*Journal of Information Hiding and Multimedia Signal Processing* ©2012 ISSN 2073-4212
**Ubiquitous International** Volume **3**, Number **1**, January **2012**

# Progressive Coding and Side Information Updating for Distributed Video Coding

Mohamed Haj Taieb
Laval University, Quebec, QC, Canada, G1K 7P4
mohamed.haj-taieb.1@ulaval.ca

Jean-Yves Chouinard and Demin Wang
Laval University, Quebec, QC, Canada, G1K 7P4
Communications Research Centre Canada, Ottawa, ON, K2H 8S2 Canada
jean-yves.chouinard@gel.ulaval.ca, Demin.Wang@crc.ca

Khaled Loukhaoukha and Grègory Huchet
Laval University, Quebec, QC, Canada, G1K 7P4
Communications Research Centre Canada, Ottawa, ON, K2H 8S2 Canada
khaled.loukhaoukha.1@ulaval.ca, gregory.huchet@crc.ca

Received October 2010; revised July 2011

Abstract. *Distributed video coding is a research field which brings together error coding techniques along with video compression methods. It is usually based on a Slepian-Wolf encoder which often involves turbo codes because of their strong error correction capabilities. Typically, the turbo encoder generates parity bits which are sent to re- fine the side information reconstructed at the decoder by interpolation of the already received neighboring key frames. In this paper we introduce a novel distributed video coding scheme with progressive decoding. The side information is updated progressively as long as the current frame is being decoded. The proposed architecture considers a chess-board structure for block grouping. A subset of blocks are first sent, decoded and then used to update the side information. Then, the remaining blocks are sent and de- coded using the updated and more accurate side information. The implementation of the progressive coding shows an improvement up to 1.7 dB over the conventional DVC architecture.*
**Keywords:** Distributed video coding, bidirectional motion estimation and compensation, progressive coding, Wyner-Ziv.

1. **Introduction.** Distributed video coding (DVC) is a relatively recent research field which began attracting several researchers since the introduction of Stanford University DVC architecture in 2002 [1]. The improvements achieved in the subsequent studies, are mainly based on the same architecture. The frames of a video sequence are divided into intraframes (key frames) and interframes (Wyner-Ziv frames). The intraframes are sent and used by the receiver to generate side information to decode the interframes. This side information is actually the result of an interpolation, or an extrapolation, of the received key frames to estimate the WZ frames. This estimation is done without any knowledge about the current frame and makes several assumptions during the motion compensation.

In this paper we present a modification to this commonly used architecture to allow for progressive video coding and decoding. A WZ frame is divided into blocks and the blocks are grouped into two sets of blocks. One of them is sent first to the receiver and decoded using the side information. After decoding the first set, the side information is updated to provide a more accurate one. The updated side information is then used to decode the second set of blocks with lower bitrates since it is affected with lower distortion. A similar approach of side information update in the wavelet domain is proposed in [2]. The decoded lower frequency coefficients are used to enhance the quality of the side information used for the next higher frequency coefficients. In [3] as well, the idea of side information refinement is used. The DWT coefficients are progressively updated after the decoding of each bit planes. This leads to better side information for the next bit planes.

2. **Conventional Distributed video coding scheme.** The architecture of a conventional transform domain distributed video coding system is depicted in figure 1. The odd-numbered frames (intraframes) are considered perfectly reconstructed at the decoder and will be used to generate the side information to decode the even frames (interframes). A block based discrete cosine transform (DCT 4x4) is applied to each interframe. The DCT coefficients are fed to a uniform quantizer and then to a turbo encoder, consisting of the two constituent rate 1/2 recursive convolution encoders. Each of the two RSCs (recursive systematic coders) associates a parity bit to a quantized pixel. The parity bits are stored in a buffer and sent gradually to the encoder upon request. To ensure compression, the systematic bits are discarded since the decoder has already an interpolated version of the even frames which can replace these bits in the turbo decoding process. At the decoder, an interpolated version of the current WZ frame is produced using the neighboring key frames already received. The interpolation technique presented by Ascendo and Pereira [4] was adopted for most of the DVC architecture. Details about this technique are presented in the next section. The interpolated frame is then DCT transformed: the DCT coefficient represents the side information to be used to decode the WZ Frame.

We consider a virtual channel which consists of the WZ DCT coefficients as the virtual channel input and the side information as the output. A Laplacian model is assumed for this virtual channel for the turbo decoding. The estimation of Laplacian distribution parameter $\alpha$ is based on the online correlation noise modeling technique at the coefficient / frame level. Parameter $\alpha$ is estimated for each coefficient band of each frame. More details on the noise modeling technique used in this paper are provided in [5].

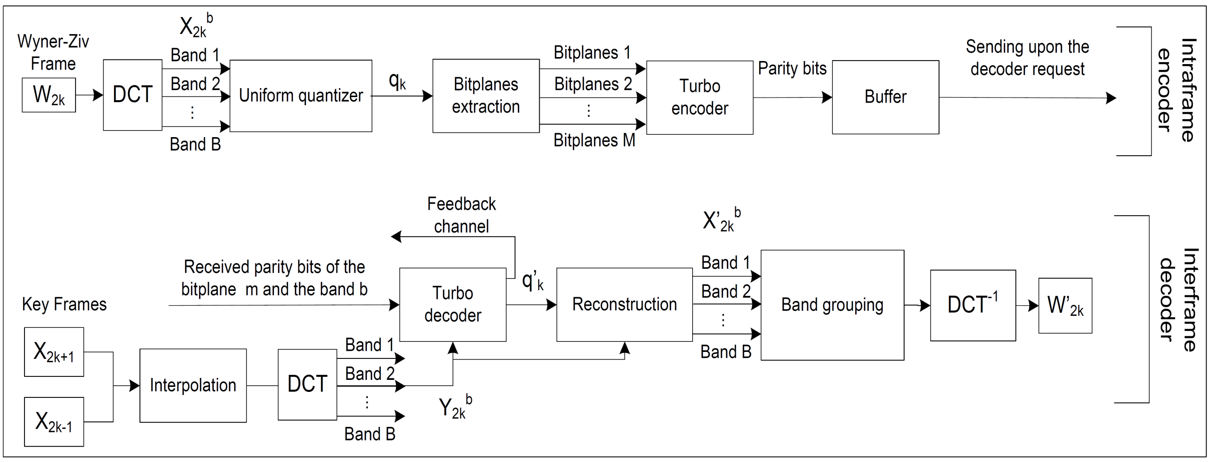


FIGURE 1. Conventional transform domain Wyner-Ziv video codec.

3. **Side information generation using Bidirectional motion compensation interpolation.** In this section we will focus on the commonly used bidirectional motion estimation technique used in the DVC systems [4] known as the Bidirectional Motion Estimation with Spatial Smoothing (BiMESS). The frame interpolation framework of this technique is shown in figure 2. First, the key neighboring frames are low pass filtered to reduce the edges impact on the estimated motion vector. The aim here is more for determining the real motion field than for finding for the best match. The BiMESS algorithm is illustrated in figure 3. For a given block of the next frame (i+1), the best match in the previous frame (i-1) is found across a search range of ± pixels. The block of the interpolated frame is halfway. To avoid overlapping and uncovered regions, the motion vector is translated such as to coincide with the center of a block in the interpolated frame. Then, the motion vectors are improved by bidirectional motion estimation. It consists of a fine tuning performed by rotating and stretching the motion vector while keeping its center on the center of the block. Finally, the motion vectors are smoothed by a weighted vector median filter to correct the outliers vectors having low spatial coherence.

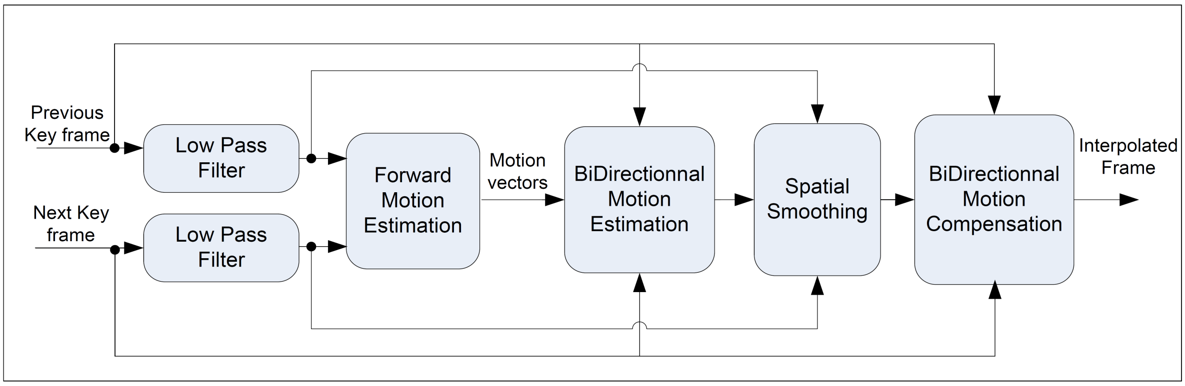


FIGURE 2. Frame interpolation framework.

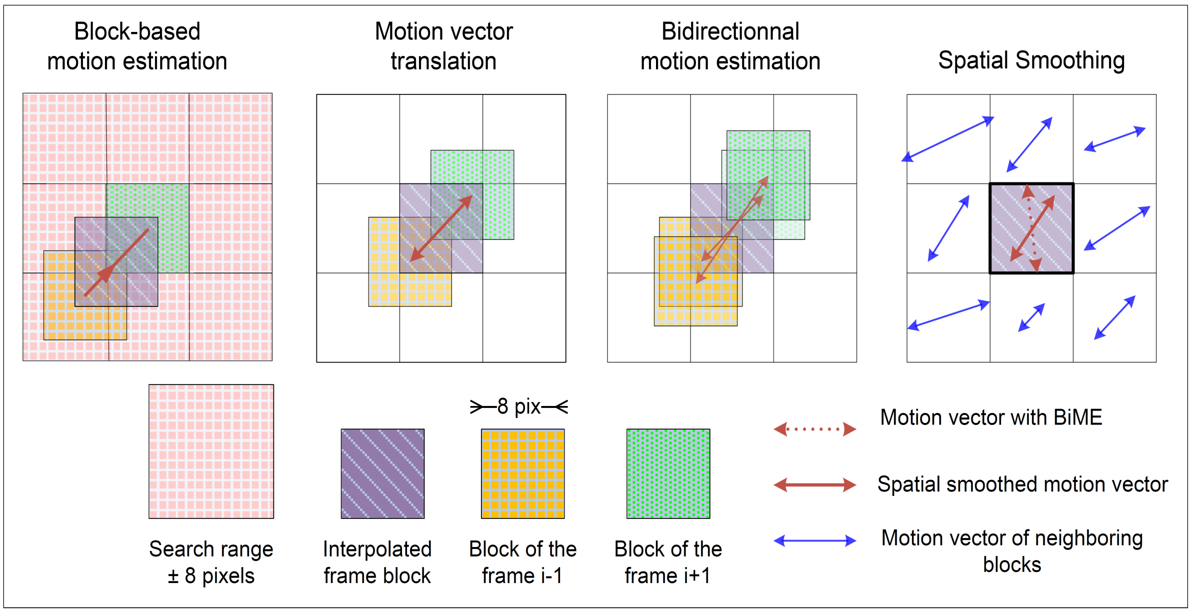


FIGURE 3. Bidirectional motion estimation with spatial smoothing.

## 4. Progressive distributed video coding scheme.

**4.1. Context and motivation.** In a conventional distributed video coding scheme, the motion vector estimation does not use the current frame as reference but instead it uses the neighboring frames in an attempt to estimate the current one. The block based interpolation in the Wyner-Ziv context is based on the assumption that each block is supposed to have a linear motion from the previous frame at t −1 to the subsequent frame at t+1 passing by t. This assumption holds in most cases. However when there is fast, irregular or complex motion, interpolation without any knowledge about the current frame may fail. An illustrative example of the block based interpolation failure is given in figure 4 where the object motion from the previous frame at t−1 to t+1 is not uniform. The estimated object position at time t is more likely to be halfway between the object positions at t −1 and at t +1. However the actual object position at time t may not as estimated and this leads to interpolation errors that requires an important number of parity bits to be corrected. To avoid this, some blocks of the reference (current) frame need to be sent first to the decoder to adjust the interpolation process for the remaining blocks of the frame. This fact lies behind the main principle of the proposed progressive architecture and will be discussed in the next section.

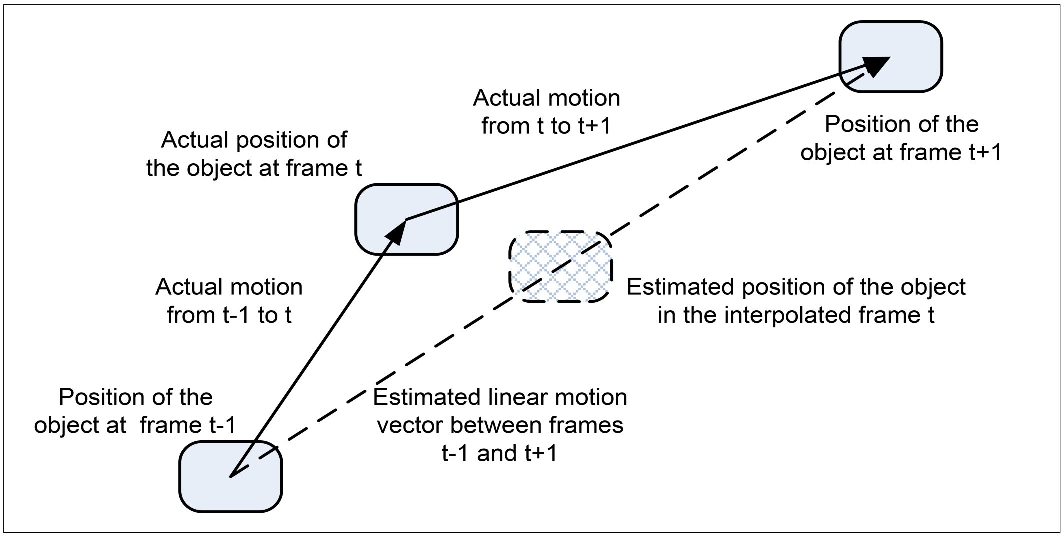


FIGURE 4. Interpolation true motion detection failure in case of non uniform motion.

**4.2. Progressive codec with side information update.** The architecture of the proposed progressive codec is depicted in figure 5. The principle of the progressive coding scheme is to divide a Wyner-Ziv frame into blocks and then group the blocks into several sets of blocks. These sets of blocks are then progressively encoded one after another using a conventional DVC coding method and transmitted one after the other to the receiver. At the receiver, these sets of blocks are progressively decoded one at the time. The previously decoded sets of blocks are used to improve the quality of the side information that is then used for decoding the current and next sets of blocks. To ensure the efficiency of the progressive side information improvement, the blocks of the successive sets have to be spatially correlated. Thus after the reconstruction of the first set, the decoder can enhance the side information quality relative to the second set by detecting the blocks where the motion is not uniform. Several patterns for WZ frame splitting can be considered as long as the spatial correlation is maintained between the sets. In this work, we study and test the grouping of the blocks from a Wyner-Ziv frame into two sets according to a Chess-Board structure demonstrating a vertical and a horizontal correlation between the two sets.

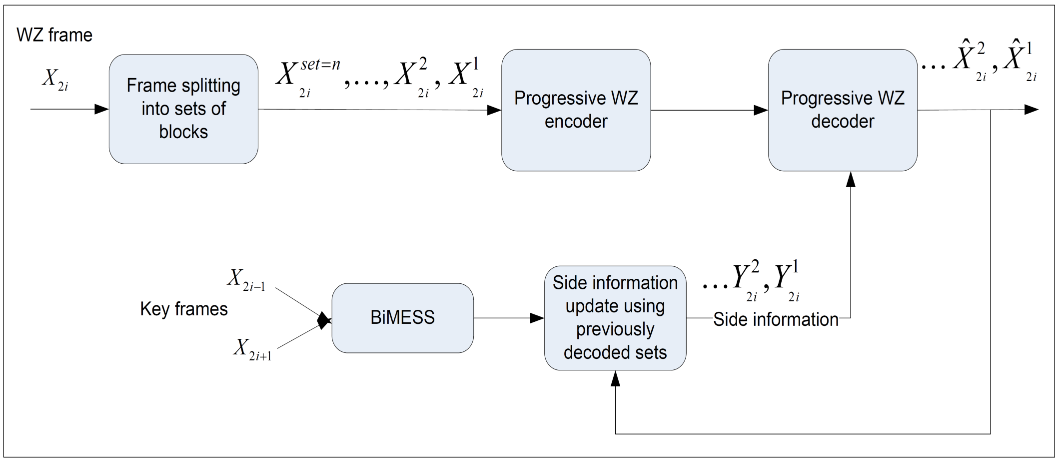


FIGURE 5. Progressive coding and progressive decoding set by set of the WZ frame.

4.3. **Chess-Board structure for block grouping.** Each Wyner-Ziv frame is divided into sets as shown in Figure 6. The first set consists of all "black" blocks and the second one consists of all "white" blocks. The encoding and decoding processes are described as follows. The set of black blocks are first encoded and transmitted using a conventional DVC coding method. At the receiver, the decoder creates side information for the Wyner-Ziv frame using a BiMESS interpolation method: the set of black blocks are decoded using the side information. Then these decoded blocks are used to improve the side information for the white blocks. After that, the encoder encodes and transmits the set of white blocks using the same coding method and the receiver decodes the white blocks using the improved side information. As the side information for the white blocks has been improved, the white blocks should need fewer bits to encode and decode, therefore resulting in a reduction of the total bit rate.

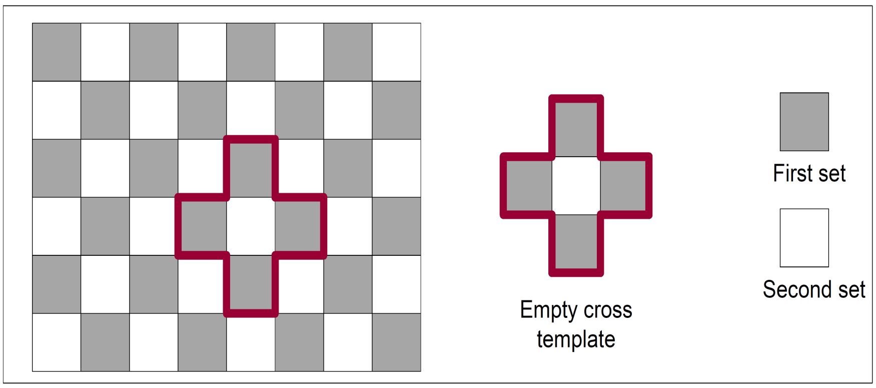


FIGURE 6. Division of blocks into two sets leading to an empty cross motif.

4.4. **Motion estimation at the decoder.** There are a number of possible approaches for improving the quality of the side information from the first set of decoded (black) blocks. Each white block (not yet received) is surrounded by four black blocks (already decoded). This can be considered as an empty cross as shown in figure 6. We first take the previously decoded key frame as a reference frame and find the best matching of each empty cross in this reference frame. The matching criterion considered is the Mean of Absolute Differences (MAD): it is computed from the four blocks surrounding the central block. Once the empty cross best matched position is found in the previous frame, the central block is considered as a first estimate of the white block. The same approach is applied for the next key frame and for the interpolated frame. This method is further illustrated in figure 7. At the end of the process there are three estimates for the white block in the three different reference frames (previous, next and interpolated frames). Motion compensation is then done for the white block using two different methods as explained in the next section.

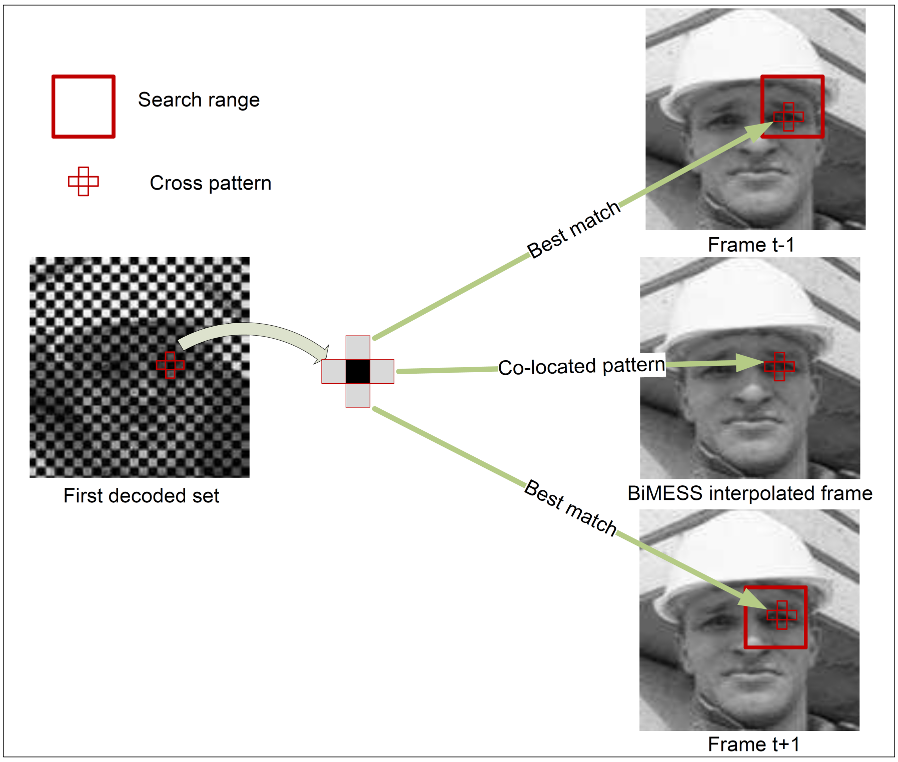


FIGURE 7. Side information update algorithm.

4.5. **Motion compensation for side information updating.** For motion compensation with side information updating, the empty block of the cross (white block) is filled according to the three best matches found in the previous step. Two motion compensation methods are considered here for the progressive architecture.

1. First method: First, the two best matched empty crosses in the previous and the forward frames are found. For each of these empty crosses, the mean absolute differences with the empty cross from the first decoded set is computed. The resulting values are denoted $MAD_{prev}$ and $MAD_{forw}$. Then the MAD between the empty cross from the first decoded set and its co-located counterpart in the interpolated frame are computed. The obtained value is denoted $MAD_{inter}$. Finally, the cross with the minimum MAD is selected and its central block is selected as the side information for decoding the second block set. However, this method is not statistically optimal and is sensitive to the errors that can occur at the previous stage. In fact, the first decoded set is not error free and the search range displays a high spatial

redundancy. The empty cross best match searching can induce some errors and using the three different candidates to fill the empty block is more reliable.

2. Second method: It consists of summing the three blocks estimated from the previous stage weighted by coefficients related to the matching criterion. Although it is not the best match that is selected, from a statistical point of view, this motion compensation method performs better than the first motion compensation method. The compensated block, $Block_{comp}$, is given by:

$$Block_{comp} = \frac{\frac{1}{MAD_{prev}} Block_{prev} + \frac{1}{MAD_{forw}} Block_{forw} + \frac{1}{MAD_{\text{inter}}} Block_{\text{inter}}}{\frac{1}{MAD_{prev}} + \frac{1}{MAD_{forw}} + \frac{1}{MAD_{\text{inter}}}} \quad (1)$$

Here, $MAD_X$ refers to the Mean of Absolute Differences between the empty cross pattern obtained from the first decoded set and the best empty cross pattern found in the previous ($prev$), forward ($forw$) or interpolated ($inter$) frame. $Block_X$ is then the block inside the empty cross. Factors $\frac{1}{MAD_x}$ ($X = prev, forw \text{ or } inter$) in (1) indicate the matching level for each block inside the empty cross from the previous, forward or interpolated frame. For a large $MAD_{prev}$ value, the block inside the empty cross from the previous frame will contribute only slightly in the compensated block.

5. **Simulation results.** In this section we present the improvement obtained by the progressive coding scheme over one of the best results found in literature by Brites et al. [6]. The transform domain Wyner-Ziv coding scheme proposed by Brites et al. is based on a previous work of Aaron et al. [7] and includes several improvements such as:

1. Conception of a new quantizer for the DCT coefficients: The dynamic range of each coefficient is sent by the encoder and, for the AC coefficients concentrated around zero, the quantizer displays a symmetrical dead zone.
2. Advanced frame interpolation technique: Bidirectional motion estimation with spatial smoothing framework is used for side information generation.
3. Online correlation noise modeling at the coefficient level: For each coefficient, a Laplacian parameter $\alpha$ is estimated online (at the decoder side) ensuring more efficient turbo decoding process.

The combination of these tools has earned the Brites et al. architecture to be one of the most powerful DVC scheme in the literature. For comparison purposes, some results from Brites et al. method are reproduced in this paper. The implementation phase of the transform domain DVC architecture is also validated. The experiment considered in [6] assumed that the key frames were perfectly known at the decoder. Then, only the rate- distortion function of the WZ frames is reported. The simulation results consider four QCIF video sequences at 30 fps (frames per second): Foreman, Mother and Daugther, Salesman and Carphone. These sequences were downloaded from the Stanford Center for Image Systems Engineering website [8]. As in [6], only the first 101 frames are considered.

For this implementation, the two motion compensation methods presented in the previous section are considered for the progressive scheme. The scheme using the first method which consists of just copying the best match block is denoted as the progressive scheme while the second scheme consisting in summing the weighted blocks is called weighted progressive scheme. The level of distortion is computed using two distortion measures: the commonly used objective measure for image processing, the PSNR, and the SSIM (Structural Similarity) index to take account the spatial coherences in the distortion evaluation. The simulation results consider the optimal reconstruction technique presented by Kubasov and Guillemot in [9].

Table 1. PSNR (in dB) of the interpolated two groups of blocks during the weighted progressive scheme.

| | set 1 (BiMESS generated SI) | set 2 (Updated SI) |
|---|---|---|
| Foreman | 36.7823 | 38.4369 |
| Carphone | 35.0049 | 37.9426 |
| Salesman | 44.2555 | 44.8114 |
| Mother and Daughter | 42.3797 | 43.7461 |
| Average | 39.606 | 41.234 |

To investigate the side information quality improvement of the second set (updated side information) in comparaison with the first set (BiMESS generated side information) , we report, in table 1 the PSNR of the interpolated frames for the different sets of the proposed progressive DVC scheme. This table shows the side information quality of each group of blocks by averaging over the different frames of the different video sequences. According to the table 1, the side information update can improve the interpolated frame PSNR by 1.628 dB.

To demonstrate the relevance of the progressive scheme over the conventional scheme, figure 8 gives the percentage of blocks compensated, using the previous and forward frames, when compared to the interpolated frame. It is observed that when the PSNR of the interpolated side information (PSNRint) is lower, the use of the neighboring frame during the progressive side information update is more important. For instance, the PSNR with interpolation for the Carphone sequence is the lowest one of the four sequences. Thus, the side information update uses more often the neighboring frames for the Carphone sequence than for the other sequences. Furthermore, the simulations results depicted in figure 9, show a larger increase in the PSNR performances over the conventional scheme when the side information is initially more corrupted. In fact, we observe an improvement of approximately 1.5 dB for the progressive scheme above that of the conventional scheme for the Carphone sequence and only a 0.6 dB improvement for the Salesman sequence. For a higher group of picture value (e.g. GOP = 4 or GOP = 8), the improvement of the progressive scheme is expected to be more important: this however requires future investigation.

Finally we report the improvement observed for the weighted progressive scheme compared to the progressive and the conventional schemes. In figure 9, the rate-distortion performances is shown for these 3 schemes using optimal reconstruction. The improvement observed for the progressive weighted scheme compared to the conventional one ranges from 0.8 dB to 1.7 dB. The use of a weighted compensation technique is more efficient than the best match compensation technique for the progressive scheme. According to figure 9, an improvement of up to 0.4 dB is noticed when using the weighted compensation.

In these simulations, the SSIM distortion measures were also computed to get an indication about the subjective quality of the decoded video sequences. The SSIM results show that the performance behavior of the rate-distortion curves is the same whether the PSNR or SSIM measures are used. The improvement obtained by the progressive coding scheme is more pronounced for the high rates quantization matrices. In fact, for a high rate matrix, the reconstruction of the first block set, to be exploit to improve the side information for the second block set, is more accurate. In other words, the empty cross used for the searching the best match has lower distortion and thus is more reliable for the improvement of the side information for the white blocks.

To extend the comparative study, it is worth stating that the progressive coding leads to an additional computational cost in the side information generation component. As

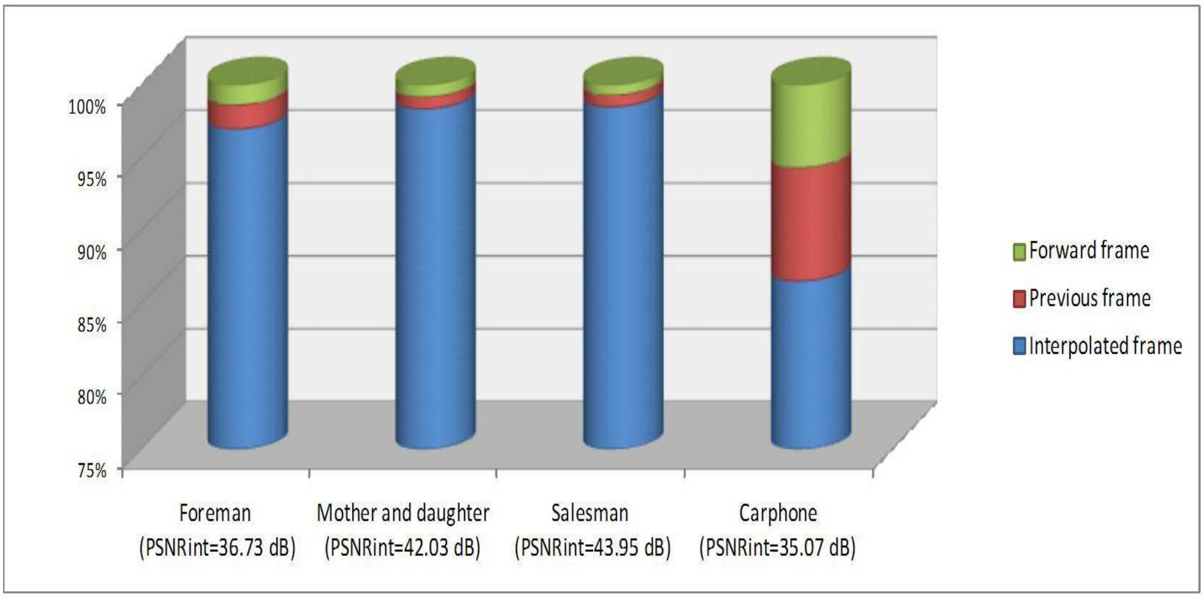


FIGURE 8. Percentage use of the neighboring frames compared to the interpolated frame during the side information update.

this component is on the decoder side, no additional implementation cost is required at the encoder and the paradigm of the distributed video coding is preserved. The side information update task leads to a better interpolation frame quality. Consequently the turbo decoding process, consuming the most decoder resources, will be faster. In fact, for better side information quality fewer parity bits are needed and less feedback loops are executed. At last the progressive scheme does not inquire additional decoder resources since it accelerates the decoding process.

**6. Conclusion.** In this paper, a new DVC architecture allowing progressive coding is proposed. The side information is being progressively updated as long as the WZ frame is being decoded. There are a number of possible approaches to divide the blocks into sets of blocks and to use the decoded blocks to improve the quality of side information. The implemented progressive technique considers a chess-board structure for block grouping and an empty cross best match searching in three reference frames. Noticeable PSNR improvements of up to 1.7 dB, as well as in terms of the SSIM, over the conventional DVC architecture are reported. The same concept can be further explored by other methods of block grouping and side information updating.

## REFERENCES

[1] A. Aaron, R. Zhang and B. Girod, Wyner-ziv coding of motion video, *Proc. of Asilomar Conference on Signals, Systems and Computers*, 2002.
[2] Anhong Wang, Yao Zhao and Lei Wei, Wavelet-domain distributed video coding with motion-compensated refinement, *Proc. of IEEE International Conference on Image Processing*, 2006.
[3] Anhong Wang, Yao Zhao and Jeng-Shyang Pan, Residual distributed video coding based on LQR-hash, *Chinese Journal of Electronics*, vol. 18, no. 1, pp. 109-112, 2009.
[4] J. Ascenso, C. Brites and F. Pereira, Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding, *Proc. of the 5th Conference on Speech and Image Processing*, 2005.
[5] C. Brites and F. Pereira, Correlation noise modeling for pixel and transform domain wyner-ziv video coding, *IEEE Trans. Circuits and Systems for Video Technology*, vol. 18, no. 9, pp. 1177-1190, 2008.
[6] C. Brites, J. Ascenso and F. Pereira, Improving transform domain wyner-ziv video coding, *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing*, pp. 525-528, 2006.

[7] A. Aaron, S. Rane, E. Setton and B. Girod, Transform-domain wyner-ziv codec for video, *Proc. of SPIE Visual Communications and Image Processing*, pp. 520-528, 2004.

[8] Stanford Center for Image Systems Engineering, Test images and videos, Stanford University, http://scien.stanford.edu/pages/labsite/scien-test-images-videos.php.

[9] D. Kubasov, J. Nayak and C. Guillemot, Optimal reconstruction in wyner-ziv video coding with multiple side information, *Proc. of IEEE International Conference on Malignant Melanoma of Soft Parts*, pp. 183-186, 2007.

# Appendix I

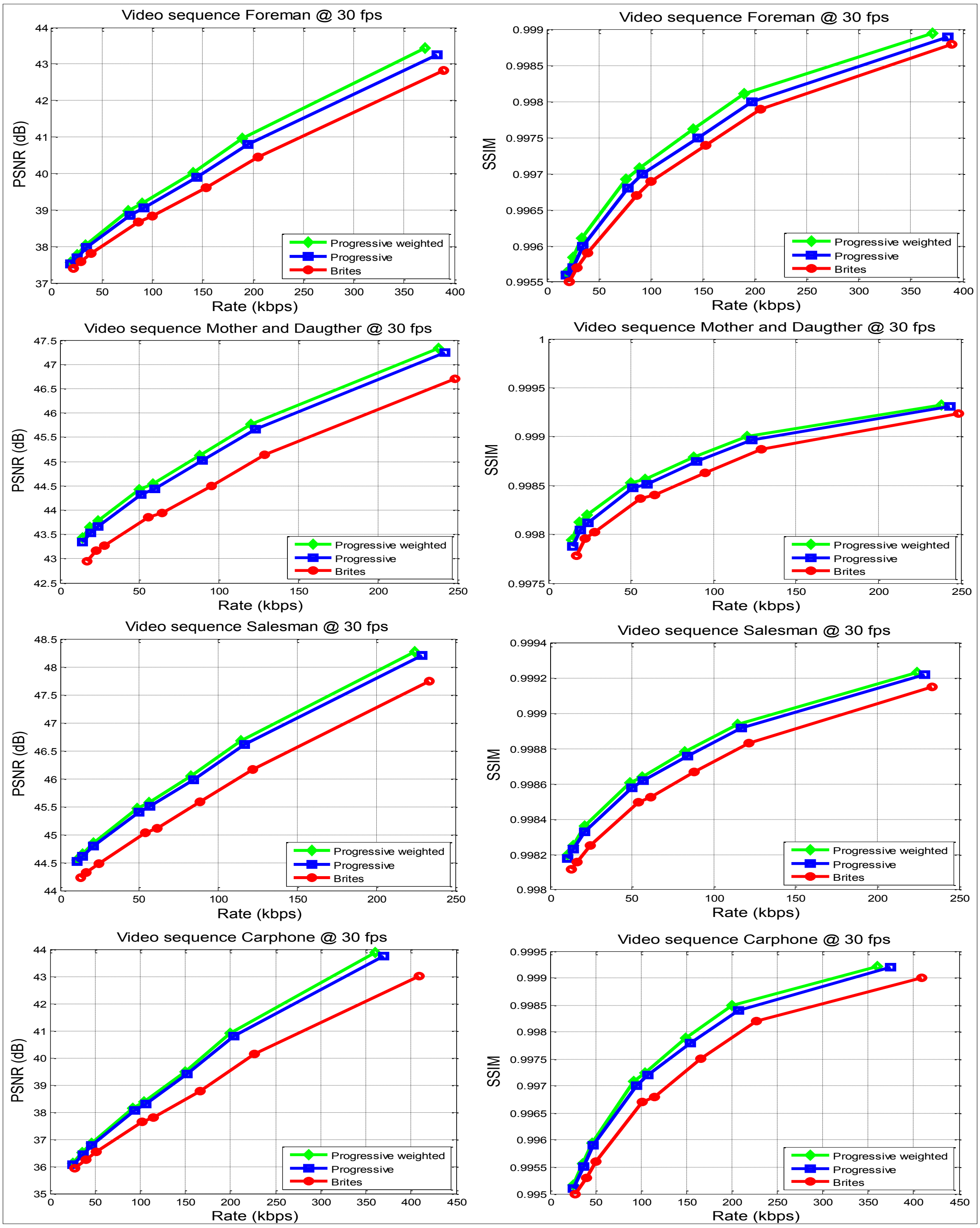


FIGURE 9. *PSNR and SSIM performance simulation results.*