

RESEARCH

Open Access

Spatial correlation-based side information refinement for distributed video coding

Mohamed Haj Taieb^{1*}, Jean-Yves Chouinard¹ and Demin Wang²

Abstract

Distributed video coding (DVC) architecture designs, based on distributed source coding principles, have benefitted from significant progresses lately, notably in terms of achievable rate-distortion performances. However, a significant performance gap still remains when compared to prediction-based video coding schemes such as H.264/AVC. This is mainly due to the non-ideal exploitation of the video sequence temporal correlation properties during the generation of side information (SI). In fact, the decoder side *motion estimation* provides only an approximation of the *true motion*. In this paper, a progressive DVC architecture is proposed, which exploits the spatial correlation of the video frames to improve the motion-compensated temporal interpolation (MCTI). Specifically, Wyner-Ziv (WZ) frames are divided into several spatially correlated groups that are then sent progressively to the receiver. SI refinement (SIR) is performed as long as these groups are being decoded, thus providing more accurate SI for the next groups. It is shown that the proposed progressive SIR method leads to significant improvements over the Discover DVC codec as well as other SIR schemes recently introduced in the literature.

Keywords: Distributed video coding; Side information refinement (SIR); Motion estimation; Spatial correlation

1 Introduction

Digital video coding standards have steadily evolved in order to achieve high compression performances using sophisticated, but increasingly complex, techniques for accurate motion estimation and compensation. These compression tasks are typically executed at the encoder, resulting in a high computational load. At the decoder, video sequences can be easily reconstructed by exploiting the motion vectors already computed at the encoder. This computational effort distribution between the encoder and decoder is well suited to common video transfer applications such as broadcasting and video streaming, where powerful encoders are typically used to compress the video sequences only once before sending them to several low-cost computationally limited video decoding devices.

However, the emergence of locally distributed wireless surveillance cameras, cellular interactive video devices, and other applications involving several low-cost video encoders have driven recent research efforts towards the

development of video standards with the opposite complexity reallocation between the encoders and decoders. Indeed, high-complexity operations, such as motion estimation, need to be done at the decoder side instead of at the encoder. Slepian and Wolf information theoretical results on lossless coding for correlated distributed sources [1] and their extension to lossy source coding with side information at the decoder by Wyner and Ziv [2] constitute the theoretical basis for this new distributed video coding paradigm.

Although these theoretical foundations were laid down back in the 1970s, the first actual distributed video coding (DVC) implementations have been proposed in 2002 by Ramchandran et al. [3,4] and Girod et al. [5,6]. Based on these [7], the European Distributed Coding for Video Services (DISCOVER) consortium [8] have investigated and proposed new DVC coding schemes as well as design tools for improving the rate-distortion performances while addressing practical issues such as scalability and robustness against transmission errors.

A critical component of the DVC architecture is the generation of the side information from neighboring key and Wyner-Ziv frames. Discover has adopted a motion-compensated temporal interpolation (MCTI) technique

*Correspondence: mohamed.haj-taieb.1@ulaval.ca

¹Laval University, Quebec, Quebec, G1V 0A8, Canada

Full list of author information is available at the end of the article

known as *bidirectional motion estimation with spatial smoothing* (BiMESS). Despite the improvements achieved with BiMESS interpolation, temporal correlation is still not as fully exploited in distributed video coding as it is in predictive video coding. Motion estimation is performed without knowledge of the original Wyner-Ziv (WZ) frame: it assumes linear motion. Even though this assumption holds fairly well in many cases, for high-speed video sequences, motion may no longer be linear and, in such cases, the interpolation is more likely to fail.

Based on our prior work [9], a novel progressive DVC architecture is proposed to mitigate the limitations of such *blind motion estimation* and to further explore the intrinsic spatial redundancy of each frame. The proposed side information refinement (SIR) technique works as follows. Each WZ frame is divided into blocks, and the blocks themselves are grouped into two, three, or four spatially correlated sets. The first set of blocks is sent to the receiver and is then used for side information refinement of the second set. The SIR is based on the spatial correlation between the first decoded set and the upcoming second set. Thus, the second set of blocks will require lower bitrates since the side information will be affected with less distortion after the first pass of refinement. The SIR process is repeated for the next sets of blocks using all the previously decoded sets, thus benefiting more and more from the video frame's spatial correlation.

The SI refinement is based on motion estimation using various templates composed of the previously reconstructed sets. These templates display a certain level of correlation with the block subject to refinement. To assess the relevance of the proposed technique, this paper first investigates the template ability to ensure the side information refinement. A new metric, termed spatio-temporal correlation factor ρ , is defined to quantify at which extent does the block spatially correlated templates are able to refine the temporal motion estimation using the neighboring frames.

The overall progressive DVC scheme is evaluated by comparing the rate-distortion (RD) performances to the state-of-the-art Discover DVC codec for five Quarter Common Intermediate Format (QCIF) sequences exhibiting different camera and motion attributes. The proposed architecture is also compared to recent DVC systems using successively refined side information (Martins et al. [10] and Deligiannis et al. [11]) in terms of the Bjøntegaard metrics [12] (Bjøntegaard delta rate and peak signal-to-noise ratio (PSNR) metrics) according to the non-refined DVC system DISCOVER. It is shown that the proposed progressive SIR method does improve the performances over the system introduced by Martins et al. [10] and Deligiannis et al. [11] for most of the test scenarios investigated in this paper.

This paper is organized as follows. Section 2 describes the Discover codec basic architecture and components. Section 3 presents an overview of the on-going research efforts on alternative techniques for SI generation and side information refinement methods. In section 4, different progressive architectures, allowing for spatial correlation based SIR techniques, are proposed. Section 5 demonstrates the PSNR improvement obtained at each refinement stage and the overall RD performance improvement. A complexity analysis is also presented in this section.

2 Overview of the Discover DVC codec

The architecture of the Discover [8] WZ system is depicted in Figure 1. As shown in this figure, the key frames are H.264/AVC intra-encoded (intra-frames) and transmitted to the H.264/AVC intra-decoder which reconstructs the intra-frames and also generates the side information that will be used to decode the Wyner-Ziv frames (inter-frames).

At the Wyner-Ziv encoder, the interframes are compressed using an integer 4×4 block-based discrete cosine transform (DCT). The DCT coefficients are then fed to a uniform quantizer, and the bitplanes are extracted. The bitplanes are in turn fed to a turbo encoder with two rate $1/2$ recursive systematic convolutional (RSC) encoders. Each RSC associates the parity bits to the bitplanes. To achieve compression, the systematic bits are discarded since the decoder has already an interpolated version of the WZ frames. The parity bits are stored in a buffer and sent gradually, packet by packet, upon decoder feedback requests according to a periodic puncturing pattern. The feedback channel helps in adapting the forward transmission rate to the time-varying virtual channel statistics. The WZ decoding process implies several turbo decoding iterations. To alleviate the decoder computational hurdle, an initial number of parity bits packets is estimated by way of a hybrid encoder/decoder rate control mechanism [13]. These parity bits packets are sent once to the decoder, and subsequent packets will eventually be sent afterwards.

At the WZ decoder, an interpolated version of the current WZ frame is produced from the neighboring reconstructed frames. The BiMESS motion-compensated temporal interpolation technique introduced in [14] is used for the Discover DVC codec. The MCTI BiMESS performances are improved using a hierarchical coarse-to-fine approach in bidirectional motion estimation [15] and subpixel precision for motion search [16]. The interpolated frame is then DCT-transformed: these DCT coefficients represent the side information for the Wyner-Ziv decoder. The WZ DCT coefficients are modeled as the input of a virtual channel and the side information as its output. For the turbo decoding process, a Laplacian model is assumed for this virtual channel. The estimation of the Laplacian distribution parameter α is based on the online

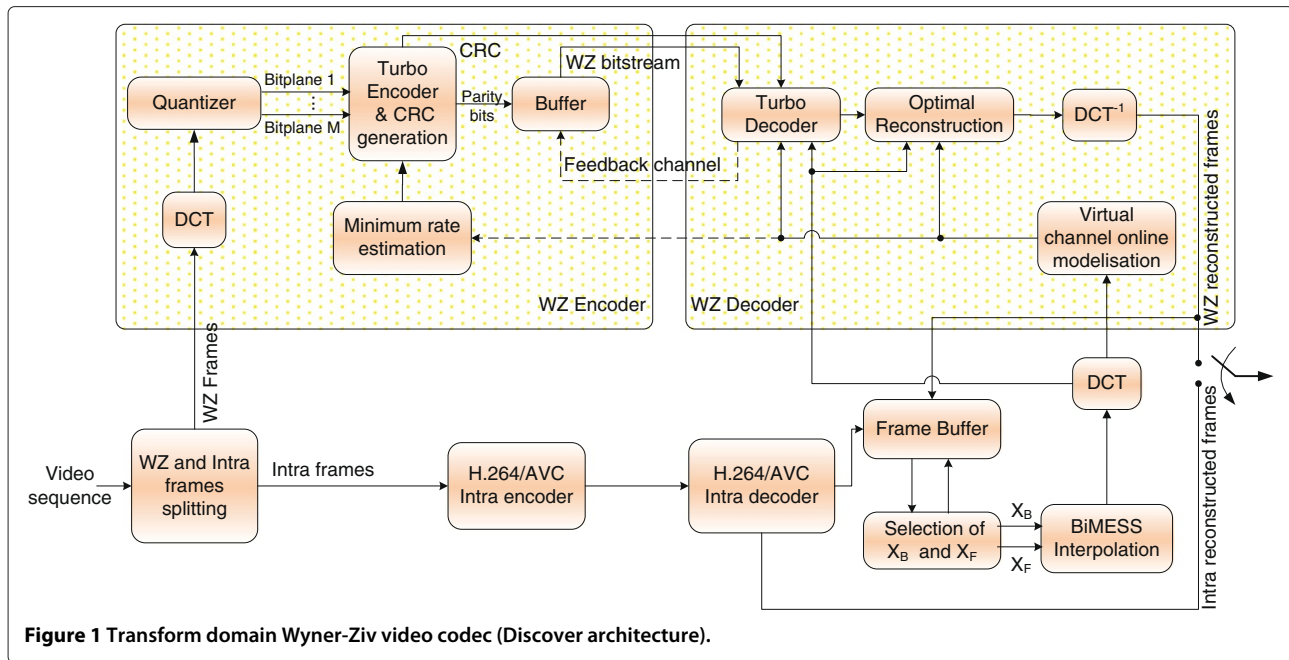


Figure 1 Transform domain Wyner-Ziv video codec (Discover architecture).

correlation noise modeling technique developed by Brites and Pereira [17]; parameter α is estimated for each coefficient of each DCT band. Alternative *on-the-fly* estimation methods are proposed in [18] and [19] to track the unpredictable and dynamic temporal changes within a video sequence. The correlation noise parameter is refined iteratively during the decoding instead of pre-estimating this parameter before decoding [17].

The side information, replacing the systematic information in the turbo decoding process, is thus corrupted by the Laplacian noise whose parameter is, beforehand, online estimated (without using original data). The received parity bits along with the side information are fed to the turbo decoder. After a number of iterations, the log-likelihood ratios are computed and then the bitplane is deducted. To estimate the decoded bitplane error rate, without knowledge of the original data, these log-likelihood ratios are used to compute a *confidence score* [13]. If this score exceeds 10^{-3} , then a *parity bits* request is sent back to the turbo encoder. Otherwise, the decoding process is likely to be satisfactory. However, some errors can still persist even if the confidence score is below the 10^{-3} threshold. For this reason, a cyclic redundancy check (CRC) code is used to detect the remaining bitplane decoding errors. If the decoded bitplane CRC corresponds to the original data CRC, then the decoding process is considered successful; otherwise, more parity bits are requested. Using jointly the confidence score and the CRC code results in error detection performances as good as for ideal error detection where the decoded bitplane is directly compared to the original bitplane [13].

After being decoded, the different bitplanes are recombined to form the quantization symbols. These symbols and the side information are used to reconstruct the DCT coefficients. An optimal reconstruction function is proposed in [20] to minimize the mean squared error according to the Laplacian correlation model. For coefficient bands that have not been transmitted, the side information is directly considered in the reconstruction. Finally, an inverse 4×4 DCT is applied to the reconstructed DCT band to restore the WZ frame back in the pixel domain.

3 Alternative SI generation techniques

Side information generation techniques (interpolation or extrapolation) in DVC architectures are affected by the absence of any information about the current WZ frames. For these, it is typically assumed that each block motion is *changing linearly* between the successive frames. Recent research works addressing the inefficiency issues inherent to *blind SI generation* are presented hereafter.

For low-delay WZ video coding, the side information is usually generated by extrapolating the two previously decoded frames. However, since no future frames are used, the motion estimation process becomes more problematic. Moreover, the motion extrapolation is based on potentially poorly reconstructed WZ frames. To mitigate this error propagation-like condition, Agrafiotis et al. [21] proposed a SI generation method based on a hybrid Key/WZ macroblock partitioning, according to a *chess-board structure*. The intra-coded macroblocks are first decoded, and then the missing macroblocks are estimated using four neighboring macroblocks during the motion estimation. In [22], auxiliary information is generated by

considering the discrete wavelet transform (DWT) compaction property. It consists of the low-low (LL) wavelet subband of the current WZ frame. At the decoder, the LL subband is upsampled by inverse DWT to refine the motion estimation using the previous reconstructed frame. In [23], the extrapolation is aided by using robust hash codewords consisting of a coarsely quantized version of each block in the WZ frame. The distance between the hash codeword for a given block and the corresponding hash codeword of the previous frame is computed at the encoder. According to this distance, the encoder decides if the hash codeword should be sent to help the decoder motion estimation.

To avoid performing transformations, e.g., DCT or DWT, at the source encoder, pixel domain DVC architectures were proposed in [24] and [25] where the spatial redundancy is exploited at the decoder side. In [24], each WZ frame is split into two sets according to a checkerboard pattern. The first SI subset is generated by extrapolation. The second subset has access to two SI components: a *temporal extrapolation* component and a *spatial interpolation* of the first decoded set component. Depending on the local difference between the temporal SI and the first decoded set, the decoder decides whether or not the spatial SI should be used. In [25], however, instead of selecting a single SI component, the turbo decoder is extended to handle likelihood value calculations based on both the temporal and spatial SI components.

In [26], the SIR is based on an iterative bitplane decoding algorithm since each decoded bitplane brings additional information that can be used to generate better SI for the subsequent least significant bitplanes. In [27], using MCTI SI, the WZ decoder is launched leading to a *partially decoded* WZ (PDWZ) frame. Then, motion refinement is performed using the neighboring frames and the PDWZ frame, leading to a better SI that is fed again to the WZ decoder. The decoding complexity is doubled for a modest improvement of 0.15 dB. A similar iterative approach is proposed in [28] involving motion refinement framework using the PDWZ frame to improve the SI quality. This approach leads to an improvement up to 1 dB in the reconstructed frames without reducing the overall bitrate. The decoding complexity is also doubled.

In [29] an iterative SIR technique is considered for a *feedback-free* DVC architecture. For this encoder-driven rate control scheme, the channel decoder may fail using the number of parity bits estimated at the encoder. Since the feedback channel is suppressed, the decoder reattempts decoding with the same number of parity bits but using an improved SI. The SI is improved iteratively through successive refinement levels (RLs). For the first RL (RL_0), the DC coefficients are reconstructed and

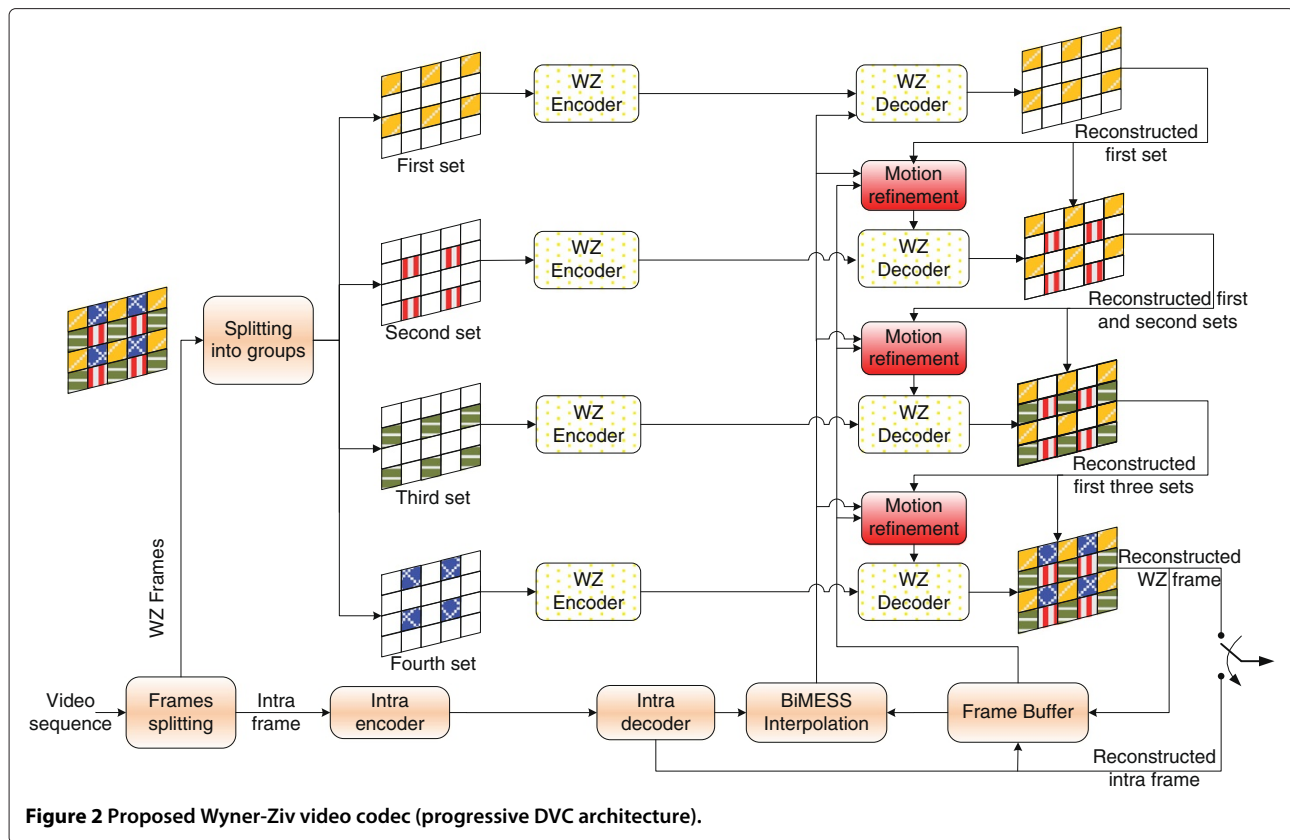
used to refine the SI. Then, in RL_1 , two other DCT bands are decoded along with the DC coefficients if the decoding fails at RL_0 . Thus, the DC band SI can be improved using the information brought by subsequent bands. This process is iterated for RL_2 where a fully decoded and reconstructed WZ frame is produced. If some failed bitplanes still persist, supplemental RLs are added. A similar SIR technique with four and six RLs is considered in [11] for a feedback channel-based architecture. This latter technique will be considered in the present paper for comparison with the proposed progressive technique using the Bjøntegaard delta rate and PSNR metrics [12].

In [30], the PDWZ frame consists of the DC (zero frequency) component which is used for SIR of the other DCT components. While the motion search during the SIR process is conducted in the DC domain (i.e., between the PDWZ frame and the neighboring frames' DC component), motion compensation itself is performed in the pixel domain, after regaining the full image resolution. After decoding a number of DCT bands, the PDWZ frame is updated to take into account the new frequency components. Similarly in [10], it is proposed to refine the side information at the DCT band level by processing motion search in the pixel domain. More specifically, after decoding each DCT band, an inverse DCT is applied leading to the PDWZ frame. Motion estimation is thus conducted between the PDWZ pixel domain frame and the neighboring frames without applying the DCT. The results given in [10] will later be used for comparison to assess the performances of the proposed progressive architecture.

4 Proposed progressive DVC scheme

The principle of the proposed progressive coding scheme is to partition a Wyner-Ziv frame into 4×4 pixel blocks and then group these blocks into several sets, as shown in Figure 2. These sets of blocks are then progressively encoded using conventional DVC coding (WZ encoding) and transmitted one after another to the receiver. At the receiver, they 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 set. To ensure side information improvement, the successive sets of blocks must be spatially correlated.

Several patterns for WZ frame splitting can be considered as long as spatial correlation is maintained between the sets. Motion refinement is performed at the pixel level after inverse DCT transformation and after reconstructing the previous sets. Three different variants of the proposed progressive coding scheme are described below, using respectively two, three, or four sets of (spatially correlated) blocks.



The *generic algorithm form* of the progressive scheme is shown in Figure 2 and given by the following steps:

1. WZ encoding of the first set.
2. WZ decoding and reconstruction of the first set using the conventional SI generated by MCTI (without any refinement).
3. WZ encoding of the next set.
4. Based on all previously decoded sets, a refinement template is constructed to improve the SI quality of the current set.
5. WZ decoding and reconstruction of the current set using the refined SI.
6. Go to step 3 until all the WZ frame sets are reconstructed.

4.1 Progressive DVC using two sets of blocks

Here, each Wyner-Ziv frame is divided into two sets of blocks according to a *chessboard* structure as shown in Figure 3: the first set consists of all the *black* blocks and the second one consists of all the *white* blocks. The set of black blocks is first encoded and transmitted with conventional DVC coding. This set of blocks is decoded using the DCT-transformed interpolated frame as side information. These decoded blocks are used to improve the side information for the white blocks. Then, the encoder

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, fewer bits are needed for the decoding processes, hence reducing the overall bitrate. Moreover, the reconstruction using the refined SI will yield to better quality.

Each white block (not yet received) is surrounded by four black blocks (already decoded). This corresponds to the *empty cross* template shown in Figure 4. Using the previously decoded key frame as a reference, the algorithm searches for the best match to the empty cross template. A search area of 28×28 pixels is considered. Thus, the displacement of the 12×12 template is 8 pixels in each direction. The search area needs to be large enough to capture fast motion especially for high group of pictures (GOP) values. However, to avoid capturing noise instead of true motion, the size of the search area cannot be indefinitely increased. The matching criterion considered here is the *mean of absolute differences* (MAD): it is computed from the four (black) blocks surrounding the central (white) block. Once the empty cross best matching 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. For the interpolated frame, however, the co-localized central white

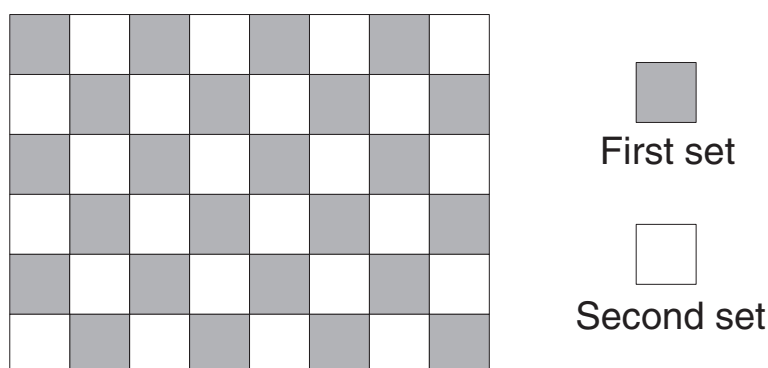


Figure 3 Partition of the Wyner-Ziv frame into two sets of blocks according to a chessboard structure.

block is taken without best match searching (Figure 4). At the end of the process, there are three estimates for the white block: B_P , B_N , and B_I from the previous, next, and interpolated reference frames, respectively.

The empty block (white block) inside the cross template is filled (motion compensated) by a weighted summation

of the three blocks, inside the empty crosses, estimated from the previous stage: B_P , B_N , and B_I . The weighting coefficients correspond to the inverse of the MAD matching criterion between the empty cross pattern obtained from the first decoded set and the best empty cross pattern found in the previous and next frames (MAD_P ,

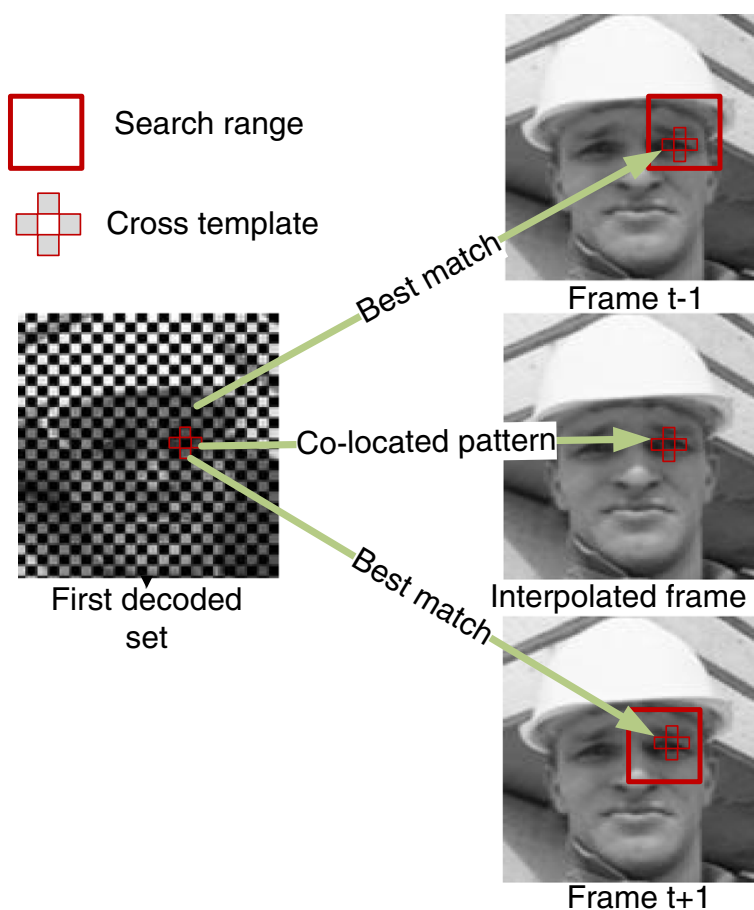


Figure 4 Progressive scheme with a single-pass side information update.

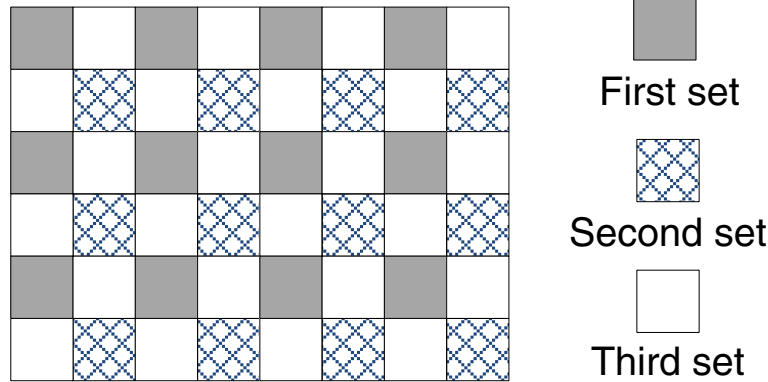


Figure 5 Partition of the Wyner-Ziv frame into three sets of blocks.

MAD_N) and the co-located pattern in the interpolated frame (MAD_I). The compensated block, B_{COMP} , is given by

$$B_{COMP} = \frac{\frac{1}{MAD_P} B_P + \frac{1}{MAD_N} B_N + \frac{1}{MAD_I} B_I}{\frac{1}{MAD_P} + \frac{1}{MAD_N} + \frac{1}{MAD_I}}. \quad (1)$$

The weighted average compensated block is statistically more appropriate than considering only the best match or the average of the three blocks. In this context, simulation tests have been conducted showing that the best match compensation approach is more sensitive to noise in fast or complex motion locations. Furthermore, the weighted average block tends towards the best match block if the two other blocks appear to be *unsuitable* (i.e., with high MAD). The unsuitable blocks will be weighted according to the inverse of the MAD.

4.2 Progressive DVC using three sets of blocks

In the previous progressive DVC scheme, only the second set benefits from the side information refinement. On the other hand, the first set, equivalent to half of the frame, is decoded using the interpolated frame. Thus, the progressive side information updating is restrained to only half the frame. In light of this, one may consider splitting the frame into more groups. However, the effectiveness of the progressive scheme depends strongly on the spatial correlation between the already decoded sets and the next set to be decoded. Thus, there is a trade-off that must be taken into account when choosing the number of sets and the location of each set. Therefore, a splitting arrangement of the Wyner-Ziv frames into three sets is considered as shown in Figure 5. These three sets of blocks display mutual spatial correlation which is necessary for the progressive architecture. Otherwise, decoding one set of blocks would not bring any further information

about the next ones and thus would not contribute to the SI refinement.

For the progressive video coding with three sets of blocks, the side information is updated with two refinement passes. The first side information refinement pass

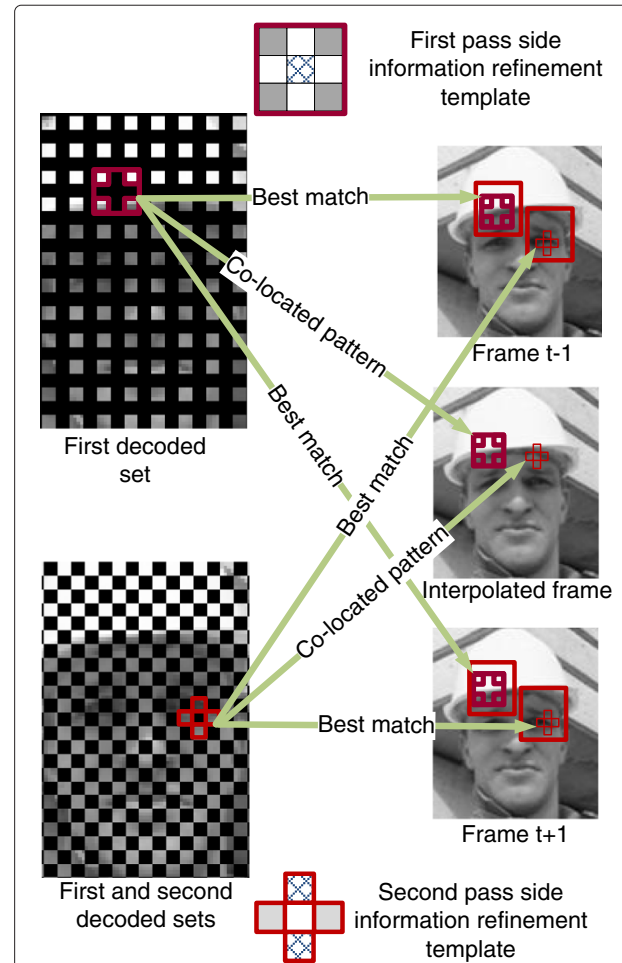
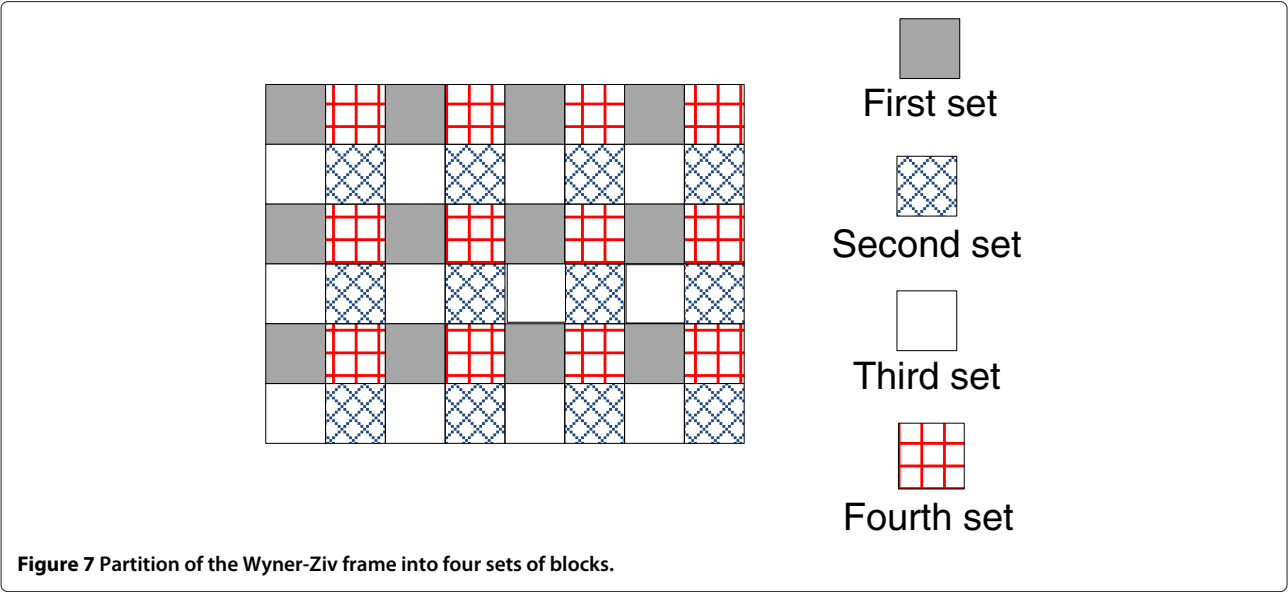


Figure 6 Progressive scheme with two-pass side information update.



uses a template consisting of the four diagonal neighboring blocks of the block to be updated. The second side information update considers the same cross template as for the previous progressive scheme with two sets of blocks (Section 4.1). Figure 6 depicts how the two side information refinement passes are done. It shows the reconstructed frame after decoding the first set bringing out the template of four diagonal blocks. A best match search of this template is conducted within a 28×28 pixel area of the previous and next frames. When the

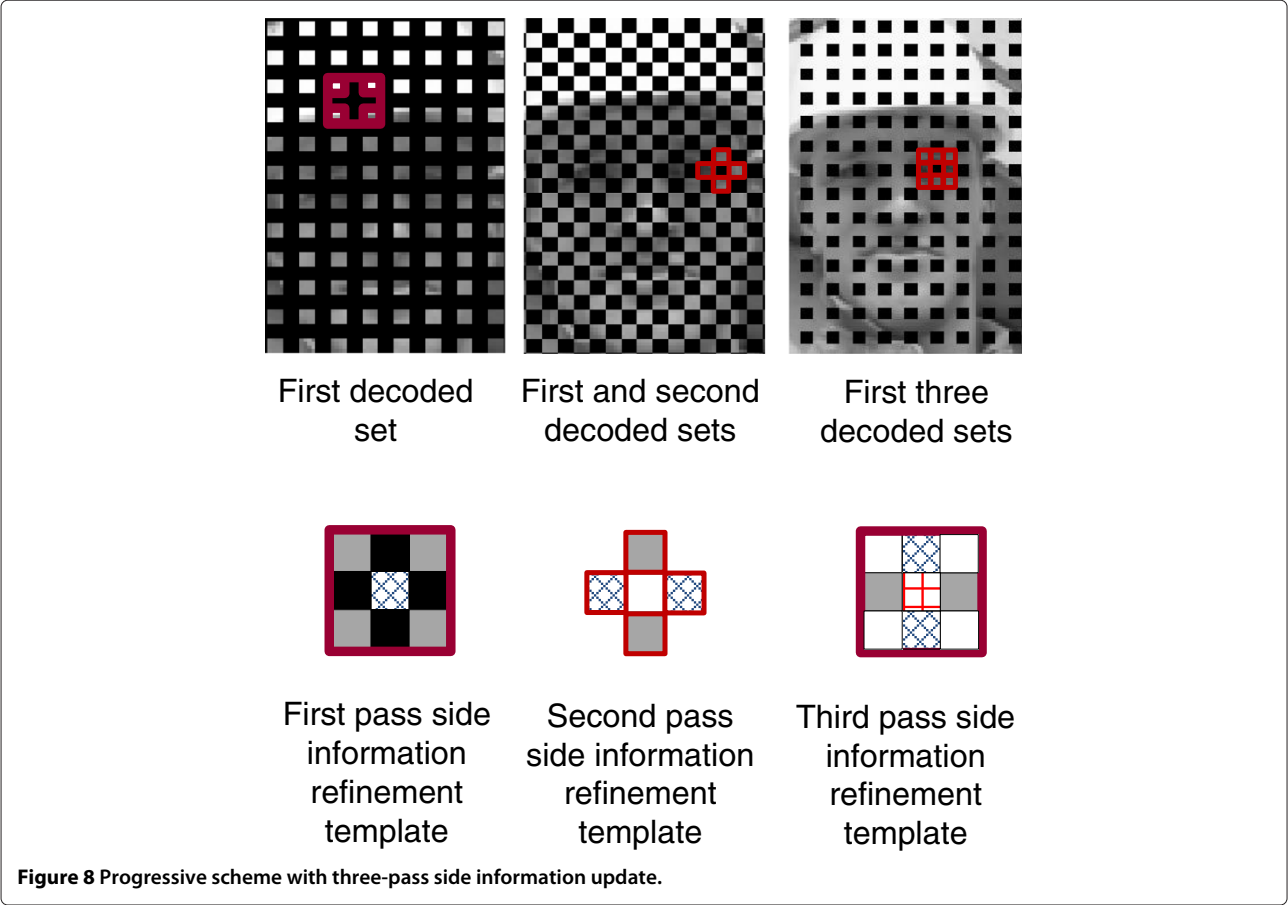


Table 1 Spatio-temporal correlation factor ρ of the refinement templates

| | 4D | EC | AN |
|--------------|--------|--------|--------|
| Foreman | 0.7775 | 0.8100 | 0.8052 |
| Coastguard | 0.8404 | 0.8582 | 0.8553 |
| Hall monitor | 0.8891 | 0.9035 | 0.9022 |
| Soccer | 0.6344 | 0.7134 | 0.6913 |
| Carphone | 0.9659 | 0.9716 | 0.9705 |
| Average | 0.8215 | 0.8513 | 0.8449 |

four diagonal blocks lead to the best matches, it means that the central block (in the previous or next frame) is more likely to be close to the original block. For the initially interpolated frame, no motion search is done and the MAD is computed using the co-located template. The computed MAD establishes the *contribution level* of the central block during motion compensation. After decoding the first and second sets of blocks, the reconstructed frame is similar to the reconstructed frame obtained after decoding the first set of the progressive scheme with two sets of blocks: the same template, consisting of an empty cross, is thus considered for the second SI refinement pass. However, the empty cross template experiences more spatial correlation with the central block than the diagonal template. Thus, the improvement of SI information is expected to be more pronounced as it will later be confirmed by simulations (see Section 5.1).

The motion estimation and compensation technique is similar to that described for the progressive schemes with two sets of blocks: it considers the weighted average of the three blocks obtained by motion search from the two neighboring frames and by *copying* the co-located block in the interpolated frame.

4.3 Progressive DVC using four sets of blocks

To further explore the performances of the progressive scheme, one more side information refinement pass is considered by splitting the Wyner-Ziv frames into four sets of blocks as shown in Figure 7. The first two passes for side information refinement are the same as in the three groups scheme.

The refinement template for the third pass considers all the neighboring blocks of the block to be updated. This template results in even more spatial correlation than the diagonal and empty cross templates. Thus, the quality of the SI relative to the fourth set of blocks is expected to be better than the other three sets. An experimental-based analysis of the SI quality for each set is given later in Section 5.1. The three side information update passes and their corresponding search templates are illustrated in Figure 8.

4.4 Spatio-temporal correlation factor of the different refinement templates

The refinement templates used along the various refinement passes are labeled as follows:

- *Empty cross template (EC)*: This template contains 4 blocks \times 16 pixels = 64 pixels. It is used in the unique refinement pass of the progressive scheme with two sets, in the second refinement pass of the progressive scheme with three sets, and in the third refinement pass of the progressive scheme with four sets.
- *Four diagonal blocks template (4D)*: This template contains 4 blocks \times 16 pixels = 64 pixels. It is used in the first refinement pass of the progressive scheme with three sets and in the first refinement pass of the progressive scheme with four sets.
- *All neighboring blocks template (AN)*: This template contains 8 blocks \times 16 pixels = 128 pixels. It is used in the third refinement pass of the progressive scheme with four sets.

The relevance of the aforementioned templates is investigated through the computation of a spatio-temporal correlation factor.

The computation of this factor is based on offline measurements with the five QCIF video sequences: Foreman, Coastguard, Hall monitor, Soccer, and Carphone (see Section 5). Each frame is divided into 4×4 pixel blocks. Then, considering a refinement template surrounding each block, B , the best match is searched in the previous and next neighboring frames. The central blocks in the previous frame, B_p , and in the

Table 2 Intra-frame quantization parameters for the tested QCIF video sequences

| Sequence | Number of frames | Intra-frame quantization | | | | | | | |
|--------------|------------------|--------------------------|-------|-------|-------|-------|-------|-------|-------|
| | | Q_1 | Q_2 | Q_3 | Q_4 | Q_5 | Q_6 | Q_7 | Q_8 |
| Foreman | 145 | 40 | 39 | 38 | 34 | 34 | 32 | 29 | 25 |
| Coastguard | 145 | 37 | 36 | 36 | 33 | 33 | 31 | 29 | 24 |
| Hall monitor | 164 | 38 | 37 | 37 | 34 | 33 | 31 | 30 | 26 |
| Soccer | 145 | 44 | 43 | 41 | 36 | 36 | 34 | 31 | 25 |
| Carphone | 177 | 40 | 39 | 38 | 34 | 34 | 32 | 29 | 25 |

Table 3 PSNR (dB) of two groups of blocks of the progressive architecture with only one refinement pass

| | Group 1 | Group 2 | $\Delta\text{PSNR}_{12}^{\text{EC}}$ |
|--------------|---------|---------|--------------------------------------|
| GOP = 2 | | | |
| Foreman | 29.558 | 33.282 | 3.724 |
| Coastguard | 31.761 | 32.858 | 1.097 |
| Hall monitor | 36.084 | 36.794 | 0.710 |
| Soccer | 21.678 | 27.51 | 5.832 |
| Carphone | 29.243 | 32.560 | 3.3170 |
| Average | 29.664 | 32.601 | 2.9370 |
| GOP = 4 | | | |
| Foreman | 27.78 | 32.399 | 4.619 |
| Coastguard | 30.144 | 31.714 | 1.570 |
| Hall monitor | 34.902 | 35.935 | 1.033 |
| Soccer | 20.509 | 26.312 | 5.803 |
| Carphone | 28.703 | 32.198 | 3.495 |
| Average | 28.407 | 31.711 | 3.304 |
| GOP = 8 | | | |
| Foreman | 26.407 | 31.332 | 4.925 |
| Coastguard | 29.082 | 30.93 | 1.848 |
| Hall monitor | 34.053 | 35.411 | 1.358 |
| Soccer | 19.908 | 26.285 | 6.377 |
| Carphone | 28.061 | 31.841 | 3.780 |
| Average | 27.502 | 31.160 | 3.658 |

$\Delta\text{PSNR}_{12}^{\text{EC}}$, PSNR difference between group 2 refined using the EC template and group 1.

next frame, B_N , are used to compute the correlation factor as follows:

$$\rho = \frac{\frac{1}{\text{MAD}_P} \frac{\text{cov}(B_P, B_P)}{\sigma_B \sigma_{B_P}} + \frac{1}{\text{MAD}_N} \frac{\text{cov}(B_P, B_N)}{\sigma_B \sigma_{B_N}}}{\frac{1}{\text{MAD}_P} + \frac{1}{\text{MAD}_N}} \quad (2)$$

where $\text{cov}(X, Y) = E[XY] - E[X]E[Y]$ is the covariance of X and Y , and $\sigma_X = \sqrt{E[X^2] - E[X]^2}$ is the standard deviation of X . $E[X]$ denotes the average or expected value of X .

Table 1 provides the computed correlation factors obtained through simulations over all the frames of the five video sequences. As expected, the EC and AN templates lead to higher correlation values than the 4D template. However, the EC template is slightly better than the AN template even if the latter contains more blocks. In fact, considering only the horizontal and vertical blocks as in the EC template is better than adding the diagonal blocks. It should be emphasized that the computations of the spatio-temporal correlation factors is conducted to assess the effectiveness of each template and that it is based on the original distortion-free frames. However,

progressive DVC applies motion refinement on reconstructed frames with some amount of distortion. Moreover, it considers the interpolated frame during motion compensation along with the neighboring frames.

5 Simulations and discussion

The proposed SIR method for progressive distributed video coding was implemented and its performances evaluated using the Discover DVC codec as a benchmark reference. The rate-distortion performances of the proposed progressive scheme with spatial correlation-based SIR is also compared with the performances obtained with the SIR method presented in [10]. To ensure fair performance comparisons, the same test conditions as those reported in [10] were applied: these are actually the same test conditions as those listed in the Discover evaluation website [31]:

- *Video sequences*: The simulations were done on the luminance component, at 15 frames per second, of the same five QCIF video sequences mentioned in Section 4.4 (Foreman, Coastguard, Hall

Table 4 PSNR (dB) of three groups of blocks of the progressive architecture with two refinement passes

| | Group 1 | Group 2 | $\Delta\text{PSNR}_{12}^{4D}$ | Group 3 | $\Delta\text{PSNR}_{13}^{\text{EC}}$ |
|--------------|---------|---------|-------------------------------|---------|--------------------------------------|
| GOP = 2 | | | | | |
| Foreman | 29.558 | 31.917 | 2.359 | 33.357 | 3.799 |
| Coastguard | 31.761 | 29.306 | -2.455 | 32.882 | 1.121 |
| Hall monitor | 36.084 | 35.82 | -0.264 | 36.792 | 0.708 |
| Soccer | 21.678 | 25.167 | 3.489 | 27.533 | 5.855 |
| Carphone | 29.353 | 31.223 | 1.870 | 32.637 | 3.284 |
| Average | 29.686 | 30.686 | 1.00 | 32.640 | 2.954 |
| GOP = 4 | | | | | |
| Foreman | 27.78 | 30.918 | 3.138 | 32.353 | 4.573 |
| Coastguard | 30.144 | 28.361 | -1.783 | 31.777 | 1.633 |
| Hall monitor | 34.902 | 34.751 | -0.151 | 35.93 | 1.028 |
| Soccer | 20.509 | 24.024 | 3.515 | 26.338 | 5.829 |
| Carphone | 28.842 | 30.833 | 1.991 | 32.262 | 3.420 |
| Average | 28.435 | 29.777 | 1.342 | 31.732 | 3.297 |
| GOP = 8 | | | | | |
| Foreman | 26.407 | 30.033 | 3.626 | 31.523 | 5.116 |
| Coastguard | 29.082 | 27.782 | -1.30 | 30.956 | 1.874 |
| Hall monitor | 34.053 | 34.106 | 0.053 | 35.398 | 1.345 |
| Soccer | 19.908 | 23.484 | 3.576 | 25.741 | 5.833 |
| Carphone | 28.202 | 30.455 | 2.253 | 31.920 | 3.718 |
| Average | 27.530 | 29.172 | 1.642 | 31.107 | 3.577 |

$\Delta\text{PSNR}_{12}^{4D}$, PSNR difference between group 2 refined using the 4D template and group 1; $\Delta\text{PSNR}_{13}^{\text{EC}}$, PSNR difference between group 3 refined using the EC template and group 1.

monitor, Soccer, and Carphone). These five video test sequences, listed in Table 2, cover a wide range of motion and texture contents.

- **Temporal correlation:** Three sizes of group of pictures are considered in the tests: $GOP = 2, 4$, and 8 , such that the efficiency of the proposed SIR scheme can be examined under different temporal correlation conditions.
- **WZ frame quantization matrix indexes (Q_i):** To investigate different codec bitrates, eight quantization 4×4 matrices for the WZ DCT bands are chosen: $Q_i = 1, \dots, 8$; that is, 1 for low bitrates and 8 for high bitrates. These matrices indicate the number of quantization levels allocated to each DCT band and can be found in [8].
- **Intra-frame quantization:** For each quantization matrix Q_i , a different quantization parameter Q_p for the H.264/AVC intra-frame is considered. Each Q_p value is selected such that the intra-frame would have a quality similar to that of the WZ frame to avoid sudden quality variations: these Q_p values are similar to those used in [8] and [11]: these values are given in Table 2 for each QCIF sequence.

- **Search area parameters:** For the proposed SIR scheme, a 4×4 pixel WZ frame block decomposition is considered. The template best match search area is 28×28 pixels. As each of the three templates (diagonal, empty cross, and all blocks templates) covers an area of 12×12 pixels, a 28×28 search area allows for a displacement of 8 pixels in each direction (right, left, up, and down).

5.1 Side information quality for the different distributed video coding schemes

Before analyzing the overall rate-distortion performances of the proposed progressive DVC architecture, the side information generation itself is examined as the interpolated frame quality has a direct impact on the achievable rate reduction and on the video sequence reconstruction quality. The effectiveness of the progressive scheme, reflected on the side information quality improvement, is reported for the three different GOP sizes implying different temporal correlation conditions. In this section, we evaluate the proposed technique's SI quality, in terms of PSNR, at the highest rate quantization point (best achievable quality) on the rate-distortion function. In other

Table 5 PSNR (dB) of four groups of blocks of the progressive architecture with three refinement passes

| | Group 1 | Group 2 | $\Delta PSNR_{12}^{4D}$ | Group 3 | $\Delta PSNR_{13}^{EC}$ | Group 4 | $\Delta PSNR_{14}^{AN}$ |
|--------------|---------|---------|-------------------------|---------|-------------------------|---------|-------------------------|
| GOP = 2 | | | | | | | |
| Foreman | 29.558 | 31.917 | 2.359 | 33.402 | 3.844 | 33.61 | 4.052 |
| Coastguard | 31.761 | 29.306 | -2.455 | 33.167 | 1.406 | 32.703 | 0.942 |
| Hall monitor | 36.084 | 35.82 | -0.264 | 36.976 | 0.892 | 36.724 | 0.640 |
| Soccer | 21.678 | 25.167 | 3.489 | 27.517 | 5.839 | 27.782 | 6.104 |
| Carphone | 29.353 | 31.223 | 1.870 | 32.476 | 3.123 | 33.079 | 3.726 |
| Average | 29.686 | 30.686 | 1.00 | 32.707 | 3.021 | 32.779 | 3.093 |
| GOP = 4 | | | | | | | |
| Foreman | 27.78 | 30.918 | 3.138 | 32.399 | 4.619 | 32.695 | 4.915 |
| Coastguard | 30.144 | 28.361 | -1.783 | 32.12 | 1.976 | 31.586 | 1.442 |
| Hall monitor | 34.902 | 34.751 | -0.151 | 36.055 | 1.153 | 35.93 | 1.028 |
| Soccer | 20.509 | 24.024 | 3.515 | 26.34 | 5.831 | 26.547 | 6.038 |
| Carphone | 28.837 | 30.838 | 2.001 | 32.103 | 3.266 | 32.733 | 3.896 |
| Average | 28.434 | 29.778 | 1.3440 | 31.803 | 3.3690 | 31.898 | 3.464 |
| GOP = 8 | | | | | | | |
| Foreman | 26.407 | 30.033 | 3.626 | 31.551 | 5.144 | 31.921 | 5.514 |
| Coastguard | 29.082 | 27.782 | -1.30 | 31.278 | 2.196 | 30.78 | 1.698 |
| Hall monitor | 34.053 | 34.106 | 0.053 | 35.204 | 1.151 | 35.15 | 1.097 |
| Soccer | 19.908 | 23.4844 | 3.5764 | 25.726 | 5.818 | 25.964 | 6.056 |
| Carphone | 28.203 | 30.449 | 2.246 | 31.8 | 3.597 | 32.375 | 4.172 |
| Average | 27.531 | 29.170 | 1.639 | 31.112 | 3.581 | 31.238 | 3.707 |

$\Delta PSNR_{12}^{4D}$, PSNR difference between group 2 refined using the 4D template and group 1; $\Delta PSNR_{13}^{EC}$, PSNR difference between group 3 refined using the EC template and group 1; $\Delta PSNR_{14}^{AN}$, PSNR difference between group 4 refined using the AN template and group 1.

words, the key frames are encoded for the highest quality, i.e., with the lowest quantization parameter Q_p .

Tables 3, 4, and 5 provide the PSNR values of the side information of each set of blocks (or group of blocks) for the progressive scheme with one, two, and three refinement passes. The PSNR of each group is the average PSNR of that group over all the WZ frames in the video sequence. More precisely, the PSNR of group 1 is computed between the blocks belonging to that group in the BiMESS-generated SI and the corresponding blocks in the original frame. Similarly, the PSNR of group 2 is computed between the blocks belonging to that group in the refined SI and the original frame and so on.

A comparison of each of the groups before and after the refinement could have also been done. However, this is practically the same as comparing the SI quality between each group. In fact, the various groups of blocks represent almost the same content since they are uniformly dispersed over the WZ frame. These tables also give the

PSNR average over all the WZ frames of the five video sequences to provide an overall statistical comparison of the side information quality for the aggregated video sequences. The SI refinement improvement is assessed through the PSNR difference, $\Delta\text{PSNR}_{1g}^{\text{template}}$, between the first group, $g = 1$ (generated using the conventional BiMESS method without any refinement) and the second, third, and fourth groups, $g = 2, 3$, or 4, after refinement using one of the aforementioned templates (EC, 4D, or AN). The PSNR differences in Tables 3, 4, and 5 are emphasized.

5.1.1 Progressive architecture with two groups of blocks

To assess the side information quality improvement during the different refinement passes, Table 3 reports the PSNR values of the interpolated frames for the two sets for GOP values of 2, 4, and 8. It shows the improvement of the side information quality of the second set (refined using the EC template) over the first set (without refinement),

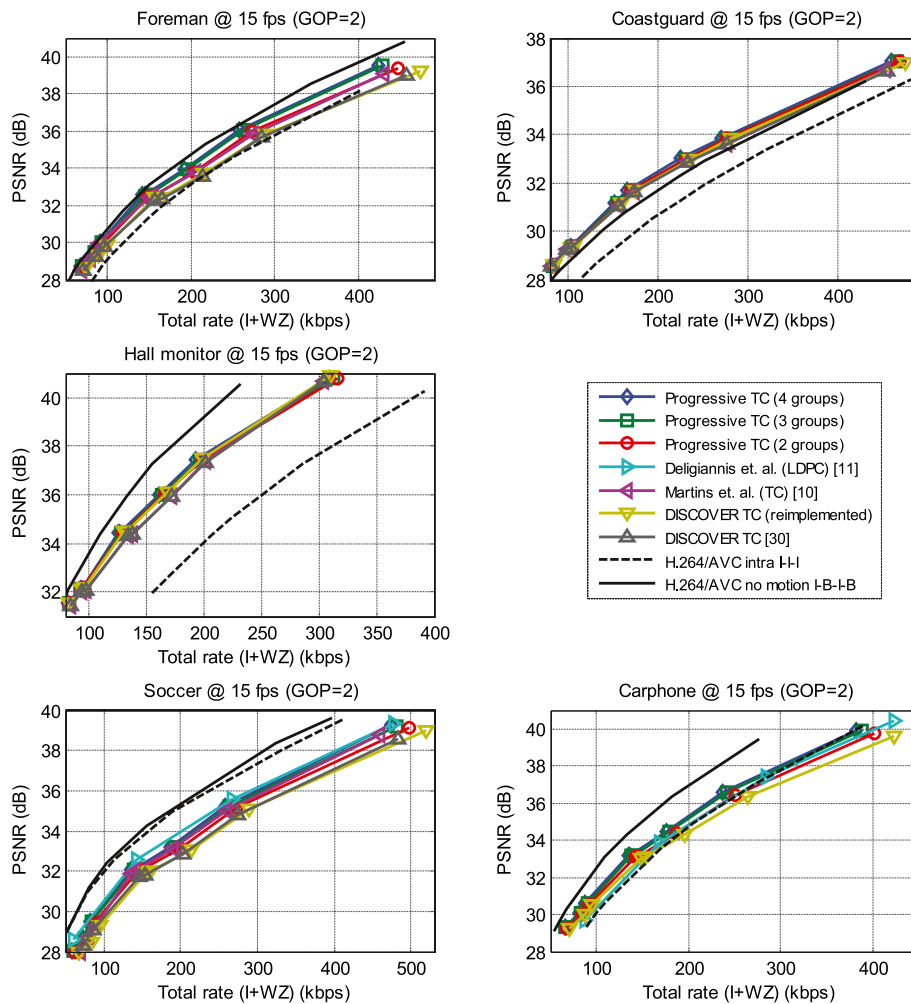


Figure 9 Progressive DVC architecture performances for GOP = 2.

obtained by averaging the PSNR over the whole sequence of frames.

Furthermore, Table 3 shows that the PSNR improvement of the progressive scheme is more important when the temporal correlation decreases, or as the GOP size increases. The improvement obtained by motion refinement, by exploiting also the spatial correlation, is more pronounced for low temporal correlation. When the GOP size increases, the MCTI estimates the motion between two temporally distant key frames. The actual motion is not easy to track in practice and the progressive scheme is thus particularly efficient for that purpose.

5.1.2 Progressive architecture with three groups of blocks

Interpolation performances of the progressive scheme with two refinement passes are reported in Table 4. The first refinement pass is based on the 4D refinement template. This template is slightly less spatially correlated with the central block than the EC template used in the second

refinement pass. Therefore, the PSNR results for the second refinement pass (third group) are higher, on average, than those obtained with the first pass (second group). For slow-motion sequences, however, the 4D refinement template is not precise enough to increase the accuracy of motion estimation. In other words, the motion estimation is already working well and the motion refinement with an inaccurate template is more likely to capture noise than true motion. For fast-motion sequences, the temporal interpolation is highly inaccurate, and because of this, even a coarse refinement template is able to enhance the SI quality. Indeed, the first refinement pass using the 4D template gives a significant improvement for the Foreman and Soccer video sequences.

5.1.3 Progressive architecture with four groups of blocks

The PSNR values of the four groups of blocks are shown in Table 5 for the progressive SIR scheme using three refinement passes. The first two refinement passes give the

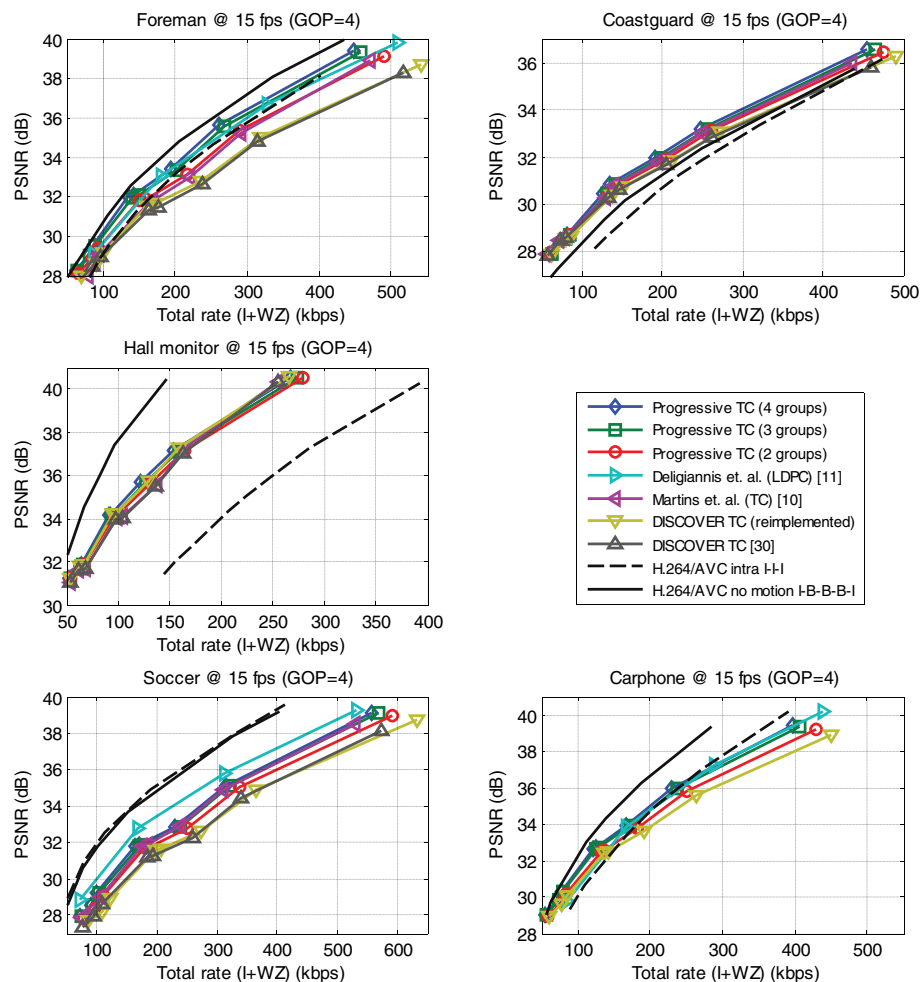


Figure 10 Progressive DVC architecture performances for GOP = 4.

same results as those achieved with the previous progressive scheme with three groups. As the third refinement pass uses a template composed of all the neighboring blocks, it is expected to lead to better motion refinements, especially for fast-motion video sequences. However, for slow-motion videos, the third refinement does not give interpolation improvement.

To summarize, it is observed from Tables 3, 4, and 5 that the improvement obtained with SIR is more significant as the GOP increases and for fast-motion video sequences. By means of progressive distributed video coding, the lack of temporal correlation is compensated by the spatial correlation during the SI generation.

5.2 Rate-distortion performances of progressive architectures

For the rate-distortion analysis, the Discover DVC codec with turbo coding was *reimplemented* according to the simulation conditions specified in [31] and recalled at the

beginning of this section. It is verified that the performance results for the reimplemented benchmark Discover codec are similar to those reported for the turbo code-based Discover architecture in [31] (see Figures 9, 10, and 11). The Discover performances are computed using the software downloaded from the Discover website [31].

The proposed progressive mechanism is incorporated over the reimplemented Discover codec. A summary of the interpolated frame quality obtained with the different DVC schemes is given in Table 6. The first scheme, identified as BiMESS, refers to the BiMESS interpolation used by Discover. The subsequent schemes are the proposed progressive DVC schemes with the three proposed block arrangements. The average PSNR over all interpolated frames with its different groups are evaluated. On the average, the interpolation quality increases as more refinement passes are performed, except for the case of slower motion video sequences (i.e., Hall monitor). The impact of the interpolation quality on the overall

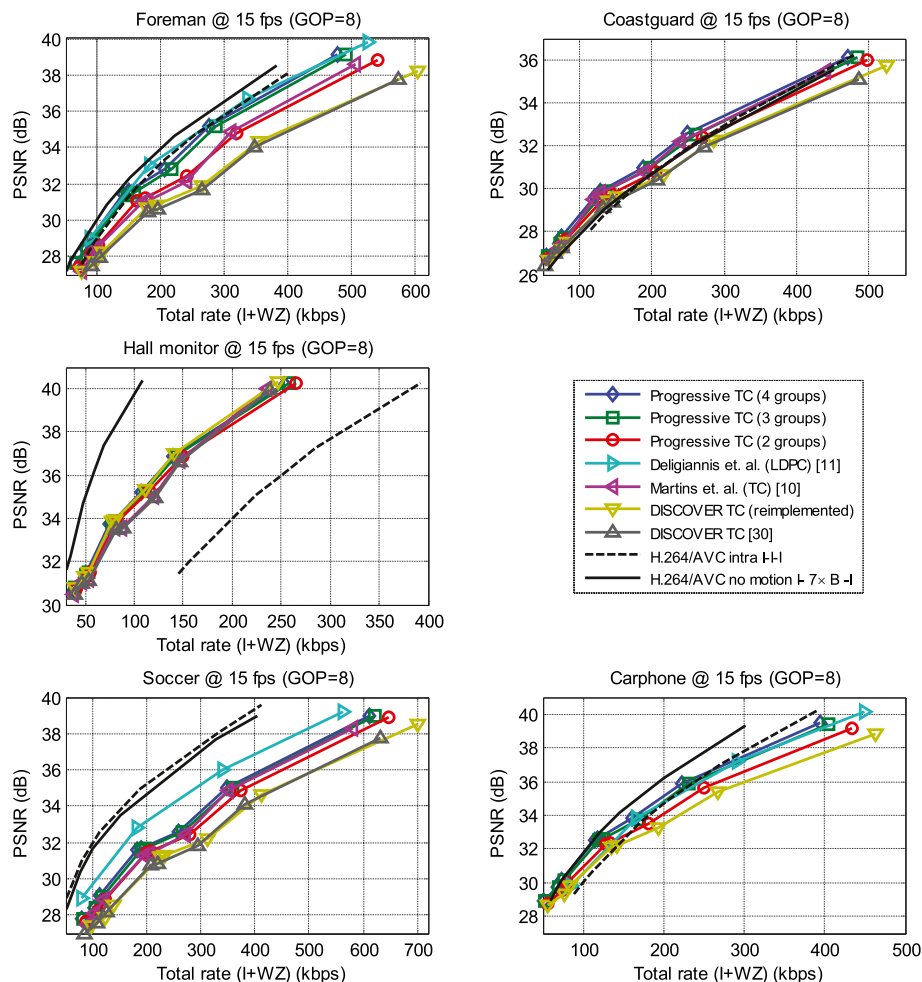


Figure 11 Progressive DVC architecture performances for GOP = 8.

PSNR as a function of the bitrate is demonstrated in the RD curves of Figures 9, 10, and 11. The different DVC and conventional video schemes presented in these figures do not perform any motion estimation at the encoder side:

- *H264/AVC no motion I-(GOP-1)B-I*: It exploits the spatial redundancy without any motion estimation.
- *H264/AVC intra I-I-I*: Each frame is encoded independently from the neighboring frames without exploiting the spatial redundancy.
- *Discover (Turbo Code) [31]*: The values are taken from the Discover website [31] and for which the Slepian-Wolf codec is based on turbo coding.
- *Discover (reimplemented)*: Reimplementation of the Discover benchmark codec based on turbo coding and considering the same simulation conditions as in [31].
- *Martins et al. [10]*: The SIR scheme proposed by Martins et al. [10]. The WZ part of this codec is based on turbo code (TC).
- *Deligiannis et al. [11]*: The SIR scheme proposed by Deligiannis et al. [11]. The WZ part of this codec is based on low-density parity-check code (LDPC).

Note that the LDPC-based SW decoder gives better compression than the TC-based SW decoder. For this reason, the Bjøntegaard metrics of this scheme is computed by considering the LDPC-based DISCOVER codec.

- *Proposed (two, three, or four groups)*: The proposed progressive scheme with two, three, and four groups of blocks.

The improvement obtained by the progressive scheme is intimately related to the interpolation quality. For larger groups of pictures, the progressive refinement improvement is more significant: the lack of temporal correlation is mitigated by exploiting the spatial correlation during the side information generation. The SIR based on progressive schemes has proven more effective for the rapid sequences, showing an improvement of up to 3 dB, for instance, for the Foreman sequence with $GOP=8$ and the progressive scheme with 4 groups. For the slow-motion sequences, such as the Hall monitor video sequence, however, the progressive scheme does not bring noticeable performance improvements even when the GOP size increases.

Table 6 Average PSNR (dB) of the interpolated frame over all the WZ frames

| | BiMESS | Progressive DVC (2 groups) | Progressive DVC (3 groups) | Progressive DVC (4 groups) |
|--------------|--------|-------------------------------|-------------------------------|-------------------------------|
| GOP = 2 | | | | |
| Foreman | 29.549 | 31.393 | 32.047 | 32.122 |
| Coastguard | 31.675 | 32.315 | 31.708 | 31.734 |
| Hall Monitor | 36.138 | 36.455 | 36.372 | 36.401 |
| Soccer | 21.698 | 24.592 | 25.478 | 25.536 |
| Carphone | 29.271 | 30.90 | 31.46 | 31.53 |
| Average | 29.666 | 31.131 | 31.413 | 31.464 |
| GOP = 4 | | | | |
| Foreman | 27.721 | 30.063 | 30.851 | 30.95 |
| Coastguard | 30.136 | 30.927 | 30.515 | 30.555 |
| Hall Monitor | 35.007 | 35.42 | 35.194 | 35.41 |
| Soccer | 20.501 | 23.405 | 24.302 | 24.353 |
| Carphone | 28.716 | 30.451 | 31.049 | 31.128 |
| Average | 28.416 | 30.053 | 30.382 | 30.479 |
| GOP = 8 | | | | |
| Foreman | 26.37 | 28.674 | 29.499 | 29.981 |
| Coastguard | 29.038 | 30.004 | 28.901 | 29.731 |
| Hall Monitor | 34.227 | 34.729 | 34.739 | 34.484 |
| Soccer | 19.896 | 23.057 | 23.154 | 23.778 |
| Carphone | 28.068 | 29.951 | 30.624 | 30.707 |
| Average | 27.519 | 29.283 | 29.383 | 29.736 |

The average over the five video sequences is emphasized.

From Table 6, one can verify that the quality of the interpolated frames is slightly improved by the progressive scheme for the Hall monitor sequence. However, this slight interpolation improvement does not lead to an improvement in the rate-distortion curves in Figures 9, 10, and 11. This can be explained from a channel coding point of view: when the frame is subdivided into more subsets, the length of the turbo code and its interleaver is smaller. Therefore, the error-correcting capability of the turbo code decreases.

The PSNR of the Coastguard sequence interpolated frames obtained by the progressive scheme with three and four sets is lower than that of the interpolated frame with the progressive scheme with two sets. This is due to the refinement pass using the coarse 4D template. However, the RD performances of the progressive schemes with three and four sets are better than that with two sets. Recall that the Coastguard sequence contains a 'moving boat' and a large texture area of 'water' in the background which is usually hard to refine since the template is vulnerable to the noise. Nevertheless, around the moving boat, there is some structure that can help the

refinement process to eliminate the motion estimation errors.

The progressive scheme leads to an improvement over the SIR technique of Martins et al. [10] by up to 1.2 dB (Foreman with GOP 8) since the motion refinement is processed using completely reconstructed templates containing the various DCT components and not only a subset of the DCT bands. The concept of the progressive scheme was inspired from the principle of the intra-frame *differential pulse code modulation* (DPCM) coding. However, since the WZ codec requires long codes, it is not possible to apply linear prediction, pixel by pixel, using the neighboring spatially correlated pixels. Thus, the frame decomposition in the progressive scheme should provide spatially correlated groups. Unlike DPCM, where the spatial correlation is used to predict the upcoming pixel, for the proposed progressive DVC scheme, the spatial correlation is used to rectify the motion field, thus leading to a better exploitation of the temporal correlation properties across the video sequence.

In Table 7 the overall RD performances of the different SIR schemes are evaluated using the Bjøntegaard

Table 7 Bjøntegaard Deltas of the different refined DVC codecs compared to the DISCOVER DVC system

| Sequence | SIR technique | GOP 2 | | GOP 4 | | GOP 8 | |
|--------------|-------------------------|----------------|--------------------|----------------|--------------------|----------------|--------------------|
| | | ΔR (%) | $\Delta PSNR$ (dB) | ΔR (%) | $\Delta PSNR$ (dB) | ΔR (%) | $\Delta PSNR$ (dB) |
| Foreman | Progressive with 4 sets | -13.1935 | 0.8249 | -24.0237 | 1.4995 | -31.3859 | 2.0677 |
| | Progressive with 3 sets | -11.3440 | 0.6997 | -21.0332 | 1.2848 | -28.0933 | 1.8069 |
| | Progressive with 2 sets | -6.9061 | 0.4149 | -12.7094 | 0.7371 | -15.6575 | 0.9305 |
| | Deligiannis et al. | -4.7693 | 0.3162 | -20.8418 | 1.3473 | -34.1192 | 2.3699 |
| | Martins et al. | -5.6930 | 0.3644 | -7.1264 | 0.4419 | -14.3356 | 0.8285 |
| Soccer | Progressive with 4 sets | -14.1737 | 0.8376 | -19.2194 | 1.2067 | -22.0967 | 1.4351 |
| | Progressive with 3 sets | -11.9026 | 0.6945 | -16.4910 | 1.0198 | -19.3024 | 1.2328 |
| | Progressive with 2 sets | -7.4463 | 0.4279 | -10.5923 | 0.6393 | -13.9559 | 0.8687 |
| | Deligiannis et al. | -17.1781 | 0.9931 | -30.2909 | 1.9246 | -38.9008 | 2.6557 |
| | Martins et al. | -9.0854 | 0.5333 | -15.2502 | 0.9139 | -15.8318 | 0.9906 |
| Coastguard | Progressive with 4 sets | -4.3803 | 0.2189 | -10.8566 | 0.4840 | -17.5782 | 0.8257 |
| | Progressive with 3 sets | -2.5786 | 0.1271 | -7.5031 | 0.3267 | -13.4620 | 0.6154 |
| | Progressive with 2 sets | -2.1362 | 0.1060 | -4.5434 | 0.1947 | -7.8475 | 0.3476 |
| | Martins et al. | 0.3071 | -0.0067 | -7.3185 | 0.2890 | -13.2512 | 0.6355 |
| Hall monitor | Progressive with 4 sets | -1.9400 | 0.1437 | -2.1744 | 0.1359 | 0.6280 | -0.0233 |
| | Progressive with 3 sets | -0.3869 | 0.0311 | 1.1726 | -0.0605 | 2.7001 | -0.1306 |
| | Progressive with 2 sets | 0.6052 | -0.0383 | 3.8789 | -0.2132 | 6.6494 | -0.3268 |
| | Martins et al. | 3.4073 | -0.2491 | 5.5937 | -0.3381 | 9.5248 | -0.4840 |
| Carphone | Progressive with 4 sets | -11.9046 | 0.7508 | -16.6542 | 0.9418 | -22.6768 | 1.2945 |
| | Progressive with 3 sets | -10.0661 | 0.6258 | -13.7520 | 0.7611 | -18.9992 | 1.0578 |
| | Progressive with 2 sets | -5.8555 | 0.3546 | -6.6457 | 0.3519 | -9.4964 | 0.4989 |
| | Deligiannis et al. | -2.2857 | 0.1941 | -6.3553 | 0.4628 | -16.3032 | 1.0443 |

For each scenario, the best SIR technique Bjøntegaard Deltas are emphasized.

Delta PSNR and bitrate metrics [12]. These metrics are computed according to the SI non-refined system DISCOVER. All the refined DVC architectures considered here use turbo coding in the WZ part, and they are consequently compared to the DISCOVER system using turbo coding, except for the refined technique of Deligiannis [11] which uses LDPC and is then compared to DISCOVER system using LDPC. Using the Bjøntegaard measure [12], it is observed that the proposed progressive codec also achieves compression gains of up to 31.3859% rate reduction when compared to DISCOVER. Moreover, the progressive scheme with four groups exceeds the performances of the SIR technique of Martins [10] for all the different setups. By comparison with the Deligiannis [11] SI refinement technique, the progressive scheme with four groups gives better results for the majority of scenarios, that is except for the Soccer QCIF sequence.

5.3 Complexity analysis

In this section, the encoding and decoding complexity of the proposed progressive scheme with four sets is investigated. The complexity is assessed by measuring the execution time required by the encoder and by the decoder using a personal computer with an Intel® Core™ i7 CPU processor at 2.67 GHz with 12 GB of RAM.

Table 8 gives the encoding execution time in seconds for three video schemes with all the frames (see Table 2) of

the Foreman and Soccer sequences and with GOP = 2 and 8:

1. Conventional H264/AVC intra-standard ($T_{\text{Intra}}^{\text{enc}}$). Four quantization parameter (QP) values are considered. The same encoder generates the key frames used by the DISCOVER and progressive DVC systems.
2. The state-of-the-art DISCOVER encoder ($T_{\text{DISC}}^{\text{enc}}$). Four quantization index (Q_i) values are considered. The execution time is the sum of the key frame encoding time and the WZ frame encoding time:
 $T_{\text{DISC}}^{\text{enc}} = T^{\text{Key}} + T_{\text{DISC}}^{\text{WZ}}$.
3. The proposed progressive DVC encoder with four sets. $T_{\text{Prog4}}^{\text{enc}} = T^{\text{Key}} + T_{\text{Prog4}}^{\text{WZ}}$.

The $\frac{T_{\text{Prog4}}^{\text{enc}}}{T_{\text{DISC}}^{\text{enc}}}$ ratio indicates that the progressive encoder complexity exceeds the DISCOVER encoder complexity by about 12% for GOP = 2 and about 57% for GOP = 8. The additional complexity is mainly due to the frame splitting into four parts and the execution of four times the encoding process of the two RSC encoders forming the turbo encoder. Despite this encoding complexity increase, the proposed progressive scheme still follows the DVC paradigm purpose as it reduces significantly the encoding complexity when compared to that of the H264/AVC

Table 8 Comparison of the encoding execution times in seconds

| | H264/AVC intra | | Q_i | T^{Key} | DISCOVER | | Progressive with 4 sets | | Ratios | |
|---------|----------------|---------------------------------|-------|------------------|-------------------------------|--------------------------------|--------------------------------|---------------------------------|--|---|
| | QP | $T_{\text{Intra}}^{\text{enc}}$ | | | $T_{\text{DISC}}^{\text{WZ}}$ | $T_{\text{DISC}}^{\text{enc}}$ | $T_{\text{Prog4}}^{\text{WZ}}$ | $T_{\text{Prog4}}^{\text{enc}}$ | $\frac{T_{\text{Prog4}}^{\text{enc}}}{T_{\text{DISC}}^{\text{enc}}}$ | $\frac{T_{\text{Prog4}}^{\text{enc}}}{T_{\text{Intra}}^{\text{enc}}}$ |
| Foreman | 40 | 27.941 | 1 | 14.659 | 1.14 | 15.79 | 2.36 | 17.02 | 1.08 | 0.61 |
| GOP 2 | 34 | 29.630 | 5 | 15.551 | 1.59 | 17.14 | 3.47 | 19.02 | 1.11 | 0.64 |
| | 29 | 31.263 | 7 | 16.451 | 1.81 | 18.26 | 4.10 | 20.55 | 1.12 | 0.62 |
| | 25 | 33.291 | 8 | 17.491 | 2.01 | 19.50 | 4.39 | 21.88 | 1.12 | 0.65 |
| Foreman | 40 | 27.941 | 1 | 3.371 | 2.10 | 5.47 | 5.07 | 8.44 | 1.54 | 0.30 |
| GOP 8 | 34 | 29.630 | 5 | 3.581 | 2.82 | 6.40 | 6.14 | 9.72 | 1.52 | 0.33 |
| | 29 | 31.263 | 7 | 3.817 | 3.01 | 6.83 | 6.47 | 10.29 | 1.51 | 0.33 |
| | 25 | 33.291 | 8 | 4.105 | 3.52 | 7.62 | 8.03 | 12.13 | 1.59 | 0.36 |
| Soccer | 44 | 27.971 | 1 | 14.281 | 1.46 | 15.74 | 3.31 | 17.59 | 1.12 | 0.63 |
| GOP 2 | 36 | 29.563 | 5 | 15.211 | 1.75 | 16.96 | 3.82 | 19.03 | 1.12 | 0.64 |
| | 31 | 31.251 | 7 | 16.121 | 1.90 | 18.02 | 4.31 | 20.43 | 1.13 | 0.65 |
| | 25 | 33.270 | 8 | 17.513 | 2.03 | 19.54 | 4.81 | 22.32 | 1.14 | 0.67 |
| Soccer | 44 | 27.971 | 1 | 3.271 | 2.16 | 5.43 | 4.96 | 8.23 | 1.51 | 0.29 |
| GOP 8 | 36 | 29.563 | 5 | 3.471 | 3.08 | 6.55 | 6.84 | 10.31 | 1.57 | 0.35 |
| | 31 | 31.251 | 7 | 3.731 | 3.29 | 7.02 | 7.29 | 11.02 | 1.57 | 0.35 |
| | 25 | 33.270 | 8 | 4.091 | 3.61 | 7.70 | 8.30 | 12.391 | 1.61 | 0.37 |

$T_{\text{Prog4}}^{\text{enc}}$, encoding execution time of the proposed progressive scheme with four sets; $T_{\text{DISC}}^{\text{enc}}$, encoding execution time of DISCOVER; $T_{\text{Intra}}^{\text{enc}}$, encoding execution time of H264/AVC intra. $T_{\text{Prog4}}^{\text{enc}}$, $T_{\text{DISC}}^{\text{enc}}$, and the time encoding ratios are emphasized.

Table 9 Comparison of the decoding execution times in seconds

| | | DISCOVER | | | Progressive with 4 sets | | | Ratios | | |
|---------|-------|--------------------------------|-------------------------------|--------------------------------|---------------------------------|--------------------------------|---------------------------------|--|--|--|
| | Q_i | $T_{\text{DISC}}^{\text{SIG}}$ | $T_{\text{DISC}}^{\text{SW}}$ | $T_{\text{DISC}}^{\text{dec}}$ | $T_{\text{Prog4}}^{\text{SIG}}$ | $T_{\text{Prog4}}^{\text{SW}}$ | $T_{\text{Prog4}}^{\text{dec}}$ | $\frac{T_{\text{Prog4}}^{\text{SIG}}}{T_{\text{DISC}}^{\text{SIG}}}$ | $\frac{T_{\text{Prog4}}^{\text{SW}}}{T_{\text{DISC}}^{\text{SW}}}$ | $\frac{T_{\text{Prog4}}^{\text{dec}}}{T_{\text{DISC}}^{\text{dec}}}$ |
| Foreman | 1 | 53.54 | 339.0 | 392.6 | 217.79 | 403.1 | 620.9 | 4.07 | 1.19 | 1.58 |
| GOP 2 | 5 | 53.06 | 849.0 | 902.1 | 217.94 | 848.2 | 1,066.2 | 4.11 | 0.99 | 1.18 |
| | 7 | 53.02 | 1,702.0 | 1,755.1 | 217.92 | 1,595.5 | 1,813.4 | 4.11 | 0.94 | 1.03 |
| | 8 | 53.15 | 2,858.6 | 2,911.7 | 217.03 | 2,681.0 | 2,898.0 | 4.11 | 0.94 | 0.99 |
| Foreman | 1 | 88.73 | 829.4 | 918.1 | 358.68 | 977.1 | 1,335.8 | 4.04 | 1.18 | 1.45 |
| GOP 8 | 5 | 88.65 | 2,075.1 | 2,163.8 | 358.75 | 2,042.5 | 2,401.3 | 4.05 | 0.98 | 1.11 |
| | 7 | 88.67 | 4,115.5 | 4,204.2 | 360.60 | 3,617.1 | 3,977.7 | 4.06 | 0.89 | 0.94 |
| | 8 | 88.73 | 6,696.9 | 6,785.6 | 358.93 | 5,633.0 | 5,991.9 | 4.04 | 0.84 | 0.88 |
| Soccer | 1 | 53.79 | 490.38 | 544.17 | 218.18 | 532.59 | 750.77 | 4.05 | 1.09 | 1.38 |
| GOP 2 | 5 | 53.08 | 1,108.59 | 1,161.67 | 217.54 | 1,165.73 | 1,383.27 | 4.10 | 1.05 | 1.19 |
| | 7 | 53.64 | 2,126.88 | 2,180.52 | 217.23 | 1,952.67 | 2,169.90 | 4.05 | 0.92 | 0.99 |
| | 8 | 53.24 | 3,357.65 | 3,410.89 | 217.39 | 2,945.31 | 3,162.70 | 4.08 | 0.88 | 0.93 |
| Soccer | 1 | 89.84 | 945.46 | 1,035.30 | 364.35 | 989.26 | 1,353.61 | 4.05 | 1.05 | 1.31 |
| GOP 8 | 5 | 89.86 | 2,172.66 | 2,262.52 | 362.52 | 2,014.83 | 2,377.35 | 4.03 | 0.93 | 1.05 |
| | 7 | 89.17 | 4,330.37 | 4,419.54 | 364.02 | 3,756.27 | 4,120.29 | 4.08 | 0.87 | 0.93 |
| | 8 | 89.76 | 6,972.57 | 7,062.33 | 365.24 | 5,102.67 | 5,467.91 | 4.07 | 0.73 | 0.77 |

$T_{\text{Prog4}}^{\text{dec}} = T_{\text{Prog4}}^{\text{SIG}} + T_{\text{Prog4}}^{\text{SW}}$, decoding execution time of the proposed progressive scheme with four sets; $T_{\text{DISC}}^{\text{dec}} = T_{\text{DISC}}^{\text{SIG}} + T_{\text{DISC}}^{\text{SW}}$, decoding execution time of DISCOVER.

$T_{\text{DISC}}^{\text{dec}}$, $T_{\text{Prog4}}^{\text{dec}}$, and $\frac{T_{\text{Prog4}}^{\text{dec}}}{T_{\text{DISC}}^{\text{dec}}}$ are emphasized.

intra-coding: the ratios $\frac{T_{\text{Prog4}}^{\text{enc}}}{T_{\text{Intra}}^{\text{enc}}}$ in Table 8 indicate a complexity reduction of almost 40% and 70% for GOP = 2 and 8, respectively.

As for the decoding complexity itself, Table 9 provides the decoding execution times for both progressive and DISCOVER schemes. The overall decoding time involves the side information generation execution time, T_{SIG} , and the Slepian-Wolf decoder execution time, T_{SW} (turbo decoding): $T^{\text{dec}} = T^{\text{SIG}} + T^{\text{SW}}$. This table shows that the additional computational complexity of the proposed decoder, due to multiple side information refinement passes, is compensated by a faster turbo decoding. For instance, for the progressive scheme with four groups of blocks, the 1,584-long bit-plane is split into four parts leading to four 396-long bitplanes: the turbo decoding process of the 1,584-long bitplane is more time-consuming than the turbo decoding of the four 396-long bitplanes, leading to comparable decoding execution times, $T_{\text{DISC}}^{\text{dec}}$ and $T_{\text{Prog4}}^{\text{dec}}$, for DISCOVER and the progressive scheme with four correlated sets.

6 Conclusions

A new distributed video coding scheme is presented in this paper. This DVC scheme, based on progressive coding, consists of splitting the Wyner-Ziv frames into spatially correlated sets. These sets are then sent and decoded

progressively to aid the motion-compensated temporal interpolation process for the decoding of the subsequent sets through motion refinement passes. This method complies perfectly with the distributed paradigm and does not involve additional decoding complexity. Moreover, the refinement passes are based on carefully chosen templates which are able to improve progressively the estimation of the motion vectors. To demonstrate the feasibility of the proposed method, the progressive scheme is incorporated over the Discover DVC codec. Significant improvements have been obtained, particularly for fast video sequences and larger group of pictures. The interpolation quality is greatly enhanced and an improvement of up to 3 dB is reported for the overall rate-distortion (PSNR versus bitrate) performances, and this does not have any significant impact on the computational complexity of the encoders and decoders.

Competing interests

The authors declare that they have no competing interests.

Acknowledgements

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the quality of the paper. They are also grateful to the authors of [11] and in particular to Dr N. Deligiannis for sharing their experimental results.

This research program was scientifically and financially supported by a research collaboration project and an academic grant from Communications Research Centre Canada (CRC). The research project was also financially supported by a Natural Sciences and Engineering Research Council of Canada (NSERC) Alexander-Graham Bell Graduate Scholarship.

Author details

¹Laval University, Quebec, Quebec, G1V 0A8, Canada. ²Communications Research Centre Canada, Ottawa, Ontario, K2H 8S2, Canada.

Received: 9 April 2013 Accepted: 23 October 2013

Published: 5 November 2013

References

1. J Slepian, J Wolf, Noiseless coding of correlated information sources. *IEEE Trans. Inf. Theory*. **19**(4), 471–480 (1973)
2. AD Wyner, J Ziv, The rate-distortion function for source coding with side information at the decoder. *Trans. Inf. Theory*. **IT-22**(1), 1–10 (1976)
3. R Puri, K Ramchandran, Prism: a new robust video coding architecture based on distributed compression principles, in *Proceedings of the Allerton Conference on Communication Control and Computing*, (Urbana-Champaign, 2–4 Oct 2002)
4. R Puri, A Majumdar, K Ramchandran, Prism: a video coding paradigm with motion estimation at the decoder. *IEEE Trans. Image Process.* **16**, 2436–2448 (2007)
5. A Aaron, R Zhang, B Girod, Wyner-Ziv coding of motion video, in *Asilomar Conference on Signals, Systems and Computers*, (Pacific Grove, 3–6 Nov 2002)
6. B Girod, A Aaron, S Rane, DR Monedero, Distributed video coding. *Proc. IEEE, Special Issue on Advances in Video Coding and Delivery*. **93**, 71–83 (2005)
7. F Pereira, C Brites, J Ascenso, M Tagliasacchi, Wyner-Ziv video coding: a review of the early architectures and further developments, in *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME '08)*, (Hannover, 23 June–26 April 2008), pp. 625–628
8. X Artigas, J Ascenso, M Dalai, S Klomp, D Kubasov, M Ouaret, The DISCOVER codec: architecture, techniques and evaluation, in *Proceedings of the Picture Coding Symposium*, (Lisbon, 7–9 Nov 2007)
9. MH Taieb, J-Y Chouinard, D Wang, Loukhaoukha K, Progressive coding and side information updating for distributed video coding. *J. Inf. Hiding Multimedia Signal Process.* **3**, 1–11 (2011)
10. R Martins, C Brites, J Ascenso, F Pereira, Refining side information for improved transform domain Wyner-Ziv video coding. *IEEE Trans. Circuits Syst. Video Technol.* **19**, 1327–1341 (2009)
11. N Deligiannis, F Verbist, J Slowack, RVD Walle, P Schelkens, A Munteanu, Progressively refined Wyner-Ziv video coding for visual sensors. *ACM Trans. Sensor Netw., Special Issue New Advancements Distributed Smart Camera Netw.* (2014, in press)
12. G Bjøntegaard, Calculation of average PSNR differences between RD-curves, in *Proceedings of the ITU-T Video Coding Experts Group (VCEG) Thirteenth Meeting*, (Austin, 2–4 Apr 2001)
13. D Kubasov, K Lajnef, C Guillemot, A hybrid encoder/decoder rate control for Wyner-Ziv video coding with a feedback channel, in *Proceedings of the International Workshop on Multimedia Signal Processing*, (Crete, 1–3 Oct 2007)
14. J Ascenso, C Brites, F Pereira, Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding, in *Proceedings of the 5th EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services*, (Smolenice, 29 June–2 July 2005)
15. J Ascenso, C Brites, F Pereira, Content adaptive Wyner-Ziv video coding driven by motion activity, in *Proceedings of the IEEE International Conference on Image Processing*, (Atlanta, 8–11 Oct 2006), pp. 605–608
16. S Klomp, Y Vatis, J Ostermann, Side information interpolation with sub-pel motion compensation for Wyner-Ziv decoder, in *Proceedings of the International Conference on Signal Processing and Multimedia Applications (SIGMAP)*, (Setúbal, 7–10 Aug 2006)
17. C Brites, F Pereira, Correlation noise modeling for efficient pixel and transform domain Wyner-Ziv video coding. *IEEE Trans. Circuits Syst. Video Technol.* **18**, 1177–1190 (2008)
18. S Wang, L Cui, L Stankovic, V Stankovic, S Cheng, Adaptive correlation estimation with particle filtering for distributed video coding. *IEEE Trans. Circuits Syst. Video Technol.* **22**, 649–658 (2012)
19. L Cui, S Wang, X Jiang, S Cheng, Adaptive distributed video coding with correlation estimation using expectation propagation. *Proc. SPIE 8499, Applications of Digital Image Processing XXXV*, 84990M (2012). doi:10.1117/12.929357
20. D Kubasov, J Nayak, C Guillemot, *Proceedings of the IEEE International Workshop on Multimedia Signal Processing*, (Crete, 1–3 Oct 2007)
21. D Agrafiotis, P Ferr, DR Bull, Hybrid key/Wyner-Ziv frames with flexible macroblock ordering for improved low delay distributed video coding, in *Proceedings of the Visual Communications and Image Processing*, (San Jose, 30 Jan 2007), pp. 3097–3100
22. B Wu, X Ji, D Zhao, W Gao, Spatial-aided low-delay Wyner-Ziv video coding. *EURASIP J. Image Video Process.* **2009**, 109057 (2009)
23. A Aaron, S Rane, B Girod, Wyner-Ziv video coding with hash-based motion compensation at the receiver, in *Proceedings of IEEE International Conference on Image Processing (ICIP 04)*, vol. 5 (IEEE, Piscataway, 2004), pp. 3097–3100
24. M Tagliasacchi, A Trapanese, S Tubaro, J Ascenso, C Brites, F Pereira, Exploiting spatial redundancy in pixel domain Wyner-Ziv video coding, in *Proceedings of IEEE International Conference on Image Processing (ICIP 06)*, vol. 5 (IEEE, Piscataway, 2006), pp. 253–256
25. M Guo, Y Lu, F Wu, S Li, W Gao, Distributed video coding with spatial correlation exploited only at the decoder, in *Proceedings of the International Symposium on Circuits and Systems (ISCAS 2007)* (IEEE, Piscataway, 2007), pp. 41–44
26. J Ascenso, C Brites, F Pereira, Motion compensated refinement for low complexity pixel based distributed video coding, in *Proceedings of IEEE Conference on Advanced Video and Signal Based Surveillance (AVSS 05)* (IEEE, Piscataway, 2005), pp. 593–598
27. X Artigas, L Torres, Iterative generation of motion-compensated side information for distributed video coding, in *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, (Genoa, 11–14 Sept 2005)
28. S Ye, M Ouaret, F Dufaux, T Ebrahimi, Improved side information generation with iterative decoding and frame interpolation for distributed video coding, in *Proceedings of IEEE International Conference on Image Processing (ICIP 08)*, (San Diego, 12–15 Oct 2008)
29. F Verbist, N Deligiannis, SM Satti, A Munteanu, P Schelkens, Iterative Wyner-Ziv decoding and successive side-information refinement in feedback channel-free hash-based distributed video coding. *Proc. SPIE 8499, Applications of Digital Image Processing, XXXV*, 84990O (2012). doi:10.1117/12.929680
30. MB Badem, WAC Fernando, JL Martinez, P Cuenca, An iterative side information refinement technique for transform domain distributed video coding, in *Proceedings of the 2009 IEEE International Conference on Multimedia and Expo, ICME'09* (IEEE, Piscataway, 2009), pp. 177–180
31. European Union: The discover codec evaluation 2005. www.discoverdvc.org. Accessed 30 Oct 2013

doi:10.1186/1687-6180-2013-168

Cite this article as: Haj Taieb et al.: Spatial correlation-based side information refinement for distributed video coding. *EURASIP Journal on Advances in Signal Processing* 2013 **2013**:168.

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Immediate publication on acceptance
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com