DISTRIBUTED VIDEO CODING WITH SHARED ENCODER/DECODER COMPLEXITY

Tom Clerckx[1]*, Adrian Munteanu, Jan Cornelis and Peter Schelkens*

Vrije Universiteit Brussel – Interdisciplinary Institute for Broadband Technology
Department of Electronics and Informatics (ETRO)
Pleinlaan 2, 1050 Brussels, Belgium,
[1]tclerckx@etro.vub.ac.be

ABSTRACT

Distributed video coding is a coding paradigm that allows complexity to be shared between encoder and decoder. In this context, video coding systems have been developed with encoder complexities similar to H.263+ intra-coding, while obtaining compression performance comparable to H.263+ inter coding. The decoders in these systems typically employ motion-compensated frame interpolation or extrapolation to generate side-information. However, as motion complexity of the video sequence increases, such generators fail to provide reliable side information. This paper proposes a pixel-domain distributed video coding method, combining low-complexity encoder-side bitplane motion estimation with decoder-side motion-compensated frame interpolation. It is shown that such a system is more suitable for sequences with increased motion complexity, compared to codecs that employ motion estimation at the decoder only.

Index Terms— Distributed Video Coding, low-complexity, compression, Slepian-Wolf

1. INTRODUCTION

In the past, video coding architectures have primarily been typecasted as downlink systems. That is, they consist of complex encoding mechanisms running on powerful machines, while the decoder is executed on devices which have only a fraction of the computational power of the encoder. In this design, the encoders are positioned at fixed locations, while the decoders maintain means of mobility.

Today, with the scaling of technologies and the upcoming of high-computational low-power mobile devices, the desire has grown to also code video on these types of microcomputers. This means however, that the downlink-model is no longer applicable and that new systems have to be designed that can cope with several challenges: (i) low-complexity encoding/decoding mechanisms, (ii) robustness against transmission-errors and (iii) high compression performance. Distributed video coding (DVC) is a video coding approach that has the potential to simultaneously meet these requirements. DVC is based on the theoretical paper of Slepian and Wolf [1] on noiseless coding of correlated information sources, which was later on extended for lossy coding by Wyner and Ziv [2]. In typical DVC systems, the encoder uses simple forward error-correcting mechanisms to encode the source in a distributed fashion [3]. It is then the decoder's task to exploit inter-source correlations by means of, for instance, motion-compensated interpolation (MCI).

Previous DVC architectures [3-6] have shown coding performances close to H.263+ inter-coding coupled with an encoder complexity comparable with H.263+ intra-coding. However, there are some downsides on these systems. When a feedback system is used as in [7], it incurs additional decoding delay and makes it unusable in applications where no feedback loop is present, such as in storage devices. Another problem is that usage of motion-compensated frame interpolation only works for sequences where the motion is fairly easy. When motion becomes too complex, the error between the motion-compensated interpolated frame and the real frame is too high to be covered by any forward error correcting code (FEC). This might also be the reason why the use of multiple Wyner-Ziv frames between two intra-coded frames has – to our knowledge – never been used in a DVC system, for complex motion scenes (such as the “Football” sequence).

This paper proposes a DVC architecture where a simple bitplane motion estimation is performed at the encoder. At the decoder side, this allows for an efficient generation of side information, even for complex motion sequences. In combination with a motion-compensated frame interpolation, this allows for a better reconstruction of the Wyner-Ziv frames than when using MCI only.

The rest of this paper is organized as follows: section 2 describes the proposed DVC architecture. The experimental setup followed by a discussion of the results is given in section 3. Finally conclusions are drawn in section 4.

2. DISTRIBUTED VIDEO CODING ARCHITECTURE

The block diagram of the DVC encoder is shown in Figure 1. In a first step, two frames $F(n)$ and $F(n+L)$ are intra coded. The frames $F(n+k)$, with $0<k<L$ are subsequently DVC coded.

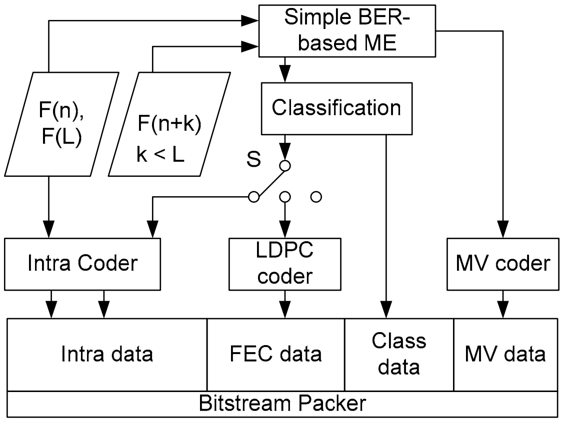

Figure 1. Proposed DVC encoder architecture.

A simple block-based motion estimation (ME) is performed between $F(n+k)$ and $F(n)$. In the ME process, only the p most significant bitplanes (out of a total of M bitplanes) are considered and only a limited set of positions in the search range for each block are tested. In contrast to classical ME, where typically the sum of absolute differences (SAD) is used as a matching criterion, the Bit-Eror Rate (BER) is used as a measure for the distortion between a block and a candidate predictor block.

Next, the total bit-error, as measured in the ME phase, is used by the encoder to classify the block and switch into one of the three following coding modes: (a) no-coding, when the error is very small, (b) entropy-coding, when the error is too high to be covered by any low density parity check (LDPC) code [8], and (c) DVC coding, when the error is sufficiently small to be covered by an LDPC code. In this later case, the block is classified into several sub-classes, corresponding to FECs of various strengths; the appropriate LDPC code rate is then selected according to the measured BER.

Finally, the intra-coded data, the FEC (LDPC) data, the class data and the coded motion vectors (MV) are packed into a bitstream and sent to the decoder.

At the decoder side (Figure 2), the first step to be performed is to intra-decode frames $F^*(n)$ and $F^*(n+L)$. The class data then decides how to decode the frames $F^*(n+k)$, by driving a switch S_1.

When set to intra-decoding (switch S_1 in position (1)), the p bitplanes of $F(n+k)$ are intra decoded, while the leftover bitplanes are retrieved from a side-information generator operating on frames $F^*(n)$ and $F^*(n+L)$.

If the switch S_1 is set to MV decoding (position (2)), an additional switch S_2 is set concurrently, which activates or deactivates LDPC decoding. If S_2 is in switch position (3), an approximation of the p most significant bitplanes of $F(n+k)$ is reconstructed using the MV data and frame $F^*(n)$.

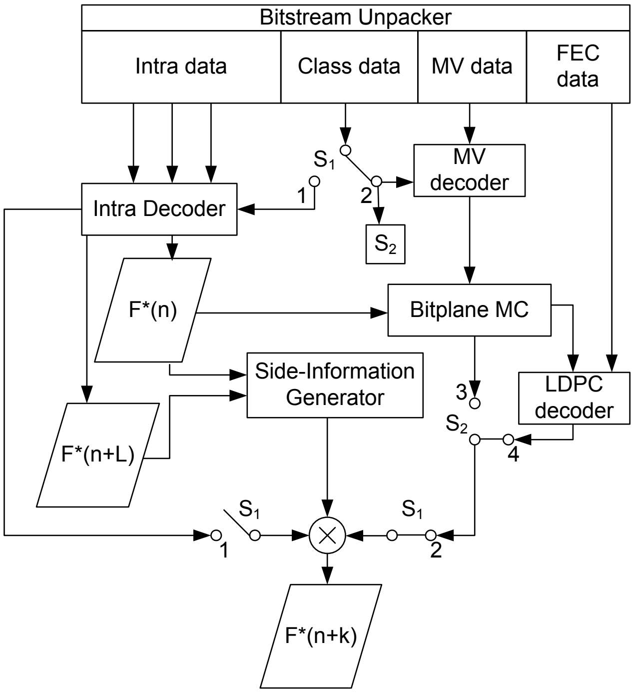

Figure 2. Proposed DVC decoder architecture.

Alternatively, when LDPC decoding is activated (S_2 in position (4)), these bitplanes are subsequently corrected to form the actual p bitplanes of $F(n+k)$, denoted as $F^p(n+k)$. Finally, the leftover $(M-p)$ bitplanes are retrieved from a side information generator operating on frames $F^*(n)$ and $F^*(n+L)$.

3. EXPERIMENTS

In these experiments we show that applying a very simple motion estimation process at the encoder is beneficial when it comes to DVC coding of complex motion scenes employing multiple Wyner-Ziv frames between two I frames. For this, an IP and IPPP structure was chosen for encoding 250 frames of the "Football" CIF sequence at 30 frames/s. Only the luminance component was used in the experiments, with a depth of 8 bit/pixel.

The motion estimation process selects a candidate predictor out of 3, 6 or 9 predictors at fixed locations from the origin of the current block of 16x16 pixels. As a consequence, the raw motion vectors are formed with only 2, 3 or 4 bits. This facilitates the motion vector coding process and will result in only a very small additional cost in motion vector rate. The distance R (see Figure 3) is trained using the first 8 Wyner-Ziv frames, by minimizing the BER for increasing R.

Motion estimation is performed on the first two bitplanes, using the bit-error rate as a *best match* criterion.

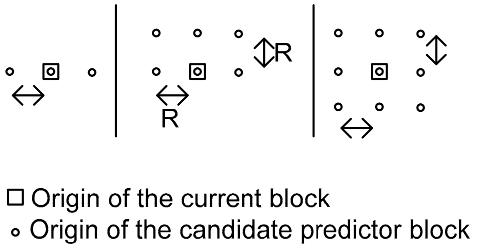

Figure 3. Candidate predictors for 3, 6 or 9 predictors

The motion compensated frame interpolation algorithm is performed using the "Motion Perfect" [9] software, which is an optical-flow interpolation method using the patent based Pelkinetics Engine[10, 11].

Figure 4 shows the percentual bit-error on the first two bitplanes between the original frames and (i) the motion-compensated interpolated frames, (ii) the motion compensated frames, using the motion vectors from the motion-estimation phase with 3 and 9 candidate predictors for an IP structure (i.e. one Wyner-Ziv frame between two intra-coded frames).

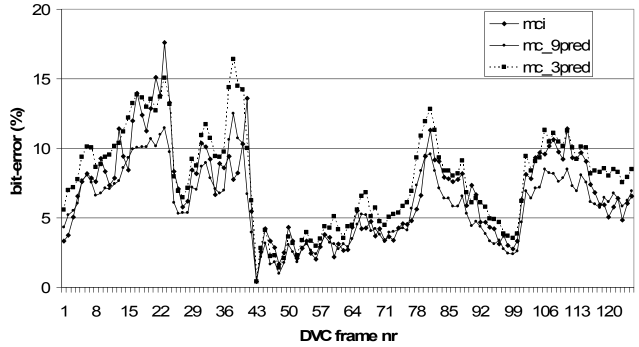

Figure 4. Percentual error between the most significant bitplane of the original frames and (i) motion compensated frames using 3 or 9 predictors (ii) motion-compensated interpolated frame (IP structure)

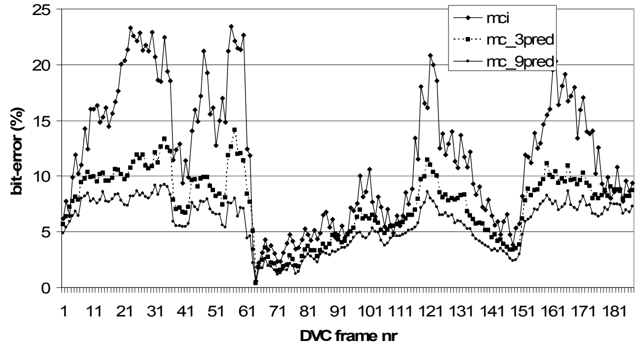

Figure 5. Percentual error between the most significant bitplanes of the original frames and (i) motion compensated frames using 3 or 9 predictors (ii) motion compensated interpolated frame (IPPP structure)

It is observed, that applying a simple encoder side motion estimation using only 3 candidate predictors leads to results which are slightly (1%) worse than motion-compensated frame interpolation. However, when 9 candidate predictors are chosen, already for the IP structure, the quality of the side-information is on average 1% better than when applying MCI.

For an IPPP structure, the benefits of performing simple motion estimation at the encoder site are evident. As shown in Figure 5, applying motion estimation with only 3 candidate predictors leads to significantly better side information, with an average decrease of 4.1% in bit-error. Increasing the number of predictors to 9, decreases the error with almost 6%.

More important than the *reduction* of the bit-error is the fact that – in the IPPP structure – the *absolute* bit-error measured when using motion-compensated frame interpolation surpasses the error threshold of 12% frequently. We note that 12% is approximately the maximum error that can be recovered from using LDPC codes [12]. Having a bit error larger than 12% implies that no LDPC codes should be used for error-correction. That is, a classical DVC scheme (performing no encoder-side ME) will require pure intra-coding in this case, as the intra-coding rate will be lower than the rate spent on error-correction.

Table 1. Summary for the results obtained for the second bitplane for (i) IP structure and (ii) IPPP structure

IP structure	
MCI	Average bit-error: 10%, No. of frames (%) with a bit-error>12%: 34.4
MC, 3 predictors	Average bit-error: 11.6% No. of frames (%) with a bit-error>12%: 54.4
MC, 6 predictors	Average bit-error: 10.1% No. of frames (%) with a bit-error>12%: 35.2
MC, 9 predictors	Average bit-error: 9.4% No. of frames (%) with a bit-error>12%: 22.4
IPPP structure	
MCI	Average bit-error: 15.7% No. of frames (%) with a bit-error>12%: 65.8
MC, 3 predictors	Average bit-error: 11.3% No. of frames (%) with a bit-error>12%: 50.8
MC, 6 predictors	Average bit-error: 9.9% No. of frames (%) with a bit-error>12%: 34.2
MC, 9 predictors	Average bit-error: 9.1% No. of frames (%) with a bit-error>12%: 19.3

The side information generated using the motion vectors with 9 predictors however, never surpasses the 12% upperbound. Additionally, when one uses motion estimation at the encoder, the error is *known*. Thus, an optimal LDPC code can be used to protect the packets against this type of errors, which is not the case when applying motion-compensated frame interpolation only.

The percentual bit-error in the case of motion estimation using 6 predictors, is similar to applying motion compensated frame interpolation i.e. the average error decreases with 0.1% for an IP structure. For an IPPP structure it also outperforms MCI i.e. the average error decreases with 5.3%.

For the second bitplane similar results are observed, with the difference that even for simple ME at the encoder, the bit-error of the generated side information surpasses the maximum recoverable error from time to time. The results for the second bitplane are summarized in Table 1.

Finally, a method where the distance R is adaptively changed every 16 frames, using 2 frames as a training set was investigated. This decreases the bit-error further with about (0.5%, 0.6%) in an IP structure and (0.2%, 0.3%) in an IPPP structure for the most significant and the second bitplane respectively at the cost of additional encoder complexity.

4. CONCLUSIONS

This paper has proposed a new method for distributed video coding, where decoder-side information generation is combined with an encoder side, bitplane based, simple motion estimation process. Results indicate that using a very simple motion-estimation process leads to better side information when multiple (e.g. IPPP) Wyner-Ziv frames are to be coded in between a set of intra coded frames. With an increasing number of Wyner-Ziv frames, or equivalently, with increasing motion complexity, it has been shown how the bit-error can be reduced by interchanging side-information accuracy with encoder-side complexity.

Future work will focus on the integration of the data from the corrected motion-compensated bitplanes and the bitplanes generated by motion-compensated frame interpolation. Furthermore it is expected that the knowledge of the first p decoded bitplanes can aid the decoder-side motion estimation process.

5. ACKNOWLEDGEMENTS

This work was supported in part by IWT (PhD bursary T. Clerckx), and the Fund for Scientific Research – Flanders (post-doctoral fellowships A. Munteanu and P. Schelkens and projects G.0021.03 & G.0053.03).

6. REFERENCES

[1] D. Slepian and J. Wolf, "Noiseless coding of correlated information sources," *IEEE Transactions on Information Theory*, vol. 19, pp. 471-480, 1973.

[2] A. D. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Transactions on Information Theory*, vol. 22, pp. 1-10, 1976.

[3] B. Girod, A. Aaron, S. Rane, and D. Rebello-Monedero, "Distributed Video Coding," *Proc. IEEE*, vol. 93, pp. 71-83, 2005.

[4] A. Aaron and B. Girod, "Wyner-Ziv video coding with low-encoder complexity," presented at *Picture Coding Symposium*, PCS 2004.

[5] R. Puri and K. Ramchandran, "PRISM: a "reversed" multimedia coding paradigm," *IEEE International Conference on Image Processing*, vol. 1, pp. 617-620, 2003.

[6] S. Pradhan and K. Ramchandran, "Distributed Source Coding Using Syndromes (DISCUS)," *IEEE Transactions on Information Theory*, vol. 49, 2003.

[7] A. Aaron, E. Setton, and B. Girod, "Towards practical Wyner-Ziv coding of video," *IEEE International Conference on Image Processing*, vol. III, pp. 869-872, 2003.

[8] R. Gallager, "Low-Density Parity-Check Codes," Massachusetts Institute of Technology, 1963.

[9] I. Dynapel Systems, "MotionPerfect," http://www.dynapel.com/

[10] M. Griessl, M. Wittkop, and S. Wonneberger, " Method and system for the estimation and compensation of brightness changes for optical flow calculations," Patent nr. 6959118, ed: Dynapel Systems, Inc.

[11] E. Steven and D. Klaus, " Motion picture enhancing system," Patent nr. 6351545, ed: DynaPel Systems, Inc.

[12] T. Richardson and R. Urbanke, "The Capacity of Low-Density Parity-Check Codes Under Message-Passing Decoding," *IEEE Trans. Inform. Theory*, vol. 47, pp. 599-618, 2001.