IMPROVING FRAME INTERPOLATION WITH SPATIAL MOTION SMOOTHING FOR PIXEL DOMAIN DISTRIBUTED VIDEO CODING[1]

João Ascenso[1], Catarina Brites[2], Fernando Pereira[3]

[1]joao.ascenso@lx.it.pt, [2]catarina.brites@lx.it.pt, [3]fp@lx.it.pt
[1]Instituto Superior de Engenharia de Lisboa – Instituto das Telecomunicações
[2,3]Instituto Superior Técnico – Instituto das Telecomunicações

Abstract: *Distributed video coding (DVC) is a new compression paradigm based on two key Information Theory results: the Slepian-Wolf and Wyner-Ziv theorems. A particular case of DVC deals with lossy source coding with side information at the decoder (Wyner-Ziv) and enables to shift the coding complexity from the encoder to the decoder. The solution here described is based on a very lightweight encoder leaving for the decoder the time consuming motion estimation/compensation task. In this paper, the performance of the pixel domain distributed video codec is improved by using better side information based derived by motion compensated frame interpolation algorithms at the decoder. Besides forward and bidirectional motion estimation, a spatial motion smoothing algorithm to eliminate motion outliers is proposed. This allows significant improvements in the rate-distortion (RD) performance without sacrificing the encoder complexity.*

Key words: *distributed video coding, Wyner-Ziv, Slepian-Wolf, frame interpolation, side information*.

1 INTRODUCTION

Nowadays, the most popular digital video coding solutions are represented by the ITU-T and ISO/IEC MPEG standards, and rely on the powerful hybrid block-based transform and interframe predictive coding paradigm. In this coding framework, the encoder architecture is based on the combination of motion estimation tools with DCT transform, quantization and entropy coding in order to exploit the temporal, spatial and statistical redundancy in a video sequence. In this framework, the encoder has a higher computational complexity than the decoder, typically 5 to 10 times more complex [1], specially related to the very computationally consuming motion estimation operations. This type of architecture is well-suited for applications where the video is encoded once and decoded many times, i.e. one-to-many topologies, such as broadcasting or video-on-demand, this means where the cost of the decoder is more critical than the cost of the encoder.

In recent years, with emerging applications such as wireless low-power surveillance, multimedia sensor networks, wireless PC cameras and mobile camera phones, the traditional video coding architecture is being challenged. These applications have different requirements than those related to traditional video delivery systems. For some applications, it is essential to have a low-power consumption both at the encoder and decoder, e.g. in mobile camera phones. In other types of applications, notably when there is a high number of encoders and only one decoder, e.g. surveillance, low cost encoder devices are needed. To fulfill these new requirements, it is essential to have a coding configuration with a low-power and low-complexity encoder device, possibly at the expense of a high-complexity decoder. In this configuration, the goal in terms of compression efficiency would be to achieve a coding efficiency similar to the best available hybrid video coding schemes (e.g. the recent H.264/AVC standard [2]); that is, the shift of complexity from the encoder to the decoder should ideally not compromise the coding efficiency. Although this is currently rather far from happening and much research needs to happen in this area, the most promising solution around is the so called distributed video coding (DVC) paradigm explained in the following.

From the Information Theory, the Slepian-Wolf theorem [3] states that it is possible to compress two statistically dependent signals, X and Y, in a distributed way (separate encoding, jointly decoding) using a

[1] The work presented was developed within VISNET, a European Network of Excellence (http://www.visnet-noe.org).

rate similar to that used in a system where the signals are encoded and decoded together, i.e. like in traditional video coding schemes. The complement of Slepian-Wolf coding for lossy compression is the Wyner-Ziv work on source coding with side information at the decoder [4]. This corresponds to a particular case of Slepian-Wolf coding, which deals with source coding of the X sequence considering that the Y sequence, known as side information, is only available at the decoder. Wyner and Ziv showed that there is no increase in the transmission rate if the statistical dependency between X and Y is only explored at the decoder compared to the case where it is explored both at the decoder and the encoder (with X and Y jointly Gaussian and a mean-square error distortion measure).

Today, one of the most studied distributed video codecs uses a turbo-based pixel domain Wyner-Ziv coding scheme [5], because of its simple and low complexity encoder architecture. The decoder is responsible to explore the source statistics, and therefore to achieve compression for the Wyner-Ziv solution, which represents a major departure from current video coding architectures. In the proposed solution, the video frames are organized into key frames and Wyner-Ziv frames; the key frames are encoded with a conventional intraframe codec and the frames between them are Wyner-Ziv encoded. At the decoder, the side information is generated using previously decoded key frames and motion interpolation tools, responsible to obtain an accurate interpolation of the frame to be Wyner-Ziv decoded. The more accurate the side information is the fewer are the Wyner-Ziv bits required to provide a reliable decoding of the Wyner-Ziv frame. Thus, the rate-distortion (RD) performance of such a Wyner-Ziv video coding scheme is highly dependent on the quality of the side information and the challenge is: how to generate the best side information (a frame) as close as possible to the current Wyner-Ziv frame to be decoded? In this context, the major contribution of this paper is to propose a novel motion compensated frame interpolation scheme based on spatial motion smoothing and evaluate the RD performance when this scheme is used in a turbo-based pixel domain Wyner-Ziv video codec in comparison with simpler motion estimation solutions.

This paper is organized as follows. First, in Section 2, the pixel domain Wyner-Ziv video codec architecture is presented, highlighting the important Slepian-Wolf rate compatible punctured turbo (RCPT) coder which is responsible to correct the mismatch (errors) between the side information and the frame to be decoded. In Section 3, the new frame interpolation scheme based on spatial motion smoothing is described in detail and in Section 4 several experiments are developed to evaluate the impact of the various motion estimation solutions in the performance of the codec. The conclusions and some future work topics are presented in Section 5.

2 PIXEL DOMAIN WYNER-ZIV (PDWZ) VIDEO CODEC ARCHITECTURE

The PDWZ solution here presented is based on the pixel domain Wyner-Ziv coding architecture proposed in [5]. The main advantage of this approach is the low computational complexity offered, since it uses only a uniform quantizer and a turbo encoder for the Wyner-Ziv frames; the key frames are perfectly reconstructed at the decoder. This scheme can provide interesting coding solutions for some applications where low encoding complexity is a major goal, e.g. video-based sensor networks.

Fig. 1 illustrates the global architecture of the PDWZ codec. This general architecture makes use of a quantizer, a turbo-code based Slepian-Wolf codec, a frame interpolation module and a reconstruction module. However, there is a major difference between the PDWZ solution proposed here and the solution in [5] regarding the frame interpolation tools used to generate the side information, which is the best estimate made at the decoder of the frame being decoded. In a nutshell, the coding procedure illustrated in Fig. 1 is described as follows: each even frame of a video sequence X_{2i}, called Wyner-Ziv frame is encoded pixel by pixel. Over the resultant quantized symbol stream q_{2i} (constituted by all the quantized symbols of X_{2i} using M levels) bitplane extraction is performed and each bitplane is then independently turbo-encoded. In the PDWZ solution, the turbo coding is performed at the bitplane level [6]. The decoder frame interpolation module generates the side information, Y_{2i}, which is then used by the turbo decoder to obtain the decoded quantized symbol stream q'_{2i}. The side information is also necessary in the reconstruction module, together with the q'_{2i} stream, to help in the X_{2i} reconstruction task. As shown in the architecture, the Slepian-Wolf encoder includes a turbo encoder and a buffer and it produces a sequence of parity bits (redundant bits) associated to each pixel bitplane; the amount of parity bits produced for each bitplane depends on the turbo encoder rate. Only the luminance or also chrominance may be encoded in a similar way. In this architecture, two identical recursive encoders of rate ½ are used with a generator matrix as defined in [6]. The parity bits generated by the turbo encoder are then stored in the buffer, punctured and

transmitted upon request by the decoder; the systematic bits are discarded. The puncturing operation allows sending only a fraction of the parity bits and follows a specific puncturing pattern.

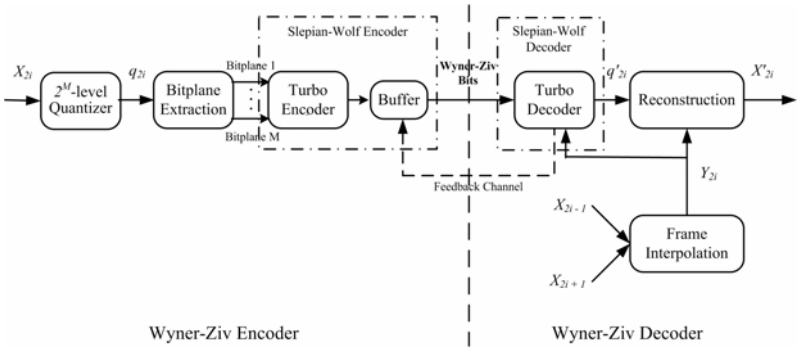

Figure 1 – PDWZ video codec architecture.

The feedback channel is necessary to adapt to the changing statistics between the side information and the frame to be encoded, i.e. to the quality (or accuracy) of the frame interpolation. In this way, it is guaranteed that only a minimum of parity bits are sent in order to correct the mismatches/errors which are present in each bitplane, and thus a minimum rate is achieved. An ideal error detection capability is assumed at the decoder, i.e. the decoder is able to measure in a perfect way the current bit-plane error rate, Pe. If Pe $\geq 10^{-3}$, the decoder requests for more parity bits from the encoder. In the decoder, the iterative MAP (*Maximum A Posteriori*) turbo decoder employs a Laplacian noise model to aid in the error correction capability of the turbo codes. This model provides a good fit to the residual distribution between the side information and the frame to be encoded. The distribution parameter of the Laplacian distribution was found by constructing the residual histogram of several sequences using the proposed techniques to generate the side information, i.e. the frame interpolation tools.

3 FRAME INTERPOLATION SOLUTIONS

There are several frame interpolation techniques that can be employed at the Wyner-Ziv decoder to generate the side information, Y_{2i}. The choice of the technique used can significantly influence the PDWZ codec rate distortion performance; more accurate side information through frame interpolation means fewer errors (Y_{2i} is more similar to X_{2i}) and therefore the decoder needs to request less parity bits from the encoder and the bitrate is reduced for the same quality. The side information is normally interpreted as an "oracle", this means an attempt by the decoder to predict the current Wyner-Ziv frame based on temporally adjacent frames (key frames).

The simplest frame interpolation techniques that can be used are to make Y_{2i} equal to X_{2i}-1, i.e. the previous temporally adjacent frame, or to perform bilinear (average) interpolation between the key frames X_{2i}-1 and X_{2i}+1. However, if these techniques are used to generate the side information in medium or high motion video sequences, Y_{2i} will be a rough estimate of X_{2i} since the similarity between two temporally adjacent frames will be rather low. In this case, the decoder will need to request more parity bits from the encoder when compared to the case where Y_{2i} is a closer estimate to X_{2i} and thus the bitrate will increase for the same PSNR. Subjectively, these simple schemes will introduce "jerkiness" and "ghosting" artifacts in the decoded image X'_{2i}, especially for low bitrates. These observations motivate the need to use more powerful motion estimation techniques since the accuracy of the decoder frame interpolation module is a key factor for the final compression performance. However, the traditional motion estimation and compensation techniques used at the encoder for hybrid video coding are not adequate to perform frame interpolation since they attempt to choose the best prediction for the current frame in the rate-distortion sense. For frame interpolation, we need to find an estimate of the current frame, and therefore a good criteria is to estimate the true motion, and based on that to perform motion compensation between temporally adjacent frames. In this paper, block-based motion compensated interpolation is proposed due to its low complexity and the need to maintain some compatibility with the current video compression standards. Fig. 2 shows the architecture proposed for the frame interpolation scheme. Besides the low pass filter and the motion compensation modules which are always used, the three modules in the middle are associated to increasingly more powerful motion estimation solutions when 1, 2 or 3 modules are used (always starting from the first module on the left, this means the forward motion estimation module). In the following all modules are described in more detail.

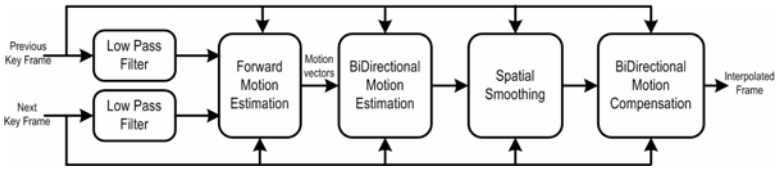

Figure 2 – Proposed frame interpolation framework.

3.1 Forward Motion Estimation

First of all, both key frames are low pass filtered to improve the reliability of the motion vectors; this will help to estimate motion vectors closer to the true motion field. Then a block matching algorithm is used to estimate the motion between the next and previous key frame. The parameters that characterize this motion estimation technique are the search window size, the search range and the step size. The step size is the distance between pixels in the previous key frame a motion vector is searched for, and enables to reduce the computational complexity of the scheme and to provide only a coarse approximation of the true motion field.

However, this rigid block based motion estimation scheme fails to capture all aspects of the motion field, and if frame interpolation is performed, overlapped and uncovered areas will appear. This is because the motion vectors obtained do not necessarily intercept the interpolated frame at the center of each non-overlapped block in the interpolated frame. The motion vectors obtained in the previous step serve as candidates for each non-overlapped block in the interpolation frame in such a way that for each block of the interpolation frame is selected, from the available candidate vectors, the motion vector that intercepts the interpolated frame closer to the center of block under consideration (see Fig. 3a). Now that each block in the interpolated image has a motion vector, bidirectional motion compensation (see Section 3.4) can be performed to obtain the interpolated frame or further processing is done in the next modules.

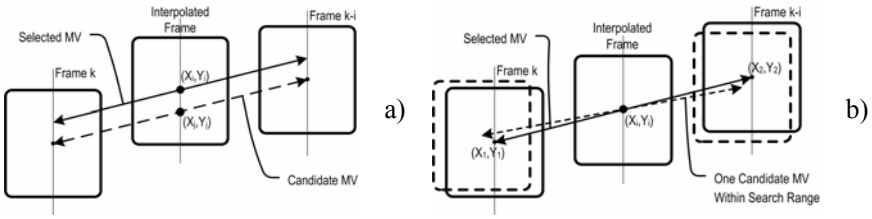

Figure 3 – a) selection of the motion vector; and b) bidirectional motion estimation.

3.2 Bidirectional Motion Estimation

The bidirectional motion estimation module refines the motion vectors obtained in the previous step by using a bidirectional motion estimation scheme similar to the B-frames coding mode used in current video standards [2]. However, since here the interpolated pixels are not known, a different motion estimation technique is used. This technique selects a linear trajectory between the next and previous key frames passing at the center of the blocks in the interpolated frame (Fig. 3b). The search range is confined to a small displacement around the initial block position and the motion vectors between the interpolated frame and previous and next key frames are symmetric, i.e. $(x_1, y_1) = (x_i, y_i) + MV(B_i)$ and $(x_2, y_2) = (x_i, y_i) - MV(B_i)$, where (x_1, y_1) is the coordinates of the block in the previous key frame, (x_2, y_2) are the coordinates of the block in the next frame and $MV(B_i)$ represents the motion vector obtained in the previous section divided by half, since the interpolated frame is equally distant to both key frames.

3.3 Spatial Motion Smoothing Based Estimation

Once the bidirectional motion field is obtained, it is observed that the motion vectors have sometimes low spatial coherence; this can be improved by spatial smoothing algorithms targeting the reduction of the number of false motion vectors, i.e. incorrect motion vectors when compared to the true motion field. The proposed scheme uses weighted vector median filters [7], extensively used for noise removal in multichannel images, since all the components (or channels) of the noisy image are to be taken into consideration. The weighted median vector filter maintains the motion field spatial coherence by looking, at each block, for candidate motion vectors at neighboring blocks. This filter is also adjustable by a set of weights controlling the filter smoothing strength (or spatial homogeneity of the resulting motion field) depending on the prediction MSE (*Mean Square Error*) error of the block for each candidate motion vector

(calculated between key frames). The proposed spatial motion smoothing algorithm is both effective at the image boundary, where abrupt changes of the direction of the motion vectors occur, as well as in homogenous regions (with similar motion) where the outliers are effectively removed. In Figure 4, a comparison between an interpolated image with and without spatial motion smoothing is presented.

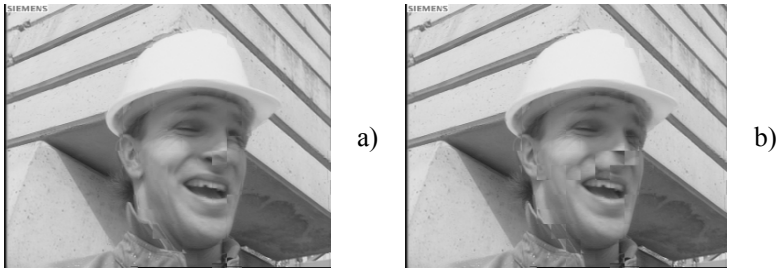

Fig. 4 – Foreman sequence, frame #7; a) with and b) without spatial smoothing.

The weighted median vector filter proposed is defined as in [7]:

$$\sum_{j=1}^{N} w_j \left\| x_{wvmf} - x_j \right\|_L \leq \sum_{j=1}^{N} w_j \left\| x_i - x_j \right\|_L , \quad (1)$$

where $x_1, \ldots, x_N$ are the motion vectors of the current block in the previously interpolated frame and the corresponding nearest neighboring blocks; $w_1, \ldots, w_N$ correspond to a set of adaptively-varying weights and x_{wvmf} represents the motion vector output of the weighted vector median filter. The vector x_{wvmf} is chosen in order to minimize the sum of distances to the other N-1 vectors, in terms of the L-norm. The choice of weights is performed according to the prediction error as defined by:

$$w_j = \frac{MSE(x_c, B)}{MSE(x_j, B)}, \quad (2)$$

where x_c represents the candidate vector for the current block B to be smoothed. The MSE (*Mean Square Error*) represents the matching success between the current block B in the next key frame and the block in the previous key frame motion compensated with vectors x_c and x_j. The weights have low values when the MSE for the candidate vector is high, i.e. when there is a high prediction error and the weights have high values when the prediction error for the candidate vector is low. Therefore, the decision to substitute the previously estimated motion vector with a neighboring vector is made by evaluating both the prediction error and the spatial properties of the motion field.

3.4 Bidirectional Motion Compensation

Once the final motion vector field is obtained, the interpolated frame can be filled by simply using bidirectional motion compensation as defined in standard video coding schemes. The assumption is that the time interval between the previous key frame and the interpolated frame is similar to the time interval between the interpolated frame and the next key frame interpolated frames, so each reference image has the same weight (½) when is performed motion compensation.

4 EXPERIMENTAL RESULTS

In order to evaluate the rate-distortion performance of the proposed PDWZ codec, four frame interpolation techniques will be considered in the following to generate the side information: i) average frame interpolation; ii) only forward motion estimation (FME), iii) forward motion estimation followed by bidirectional motion estimation (BiME) and finally iv) forward motion estimation followed by bidirectional motion estimation and spatial motion smoothing (SS). Bidirectional motion compensation is always performed to fill the interpolated frame.

These experiments will show the contribution of each functional block proposed for frame interpolation framework in the overall PDWZ performance. In all the experiments, the block size is 8×8; the search range is ±8 pixels and the step size is 2 for the forward motion estimation; for the refinement process the search range is adjusted by ±2 pixels. These parameters were obtained after performing extensive experiments and are those that better fit for QCIF resolution sequences. The PSNR versus bitrate results for all the frames of the Foreman and Coastguard QCIF sequences are shown in Fig. 5. In both figures, only the luminance rate and distortion of the Wyner-Ziv frames are included; the Wyner-Ziv frame rate is 15 fps. It is assumed that the odd frames (key frames) are available at the decoder perfectly reconstructed. The

results are compared against the H.263+ intraframe coding and the H.263+ interframe coding with a IBIB structure. In the last case, only the rate and PSNR of the B frames is shown.

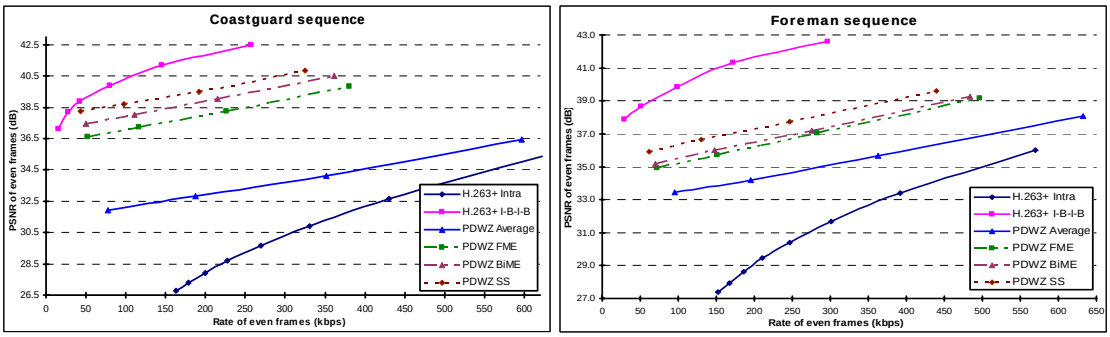

Fig. 5 – PDWZ RD performance for the Coastguard and Foreman QCIF sequences.

The results obtained show that the proposed motion estimation techniques improve significantly the PDWZ RD performance, especially when compared to the average frame interpolation. RD improvements are observed for both sequences when the frame interpolation solution is successively made more powerful by adding additional tools, which validates the approach (and architecture) of the proposed frame interpolation scheme. Bidirectional ME provides better gains for the Coastguard sequence (up to 0.8 dB) than Foreman (up to 0.3 dB) when compared to the forward ME scheme and spatial smoothing has similar gains for both sequences when compared to the BiME scheme (up to 0.8 dB). From the results, it is also possible to observe remarkable gains over H.263+ intraframe coding for all bitrates and sequences. However, there is still a gap when compared to H.263+ interframe coding (IBIB); as expected, this gap is smaller for sequences with well-defined camera motion like Coastguard, since the interpolation tools can provide better performance for this type of sequences. Additionally, when the results of the first 100 frames are compared (the same number of coded frames in [6]) with the most recent pixel domain Wyner-Ziv coding results available in the literature [6], the PDWZ SS solution shows an improvement of up to 2 dB in coding efficiency, for the conditions stated above.

5 CONCLUSIONS AND FUTURE WORK

In this paper, new motion compensated frame interpolation tools are proposed and compared in the context of a pixel domain Wyner-Ziv (PDWZ) video decoder. The proposed frame interpolation framework is composed of four major tools: forward motion estimation, bidirectional motion estimation, spatial smoothing, and bidirectional motion compensation. Experimental results prove that this framework improves the PDWZ coding efficiency compared to other similar solutions, without sacrificing the encoder complexity. This way, it is possible to approximate the PDWZ performance to the interframe H.263+ performance, thus reducing the gap in quality between the two. As future work, it is planned to further enhance the RD performance of the codec with algorithms that take into account the strong spatial correlation among neighboring pixels or by using some iterative motion refinement approach using an intermediate decoded frame.

6 REFERENCES

[1] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockammer and T. Wedi, “Video Coding with H.264/AVC: Tools, Performance, and Complexity”, IEEE Circuits and Systems, Vol. 4, No. 1, 2004.

[2] ISO/IEC International Standard 14496-10:2003, “Information Technology – Coding of Audio-visual Objects – Part 10: Advanced Video Coding”.

[3] J. Slepian and J. Wolf, “Noiseless Coding of Correlated Information Sources”, IEEE Trans. on Information Theory, Vol. 19, No. 4, July 1973.

[4] A. Wyner and J. Ziv, “The Rate-Distortion Function for Source Coding with Side Information at the Decoder”, IEEE Trans. on Information Theory, Vol. 22, No. 1, January 1976.

[5] A. Aaron, R. Zhang and B. Girod, “Wyner-Ziv Coding for Motion Video”, Asilomar Conference on Signals, Systems and Computers, Pacific Grove, USA, November 2002.

[6] A. Aaron, S. Rane, E. Setton and B. Girod, “Transform-Domain Wyner-Ziv Codec for Video”, VCIP, San Jose, USA, January 2004.

[7] L. Alparone, M. Barni, F. Bartolini, V. Cappellini, “Adaptively Weighted Vector-Median Filters for Motion Fields Smoothing”, IEEE ICASSP, Georgia, USA, May 1996.