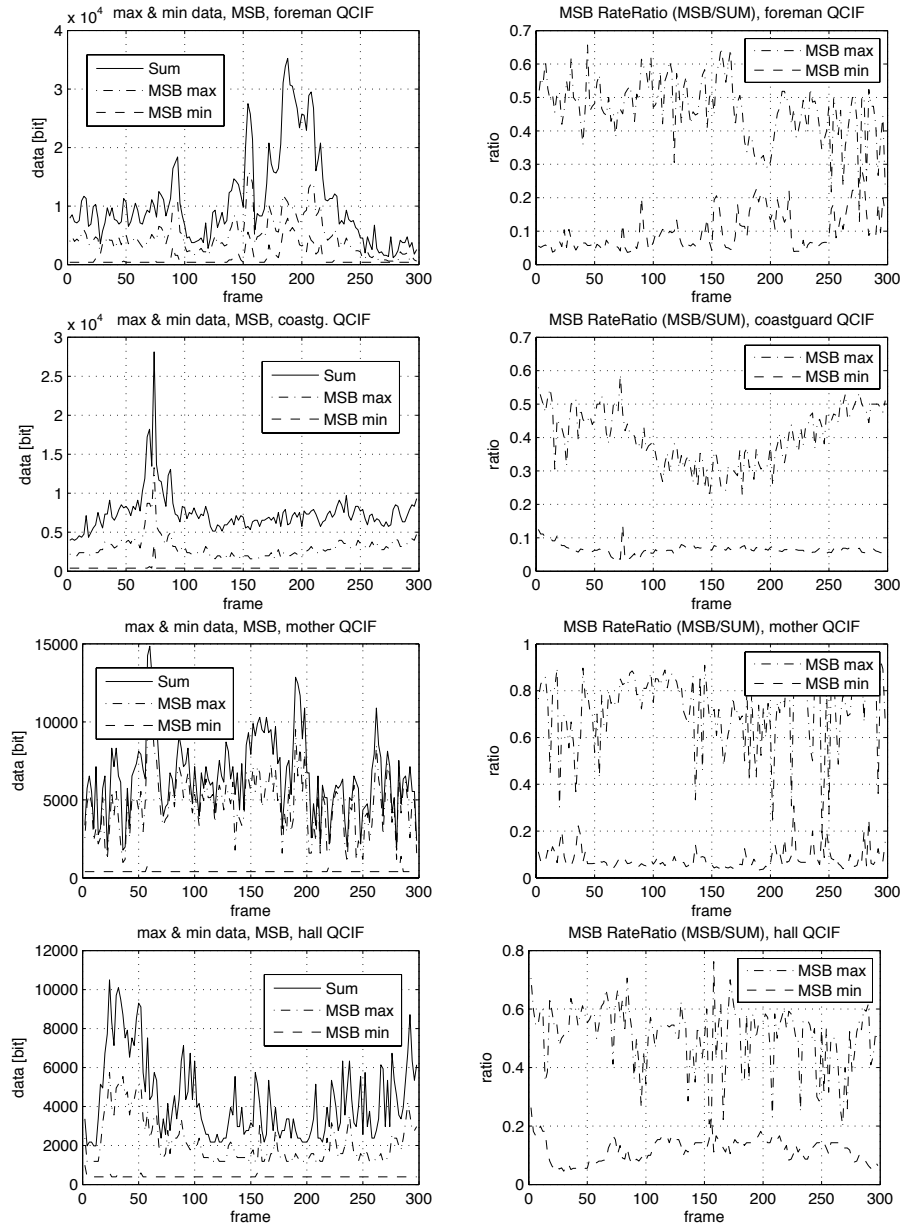


Fig. 8. (left) Bits per bit plane, (right) $ratio_{(min)}^{(1)} \dots ratio_{(max)}^{(1)}$ for sequences foreman, coastguard, mother and daughter, and hall

4.2 Fixed Rate Ratio Coding

The motivation for the "Flexible Slepian-Wolf Decoding" is, that the encoder has not to adjust the data rate for each bit plane, but only the sum data rate. In figure 8 the amount of data needed for successfully decoding of the MSB is shown for both MSB to LSB (max) and LSB to MSB (min) decoding order. This two cases mark the range within the inter bit plane coding can work. The upper and lower borders for the data rate are R_{min} and R_{max} , see figure 3(b). Thus, the needed data for the MSB ($R^{(1)}$) can only be adapted within the range $R_{min}^{(1)} \leq R^{(1)} \leq R_{max}^{(1)}$.

The rate ratio for the MSB ($ratio^{(1)} = R^{(1)}/R_{sum}$) can vary in a given range ($ratio_{min}^{(1)} \dots ratio_{max}^{(1)}$). A rate ratio of $ratio^{(1)} = 0.25$ is in the majority of frames within the range (fig.8). Consequently, for a fixed ratio of 0.25 the transmitted data can be decoded in nearly all cases.

The rate control has thus only to control the sum data rate precisely. On the one hand a encoder rate control only has to estimate the sum data rate, and on the other hand a decoder rate control only has to request more sum rate and not data for a specific plane. This greatly facilitates both rate control schemes.

5 Conclusion

In this paper, we presented a flexible Slepian-Wolf decoding scheme facilitating the rate control scheme for distributed video coding. Thus, it is a step toward a feedback channel-free distributed video coding system, which transmits data over an unidirectional communication channel (e.g. video storage, unidirectional mobile video streaming).

The encoder has not to control the data rate for each bit plane precisely, because nearly all frames can be decoded at a rate ratio of 0.25 (25% of data for MSB and 75% for LSB). In this way, the encoder has only to estimate the sum data rate precisely.

The fixed rate ratio coding is only possible by the use of the proposed inter bit plane decoding (Flexible Slepian-Wolf Decoding). It does data exchange between the bit planes at the decoder side. Furthermore, the inter bit plane decoding will preserve the reconstruction quality of the video, while the data rate is slightly increased.

Further work will include investigation on the case of sub sum rate decoding, when the amount of data is lower than necessary for successful decoding.

Acknowledgment

Special thanks to my colleagues for some inspiring discussions and also for some helpful editorial remarks.

References

1. Slepian, D., Wolf, J.: Noiseless coding of correlated information sources. *IEEE Transactions on Information Theory* 19(4), 471–480 (1973)
2. Wyner, A.D., Ziv, J.: The rate-distortion function for source coding with side information at the decoder. *IEEE Transactions on Information Theory* 22(1), 1–10 (1976)
3. Aaron, A., Zhang, R., Girod, B.: Wyner-Ziv Coding of Motion Video. In: Proc. Asilomar Conference on Signals, Systems, and Computers, Pacific Grove, CA (November 2002)
4. Pereira, F., Ascenso, J., Brites, C.: Studying the GOP Size Impact on the Performance of a Feedback Channel-Based Wyner-Ziv Video Codec. In: Proc. Pacific-Rim Symposium on Image and Video Technology, pp. 801–815 (December 2007)
5. Aaron, A., Girod, B.: Wyner-Ziv Video Coding with Low Encoder Complexity. In: Proc. Picture Coding Symposium (PCS), SF/USA, December 15-17 (2004)
6. Artigas, X., Ascenso, J., Dalai, M., Klomp, S., Kubasov, D., Ouaret, M.: The DISCOVER Codec: Architecture, Techniques and Evaluation. In: Proc. Picture Coding Symposium (PCS), Lisboa (November 2007)
7. Brites, C., Pereira, F.: Encoder Rate Control for Transform Domain Wyner-Ziv Video Coding. In: Proc. IEEE International Conference on Image Processing (ICIP) (September 2007)
8. Vatis, Y., Klomp, S., Ostermann, J.: Inverse Bit Plane Decoding Order for Turbo Code Based Distributed Video Coding. In: Proc. IEEE International Conference on Image Processing (ICIP) (September 2007)