IMPROVED SIDE INFORMATION GENERATION FOR DISTRIBUTED VIDEO CODING

A. Abou-Elailah [1], J. Farah [2], M. Cagnazzo [1], B. Pesquet-Popescu [1], and F. Dufaux [1]

[1] Signal and Image processing Department, Institut TELECOM - TELECOM Paristech,
46 rue Barrault, F - 75634 Paris Cedex 13, FRANCE
{elailah, frederic.dufaux, marco.cagnazzo, beatrice.pesquet-popescu}@telecom-paristech.fr
[2] Telecommunications Department, Faculty of Engineering, Holy-Spirit University of Kaslik
P.O. Box 446, Jounieh, Lebanon
joumanafarah@usek.edu.lb

ABSTRACT

In distributed video coding, the side information is commonly generated by motion-compensated temporal interpolation of the neighboring reference frames at the decoder side. The side information quality has a strong impact on the final rate-distortion performance of the codec. In this paper, we propose a successive refinement after the decoding of each DCT subband to improve the accuracy of motion compensation between reference frames, in order to obtain a new side information estimation closer to the original Wyner-Ziv frame. Here, we propose two different algorithms in the refinement process of the side information. The experimental results show that the proposed techniques allow an improvement in rate-distortion performance that can reach 1.15 dB for GOP size of 2 and 3.2 dB for longer GOP size, with respect to state-of-the-art techniques, for sequences containing high motion.

Index Terms— Distributed video coding, side information refinement, motion vectors, motion compensation, rate-distortion (RD).

1. INTRODUCTION

Today, most common digital video coding schemes represented by the standardization efforts of ISO/IEC MPEG and ITU-T H.26x are based on Discrete Cosine Transform (DCT), inter-frame and intra-frame predictive coding. In these standards, the encoder exploits the spatial and temporal redundancy existing in a video sequence. In this case, the encoder is significantly more complex than the decoder (typically 5 to 10 times [1]). This kind of architecture is well-suited for applications where the video sequence is encoded once and decoded many times, such as broadcasting or video streaming.

In recent years, this architecture has been challenged by several emerging applications such as wireless video surveillance, multimedia sensor networks, wireless PC cameras, and mobile cameras phones. In these emerging applications, it is essential to have a low complexity encoding.

Distributed Video Coding (DVC) is a new video paradigm which fits well these emerging scenarios since it enables the exploitation of video redundancy at the decoder side only, making the encoder less complex. In other words, the complexity is shifted from the encoder to the decoder. From information theory, the Slepian-Wolf theorem for lossless compression [2] states that it is possible to encode correlated sources (let us call them X and Y) independently and decode them jointly, using a rate similar to that used in a system where frames are encoded and decoded jointly. The Wyner-Ziv theorem [3] extends the Slepian-Wolf theorem to lossy compression, which deals with lossy source coding of X when Side Information (SI) Y is available at the decoder only.

Based on these theoretical results, practical implementations of DVC have been proposed in [4, 5]. DISCOVER [6, 7] is one of the most efficient and popular existing architectures, which is based on transform domain Wyner-Ziv coding. In DISCOVER codec, the images of the sequence are split into two sets of frames, the key frames and the Wyner-Ziv (WZ) frames. The Group of Pictures (GOP) of size n is defined as a set of frames consisting of one key frame and $n - 1$ WZ frames. The key frames are independently encoded and decoded using Intra coding techniques such as H.264/AVC Intra mode. The WZ frames are encoded independently, transformed and quantized. At the decoder, the decoded key frames are used to compute the SI, which is an estimation of the WZ frame being decoded. In order to estimate the SI, the technique used in DISCOVER codec is based on Motion-Compensated Temporal Interpolation (MCTI) [8].

DVC has not reached the performance level of classical inter frame coding yet. It is in part due to the quality of the SI, which has a strong impact on the final Rate-Distortion (RD) performance. Several works have been proposed in order to enhance the SI. An approach proposed by Aaron et al. [9] and by Ascenso et al. [10] consists in sending a hash of the WZ frame being decoded to enhance the interpolation of the SI. However, these techniques demand some additional data (the

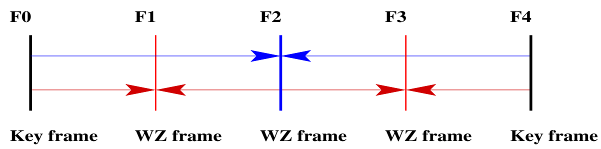


Fig. 1. Interpolation steps for a GOP size 4.

hash) to be sent through the channel. Other techniques exist that can avoid this overhead. They are based on the successive refinement of the SI. A solution proposed by J. Ascenso et al. [11] for pixel domain DVC uses a motion compensated refinement of the SI successively after each decoded bit plane, in order to achieve a better reconstruction of the decoded WZ frame. In [12], the authors proposed a novel DVC successive refinement approach to improve the motion compensation accuracy and the SI. This approach is based on the N-Queen sub-sampling pattern. In [13, 14], solutions are proposed for transform-domain DVC based on the successive refinement of the SI after each decoded DCT band. The authors in [15] propose a solution for transform-domain DVC, which refines the SI after the decoding of all DCT bands in order to improve reconstruction. In VISNET 2 codec [16], the refinement process of the SI is carried out after decoding all DCT bands, and a deblocking filter is used.

In this paper, we propose a new approach in order to enhance the SI in transform-domain DVC. This solution consists in progressively improving the SI after each decoded DCT-band and is particularly efficient for high motion regions and in the case where key frames are separated by a significant number of WZ frames. We first start by generating an Initial Side Information (INSI) by using the backward and forward reference frames, similarly to the SI generated in DISCOVER codec. The decoder reconstructs a Partially Decoded Wyner-Ziv (PDWZ) frame by correcting the INSI with the parity bits of the first DCT-band. Then, the PDWZ frame, along with the backward and forward reference frames, is used to refine the INSI. The refinement approach consists of four modules: Suspicious Vector Detection, Refinement, Mode Selection, and Motion Compensation. More specifically, in the module responsible for refining the motion vectors, we propose two different algorithms in order to prove the efficiency of each algorithm in enhancing the quality of the final decoded WZ frame. Finally, we correct this refined INSI with the parity bits of the next DCT-band, and we repeat the same procedure to decode all DCT-bands of the current WZ frame.

In contrast, the authors in [14] only used the SI and the PDWZ frame to refine the SI for decoding the next band, even though there is more information to be extracted from the backward and forward reference frames. The solution proposed in [13] relies on a much simpler SI refinement technique compared with our proposed approach. Moreover, the same refinement procedure is applied to all blocks regardless of the motion vector reliability. Finally, bidirectional prediction is always used.

band	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
QI 1	16	8	8	0	0	0	0	0	0	0	0	0	0	0	0	0
QI 2	32	8	8	0	0	0	0	0	0	0	0	0	0	0	0	0
QI 3	32	8	8	4	4	4	0	0	0	0	0	0	0	0	0	0
QI 4	32	16	16	8	8	8	4	4	4	4	0	0	0	0	0	0
QI 5	32	16	16	8	8	8	4	4	4	4	4	4	4	0	0	0
QI 6	64	16	16	8	8	8	8	8	8	8	4	4	4	4	4	0
QI 7	64	32	32	16	16	16	8	8	8	8	4	4	4	4	4	0
QI 8	128	64	64	32	32	32	16	16	16	16	8	8	8	4	4	0

Fig. 2. Various 4×4 quantization matrices corresponding to eight rate-distortion points. For each QI, the number of levels is given for the 16 bands

This paper is structured as follows. First, the DISCOVER architecture is introduced in Section 2. The proposed approaches by successive refinement of the SI are described in Section 3. Experimental results are then shown in Section 4 in order to evaluate and compare the RD performance of the proposed approaches. Finally, conclusions and future work are presented in Section 5.

2. DISCOVER CODEC

In this section, we briefly present the DISCOVER codec [6, 7]. First, the video sequence is divided into WZ frames and key frames. The latters are encoded using H.264/AVC Intra coding. Figure 1 shows all necessary interpolations for a GOP size 4. For example, during the interpolation of WZ frame F2, the forward and backward reference frames are key frames F0 and F4. For the interpolation of F1, the reference frames are the key frame F0 and the previously decoded WZ frame F2.

The WZ frame encoding and decoding procedures are detailed in the following.

- **Wyner-Ziv encoder -** At the encoder side, the WZ frame is first transformed using a 4×4 block-based DCT. The DCT coefficients of the entire WZ frame are then organized in 16 bands, indicated by b_k with $k \in [1, 16]$, according to their position within the 4×4 blocks. The low frequency information (i.e. the DC coefficients) are placed in the first band $k = 1$, and the others coefficients are grouped in the AC bands $k = 2, 3, ...16$. Next, each DCT coefficients band b_k is uniformly quantized with 2^{M_k} levels (where the number of bits reserved M_k depends on the DCT band b_k). Figure 2 shows the number of levels for each band for eight different rates $QI = 1, 2, ...8$. The resulting quantized symbols (associated to DCT band b_k) are then split into bit planes. For a given band, the quantized symbols bits of the same significance are grouped together in order to form the corresponding biplane array which is then independently encoded using a channel encoder. The latter, also known as the Slepian-Wolf encoder, is a

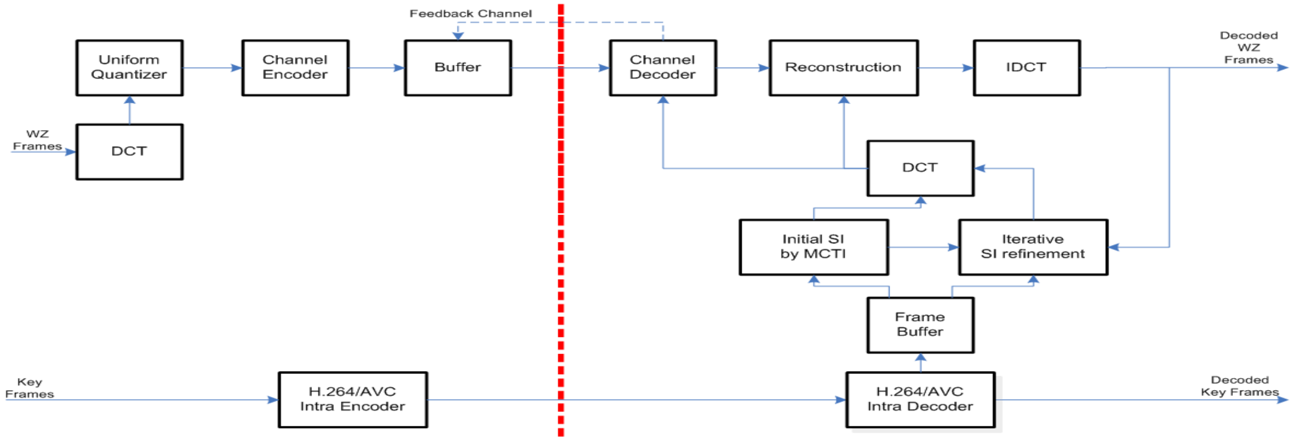


Fig. 3. Overall structure of the proposed DVC codec.

rate-compatible Low-Density Parity Check (LDPC) accumulate code. Each bitplane is successively fed into the channel encoder in order to compute a separate set of parity bits, while the systematic bits are discarded. The parity information is then stored in a buffer and progressively sent in chunks, upon request by the decoder, through the feed-back channel.

- **Generation of side information -** In the DISCOVER scheme, the frame interpolation framework is composed of four modules to obtain high quality SI [8] (preceded by low-pass filtering of the reference frames in order to improve the motion vectors reliability): forward motion estimation between the previous and next reference frames, bi-directional motion estimation to refine the motion vectors, spatial smoothing of motion vectors in order to achieve higher motion field spatial coherence (reduction of the number of false motion vectors), and finally bi-directional motion compensation.
- **Wyner-Ziv decoder -** A block-based 4×4 DCT is carried out over the generated SI (using the previous step) in order to obtain the DCT coefficients which can be seen as a noisy version of the WZ frame DCT coefficients. In order to model the error distribution between corresponding DCT bands of SI and WZ frame, the DISCOVER codec uses a Laplacian distribution. The Laplacian parameter is estimated on-line at the decoder. Once the DCT transformed SI and the residual statistics for a given DCT band b_k are known, the Slepian-Wolf decoder corrects the bit errors in the DCT transformed SI using the parity bits of WZ frame requested through the feedback channel.
- **Reconstruction and inverse transform -** The reconstruction corresponds to the inverse of the quantization using the SI DCT coefficients and the decoded Wyner-Ziv DCT coefficients. After that, the inverse 4×4 DCT transform is carried out, and the entire frame is restored in the pixel domain.

3. PROPOSED METHOD

The block diagram of our proposed codec architecture is depicted in Figure 3. It is based on the DISCOVER codec [6, 7]. The INSI is first computed by MCTI with spatial motion smoothing exactly as in DISCOVER codec. The LDPC parity bits of the first band (DC band) are used in order to correct the corresponding DCT coefficients in INSI; the obtained decoded frame is denoted as Partially Decoded Wyner-Ziv (PDWZ) frame. Here, the two adjacent reference frames and the PDWZ frame are used in order to improve the SI interpolation by using two different approaches. Then, the obtained decoded frame after the first improvement of the INSI is used again as PDWZ in order to improve the last SI for decoding the next DCT band, and so on after each DCT band. The proposed scheme for SI enhancement is illustrated in Figure 4. It consists of three steps based on PDWZ to improve the SI: suspicious vector detection based on the matching criterion, motion vector refinement and smoothing, and optimal motion compensation mode selection.

The first approach is similar to the method in [15]. However, it has been improved in such a way that the SI is progressively refined after the decoding of each DCT band. Moreover, both the matching criterion and the mode selection have been modified as detailed in subsection 3.1, resulting in improved performances.

The second approach consists of the same modules of the first approach. However, a different algorithm is applied in the motion vector refinement module.

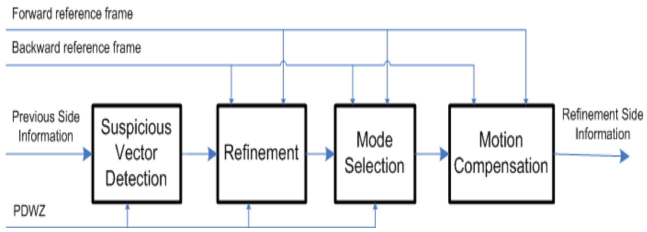


Fig. 4. Proposed Side Information refinement procedure.

3.1. Proposed modules

The proposed SI refinement procedure is illustrated in Figure 4. The used matching criterion and the three main modules, suspicious vector detection, motion vector refinement, and optimal motion compensation mode selection are described as follows.

- **Matching criterion -** In order to exploit the spatial-temporal correlations to enhance the estimated motion vectors, the matching criterion used in this paper is based on the Mean Absolute Difference (MAD). The MAD for the estimated motion vector MV of a Block B is defined as:

$$\text{MAD}(\mathbf{P_0}, F_c, F_r, \mathbf{MV}) = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |F_c(i + x_0, j + y_0) - F_r(i + x_0 + MV_x, j + y_0 + MV_y)| \quad (1)$$

  where $P_0 = (x_0, y_0)$ is the coordinate of the top-left pixel for the current block, F_c is the current frame, F_r the reference one, and $\mathbf{MV} = (MV_x, MV_y)$ the candidate motion vector. The current block has M rows and N columns.

- **Suspicious vector detection -** The motion vectors estimated by MCTI for sequences with low motion are close to the true motion. However, false motion vectors may occur in sequences with high motion and occlusions. In order to identify suspicious vectors, a threshold T_1 is used. For a given block, the MAD is calculated between the PDWZ frame and the last SI (e.g. previous refinement of the SI):

$$\text{MAD}(\mathbf{P_0}, F1, F2(\mathbf{MVB}), \mathbf{0}) < T_1, \quad (2)$$

  where $\mathbf{0} = (0, 0)$ is the null vector, $F1$ and $F2$ are the PDWZ frame and the last refinement SI respectively and $\mathbf{P_0}$ the top left point for the processed block B. If this estimation satisfies the condition defined in equation (2), it is considered to be a true estimation (e.g. the motion vector $\mathbf{MVB}$ for this block is not modified in the first algorithm). Otherwise, it is identified as a suspicious vector and will be further refined. On the other hand, the motion vector $\mathbf{MVB}$ that is identified as a true motion is refined within a search area of ± 2 pixels only in the second algorithm. This refinement is only applied two times within the decoding of DCT bands, the first time being after the decoding of the first band, and the second one after the decoding of all DCT bands.

- **Motion vector refinement -** In order to refine the motion vectors which are identified as suspicious vectors, these motion vectors are re-estimated by bi-directional motion estimation using the matching criterion defined in Eq. (1). For the current block in the PDWZ frame, the block motion estimation determines, between all candidate blocks within a search area in the previous reference frame, the most similar one to the current block, according to the predefined criterion in Eq. (1). The motion vector between the PDWZ frame and the forward reference frame is estimated in the same way. These estimated motion vectors are considered as refined motion vectors for the processed block. In this module, two different algorithms are carried out in order to re-estimate the motion vectors, which are identified as suspicious vectors. However, the motion vectors that are identified as true motion vectors are only refined in the second algorithm. These proposed algorithms are presented below and illustrated in Table 1.

  Algorithm I : This algorithm searches for the most corresponding block in the X frame (X is the backward or forward reference frame) within a search area of ± 16 pixels in one pixel accuracy. The obtained motion vectors are considered to be the bi-directional motion vectors of the current block in the PDWZ frame.

  Algorithm II : Even though, the block size is 8×8 pixels, a window of 16×16 pixels is considered when MAD is calculated in this algorithm. First, the motion vectors between the current block in PDWZ frame and X frame (X is the backward or forward reference frame) is re-estimated within a search area of ± 16 pixels in two pixels accuracy. These motion vectors are then refined within a search area of ± 3 pixels in half-pixel accuracy. For the motion vectors that are identified as true motion vectors, these vectors are only refined two times in this algorithm, within a search area of ± 2 pixels in half-pixel accuracy, once after decoding the first DCT band, and another time after decoding all DCT bands.

  It can be verified that, in terms of the computational load, the two algorithms have the same complexity in finding the corresponding blocks.

- **Optimal motion compensation mode selection -** The objective of this step is to generate an optimal motion-compensated estimate by selecting the most similar block to the current block from three sources: the previous reference frame (BACKWARD MODE), the

Proposed Algorithms	
step 1: Divide the PDWZ frame into 8×8 square blocks of pixels.	
step 2: Compute the Mean Absolute Difference (MAD) between the current block and the corresponding block in the last SI (Equation (1)).	
Algorithm 1	**Algorithm 2**
	In this algorithm, when the MAD is calculated, the block size is considered to be 16×16 pixels. **Set N** to **1** if only the first DCT band is decoded, and **set N** to ∞ if all DCT bands are decoded.
step 3: if MAD $<$ T1, go to **step 2** for treating the next block.	**step 3** : if MAD $<$ T1, and N = 1 or = ∞, go to **step 9**. **step 4**: if MAD $<$ T1, and N $\neq$ 1 and $\neq \infty$, go to **step 2** for treating the next block.
step 4: if MAD $\geq$ T1, Set the search area (±16 pixels) to re-estimate the motion vector.	**step 5**: if MAD $\geq$ T1, Set the search area (±16 pixels) to re-estimate the motion vector.
step 5: Find the corresponding block in X frame (X is the backward or forward reference frame) to the current block in PDWZ, in one pixel accuracy, within the search area.	**step 6** : Find the corresponding block in X frame (X is the backward or forward reference frame) to the current block in PDWZ in two pixels accuracy within the search area. **step 7**: Refine the obtained motion vector within a search area of ±3 pixels in half-pixel accuracy.
step 6: Save the obtained motion vectors, and go to **step 2** for treating the next band.	**step 8**: Save the obtained motion vectors, and go to **step 2** for treating the next band. **step 9**: Refine the motion vector of the last SI within a search area of ±2 pixels in half-pixel accuracy. Then, go to **step 2** for treating the next band.
Final step : Optimal motion compensation mode selection.	

Table 1. Proposed algorithms

next reference frame (FORWARD MODE), and bi-directional motion-compensated average of the previous and next reference frames (BIMODE). The decision among these modes is performed according to the following equations, and a threshold T_2 is established.

$$\begin{cases} \text{if } |\text{MAD}_n - \text{MAD}_p| < T_2 \\ \quad \text{MODE=BIMODE} \\ \text{otherwise} \\ \quad \text{if MAD}_n < \text{MAD}_p \\ \quad\quad \text{MODE=FORWARD MODE} \\ \quad \text{otherwise} \\ \quad\quad \text{MODE=BACKWARD MODE} \end{cases}$$

where MAD_p and MAD_n are the estimated mean absolute differences between the current block (in PDWZ) and the corresponding blocks (e.g. the blocks which minimize the MAD) in the previous and next reference frames respectively.

The refinement SI obtained after decoding the band b_k is used at the Wyner-Ziv decoder as a new SI for the decoding of the next band b_{k+1}, and so forth for all bands of the WZ frame being decoded. Then, after decoding all bands, a new SI is generated to perform again the reconstruction step to get the final WZ frame. The proposed scheme for this procedure is illustrated in Figure 5.

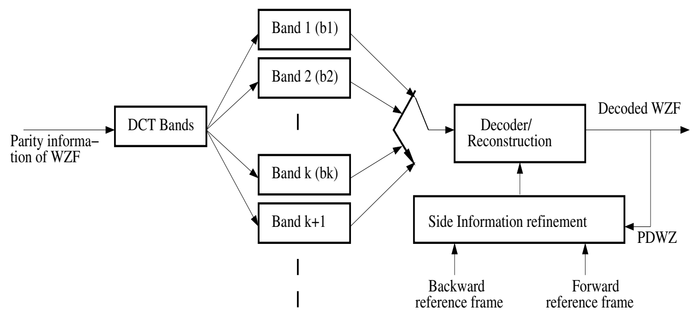


Fig. 5. Proposed technique for successive refinement of SI and WZ frame decoding.

4. EXPERIMENTAL RESULTS

In order to evaluate the performance of the proposed algorithms, we performed extensive simulations, adopting the same test conditions as described in DISCOVER [6, 7] (i.e. test video sequences are at QCIF spatial resolution and sampled at 15 frames/sec).

In Table 2, the RD performance of the proposed algorithms is shown for different values of T_1 and T_2 compared to DISCOVER codec. In the simulations, we have set $T_1 = T_2 = 4$ due to the high performance and low computational load achieved for these values.

The obtained results are compared to the recent projects: DISCOVER [6, 7] and VISNET2 [16] codec. Moreover, the results of the proposed algorithms are also compared to H264/AVC Intra and H264/AVC No motion.

Figure 6 shows the visual results of the decoded frames for Foreman. For a GOP equal to 8, the decoded frame obtained by DISCOVER codec contains block artifacts (frame number 125). On the contrary, the decoded frames obtained by the proposed algorithms are significantly better. The improvement of the decoded frame obtained by the first algorithm (center) is up to 3.7 dB with less requested bits (down from 30.84 Kbits to 21.82 Kbits). The second algorithm (right) gives even better performance compared to DISCOVER codec, since it achieves an improvement up to 5.8 dB with a reduction of 40% in the requested bits (30.84 Kbits to 18.46 Kbits).

The RD performance of the proposed algorithms (Alg. I and Alg. II) for the Foreman and Soccer sequences is shown

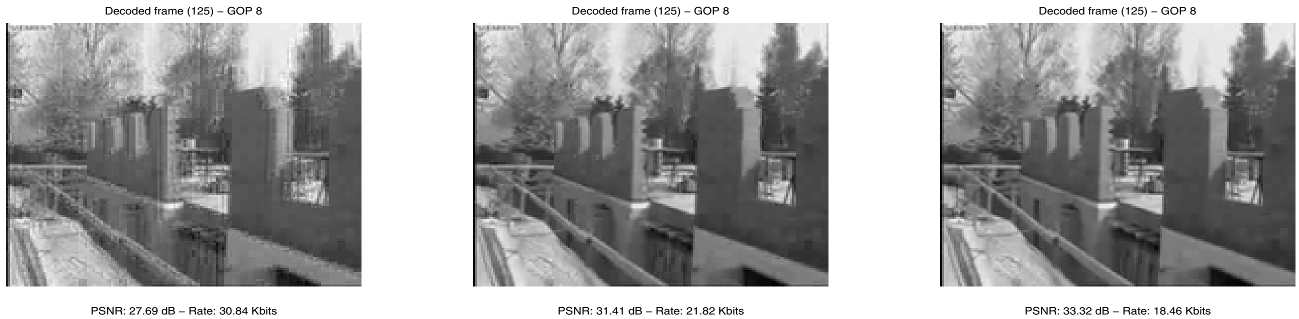


Fig. 6. Visual results comparisons between the proposed algorithms (alg. 1 - center, and alg. 2 - right) and DISCOVER codec (left) for the decoded frame 125 of Foreman sequence for GOP size 8.

Foreman - GOP size = 2					
	Alg. I				
	$T_2 = 4$			$T_1 = 4$	
	$T_1 = 0$	$T_1 = 4$	$T_1 = 8$	$T_2 = 0$	$T_2 = 8$
Δ_R (%)	-11.48	**-13.40**	-08.98	-09.88	**-13.43**
Δ_{PSNR} [dB]	00.60	**00.73**	00.47	00.54	**00.73**
	Alg. II				
	$T_2 = 4$			$T_1 = 4$	
	$T_1 = 0$	$T_1 = 4$	$T_1 = 8$	$T_2 = 0$	$T_2 = 8$
Δ_R (%)	**-20.91**	**-20.67**	–19.30	-17.02	**-20.53**
Δ_{PSNR} [dB]	**01.16**	**01.14**	01.05	00.94	**01.13**

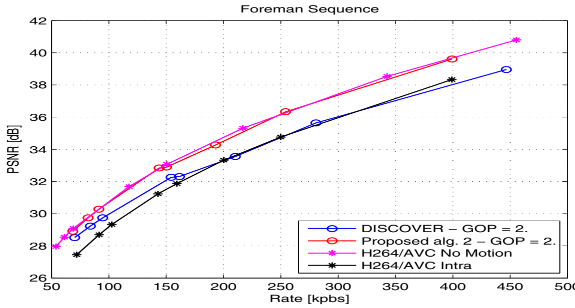
Table 2. Rate-distortion performance gain for *Foreman*, towards DISCOVER codec, for different values of T_1 and T_2 using Bjontegaard metric.

Foreman					
	VIS. 2	Alg. I	Alg. II	H264 Intra	H264 No M.
GOP size = 2					
Δ_R (%)	-3.44	-13.40	**-20.95**	2.52	**-22.50**
Δ_{PSNR} [dB]	0.16	0.73	**1.15**	-0.18	**1.25**
GOP size = 4					
Δ_R (%)	-12.72	-32.33	**-46.98**	-20.29	**-43.5**
Δ_{PSNR} [dB]	0.54	1.6	**2.38**	1.07	**2.17**
GOP size = 8					
Δ_R (%)	-22.13	-48.09	**-67.96**	-37.45	**-57.50**
Δ_{PSNR} [dB]	0.85	2.28	**3.28**	2.13	**2.88**
Soccer					
GOP size = 2					
Δ_R (%)	-9.48	-17.88	**-20.90**	-41.76	**-49.40**
Δ_{PSNR} [dB]	0.40	0.87	**1.05**	2.35	**2.80**
GOP size = 4					
Δ_R (%)	-15.94	-26.89	**-31.73**	-76.00	**-74.00**
Δ_{PSNR} [dB]	0.63	1.32	**1.61**	4.35	**4.25**
GOP size = 8					
Δ_R (%)	-20.40	-31.90	**-39.16**	-90.00	**-85.00**
Δ_{PSNR} [dB]	0.74	1.53	**1.90**	5.57	**5.22**

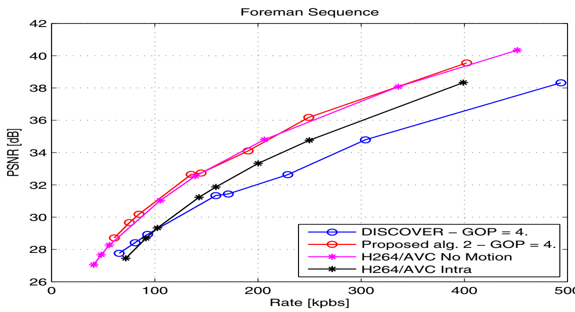
Table 3. Rate-distortion performance gain for *Foreman* and *Soccer*, towards DISCOVER codec, using Bjontegaard metric.

in Table 3 with different GOP sizes (2, 4 and 8) in comparison to DISCOVER codec, using Bjontegaard metric[17]. We start by comparing our algorithms to existing DVC schemes (by considering the first three columns). For Foreman sequence, the VISNET 2 (VIS. 2 in Table 3) codec allows respectively a gain up to 0.16, 0.54, and 0.85 dB with a reduction in the rate up to 3.44, 12.72, and 22.13 % compared to DISCOVER codec for a GOP size 2, 4, and 8. The first proposed algorithm (Alg. I) gives respectively a gain up to 0.73, 1.6, and 2.28 dB and an average rate reduction of 13.4, 32.33, and 48.09 % for a GOP size 2, 4, and 8 compared to DISCOVER codec. It is clear that the first algorithm is better than DISCOVER and VISNET 2 codecs for all GOP sizes. Moreover, the second algorithm (Alg. II) can achieve a significant gain up to 3.28 dB with a reduction in the rate up to 67.96 % compared to DISCOVER codec for a GOP size 8. For a GOP length equal to 2 and 4, the second algorithm allows respectively a gain up to 1.15 and 2.38 dB compared to DISCOVER codec. It is clear that the gain in RD performance increases with the GOP length. In this case, classical interpolation techniques for SI generation become less effective. For Soccer sequence, the second algorithm can also achieve the best gain in RD performance compared to DISCOVER and VISNET 2 codecs. The second algorithm can respectively achieve a significant gain up to 1.05, 1.61, and 1.9 dB and an average rate reduction of 20.9, 31.73, and 39.16 % for GOP size equal to 2, 4, and 8. Surprisingly, for Foreman and with a GOP length equal to 8, the second algorithm can achieve a gain up to 0.4 dB compared to H.264/AVC No motion (H264 No M. in Table 3).

In Figure 7, The performance of the second algorithm is compared to H.264/AVC Intra, H.264/AVC No motion and Discover codec, for Foreman sequence. The difference between the performance of DISCOVER codec and H.264/AVC No motion is up to 1.25 dB for a GOP size 2. In this case, the proposed algorithm can achieve the same performance as H.264/AVC No motion, whereas the performance of DISCOVER codec is close to H.264/AVC Intra. For a GOP size 4, the proposed algorithm (Alg. II) gives a gain up to 0.15 dB compared to H.264/AVC No motion. Moreover, the performance of DISCOVER codec is less than the H.264/AVC Intra by 1.07 dB.

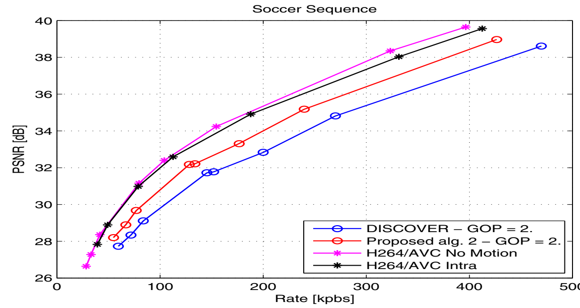


(a) RD performance for Foreman sequence with GOP = 2.

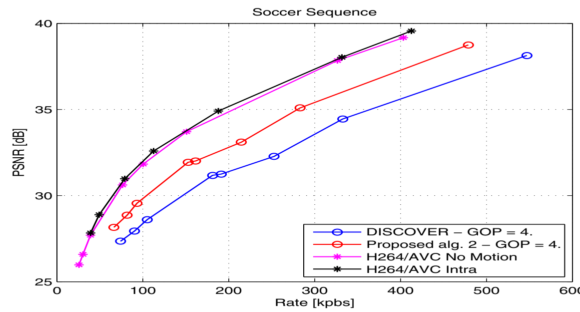


(b) RD performance for Foreman sequence with GOP = 4.

Fig. 7. RD performance comparison - DICOVER, Algorithm II, H264/AVC Intra, and H264/AVC No motion.



(a) RD performance for Soccer sequence with GOP = 2.



(b) RD performance for Soccer sequence with GOP = 4.

Fig. 8. RD performance comparison - DISCOVER, Algorithm II, H264/AVC Intra, and H264/AVC No motion.

The performance of Soccer sequence is shown in Figure 8 for a GOP size 2 and 4. It is clear that the difference between the performance of DISCOVER codec, H.264/AVC Intra and H.264/AVC No motion is significant. The proposed algorithm can reduce this gap by half (see also Table 3).

In Figure 9, we show the performance of DISCOVER codec and algorithm II for all GOP sizes. For Foreman sequence, the performance loss, using DISCOVER codec, exceeds 1.5 dB, when the GOP size increases from 2 to 4 and from 4 to 8. In our proposed algorithm, this loss is less than 0.1 dB, when the GOP size increases from 2 to 4, and less than 0.4 dB between the GOP sizes 2 and 8, despite a big difference of 3 dB in the case of DISCOVER codec. The performance of Soccer sequence is also shown, which shows that the difference between the performances for different GOP sizes becomes small.

For Hall Monitor and Coastguard sequences, the gains between proposed algorithms and DISCOVER codec are smaller and range between 0.2 and 0.9 dB.

It can be noticed that the performance gains are more substantial for high motion sequences and for long GOP sizes. They are mainly associated with the proposed technique for successive refinement of the side information interpolation, which is the major contribution with respect to the reference codec.

5. CONCLUSION AND FUTURE WORK

Successive refinement of the side information using the partially decoded frame and the two adjacent reference frames in DISCOVER codec was proposed in this paper, based on the successive decoding of the DCT bands.

Experimental results showed that our proposed method can achieve a gain in RD performance up to 1.15 dB for a GOP size of 2 and 3.2 dB for longer GOP sizes, compared to DISCOVER codec, especially when the video sequence contains high motion. The improvement becomes even more important as the GOP size increases.

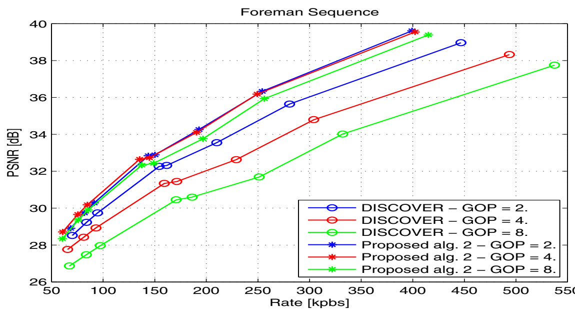
Future work will be focusing on further improvement of the side information in order to achieve a better RD performance.

6. ACKNOWLEDGMENT

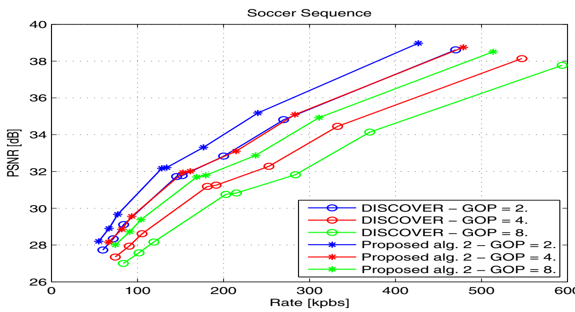
This work was partly supported by a research grant from the Franco-Lebanese CEDRE program.

7. REFERENCES

[1] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding



(a) RD performance for Foreman sequence with GOP = 2, 4, and 8.



(b) RD performance for Soccer sequence with GOP = 2, 4, and 8 .

Fig. 9. RD performance comparison - DISCOVER and Second algorithm for GOP = 2, 4, and 8.

standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, pp. 560–576, 2003.

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

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

[4] R. Puri and K. Ramchandran, "PRISM: A video coding architecture based on distributed compression principles," *EECS Department, University of California, Berkeley, Tech. Rep. UCB/ERL M03/6*, 2003.

[5] B. Girod, A. Aaron, S. Rane, and D. Rebello-Monedero, "Distributed video coding," *Proceedings of the IEEE*, vol. 93, pp. 71–83, Jan. 2005.

[6] X. Artigas, J. Ascenso, M. Dalai, S. Klomp, D. Kubasov, and M.Ouaret, "The DISCOVER codec: Architecture, techniques and evaluation," *Proc. of Picture Coding Symposium*, Oct. 2007.

[7] "Discover project," `http://www.discoverdvc.org/`.

[8] C. Brites, J. Ascenso, and F. Pereira, "Improving transform domain Wyner-Ziv video coding performance," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2006)*, vol. 2, pp. 525–528, May 2006.

[9] A. Aaron, S. Rane, and B. Girod, "Wyner-Ziv video coding with hash-based motion compensation at the receiver," *Proc. Int. Conf. on Image Processing*, vol. 05, pp. 3097–3100, Oct. 2004.

[10] J. Ascenso and F. Pereira, "Adaptive hash-based side information exploitation for efficient Wyner-Ziv video coding," *Proc. Int. Conf. on Image Processing*, vol. 03, pp. 29–32, Oct. 2007.

[11] J. Ascenso, C. Brites, and F. Pereira, "Motion compensated refinement for low complexity pixel based distributed video coding," *Proceedings of the IEEE international conference on Advanced Video and Signal-Based Surveillance*, pp. 593 – 598, Sept. 2005.

[12] X. Fan, O. Au, N. Cheung, Y. Chen, and J. Zhou, "Successive refinement based Wyner-Ziv video compression," *Signal Processing: Image Communication*, vol. 25, pp. 47–63, Jan. 2010.

[13] M.B. Badem, W.A.C. Fernando, J.L. Martinez, and P. Cuenca, "An iterative side information refinement technique for transform domain distributed video coding," *IEEE International Conference on Multimedia and Expo, ICME*, pp. 177 – 180, 2009.

[14] R. Martins, C. Brites, J. Ascenso, and F. Pereira, "Refining side information for improved transform domain wyner-ziv video coding," *IEEE Transactions on circuits and systems for video technology*, vol. 19, no. 9, pp. 1327 – 1341, Sept. 2009.

[15] S. Ye, M. Ouaret, F. Dufaux, and T. Ebrahimi, "Improved side information generation for distributed video coding by exploiting spatial and temporal correlations," *EURASIP Journal on Image and Video Processing*, vol. 2009, pp. 15 pages, 2009.

[16] J. Ascenso, C. Brites, F. Dufaux, A. Fernando, T. Ebrahimi, F. Pareira, and S. Tubaro, "The VISNET II DVC Codec: Architecture, Tools and Performance," *Proc. of the 18th European Signal Processing Conference (EUSIPCO 2010)*, 2010.

[17] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," in *VCEG Meeting*, Austin, USA, Apr. 2001.