



HAL
open science

Side information improvement in transform-domain distributed video coding

Abdel-Bassir Abou-Elailah, Giovanni Petrazzuoli, Joumana Farah, Marco Cagnazzo, Beatrice Pesquet-Popescu, Frederic Dufaux

► To cite this version:

Abdel-Bassir Abou-Elailah, Giovanni Petrazzuoli, Joumana Farah, Marco Cagnazzo, Beatrice Pesquet-Popescu, et al.. Side information improvement in transform-domain distributed video coding. Applications of Digital Image Processing XXXV, SPIE, Aug 2012, San Diego, United States. hal-01436384

HAL Id: hal-01436384

<https://hal.science/hal-01436384>

Submitted on 10 Jan 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Side Information Improvement in Transform-Domain Distributed Video Coding

A. Abou-Elailah^a, G. Petrazzuoli^a, J. Farah^b, M. Cagnazzo^a, B. Pesquet-Popescu^a and F. Dufaux^a

^aSignal and Image processing Department, Institut TELECOM - TELECOM Paristech,
46 rue Barrault, F - 75634 Paris Cedex 13, FRANCE;

^bTelecommunications Department, Faculty of Engineering, Holy-Spirit University of Kaslik
P.O. Box 446, Jounieh, Lebanon

ABSTRACT

Side Information (SI) has a strong impact on the rate-distortion performance in distributed video coding. The quality of the SI can be impaired when the temporal distance between the neighboring reference frames increases. In this paper, we introduce two novel methods that allow improving the quality of the SI. In the first approach, we propose a new estimation method for the initial SI using backward and forward motion estimation. The second one consists in re-estimating the SI after decoding all WZFs within the current Group of Pictures (GOP). For this purpose, the SI is first successively refined after each decoded DCT band. Then, after decoding all WZFs within the GOP, we adapt the search area to the motion content. Finally, each already decoded WZF is used, along with the neighboring ones, to estimate a new SI closer to the original WZF. This new SI is then used to reconstruct again the WZF with better quality. The experimental results show that, compared to the DISCOVER codec, the proposed method reaches an improvement of up to 3.53 dB in rate-distortion performance (measured with the Bjontegaard metric) for a GOP size of 8.

Keywords: Distributed video coding, Side information, Rate-distortion performance, Long duration GOP, Adaptive search area

1. INTRODUCTION

Video coding standards, such as H.264/AVC, present an encoder much more complex than the decoder. This is suitable for one-to-many broadcast transmission: the video is encoded once and decoded by many users. In such a scheme, temporal correlation is exploited at the encoder side: motion estimation and compensation is performed between adjacent frames. As a consequence, huge memory and computational resources are necessary at the encoder. In contrast, for some applications, such as in wireless sensor networks, a low-complexity encoder is required. More precisely, wireless sensors have limited memory and power and the task of motion estimation has to be shifted to the decoder side. In this context, Slepian-Wolf and Wyner-Ziv theorems^{1,2} establish, under some constraints, that the dependence between correlated sources can be exploited only at the decoder side, without loss in the Rate-Distortion (RD) performance with respect to classical predictive coding.

This paradigm is called Distributed Video Coding (DVC)³ when it is applied to video sequences. A very popular architecture in this context is the so-called Stanford codec⁴, which is adopted in this paper (see Fig. 1): the video stream is split into Key Frames (KFs) and Wyner-Ziv Frames (WZFs). The distance between two KFs is called GOP size, and usually is a power of two. The KFs are INTRA-coded. The WZFs are transformed, quantized and fed to a channel encoder. Only the parity bits are sent to the decoder, while the systematic bits are discarded. At the decoder, an estimation of the WZF is needed, by

Further author information: (Send correspondence to A. Abou-Elailah)

A. Abou-Elailah, G. Petrazzuoli, M. Cagnazzo, B. Pesquet-Popescu and F. Dufaux: E-mail: {elailah, petrazzuoli, marco.cagnazzo, beatrice.pesquet, frederic.dufaux}@telecom-paristech.fr

J. Farah: E-mail: joumanafarah@usek.edu.lb

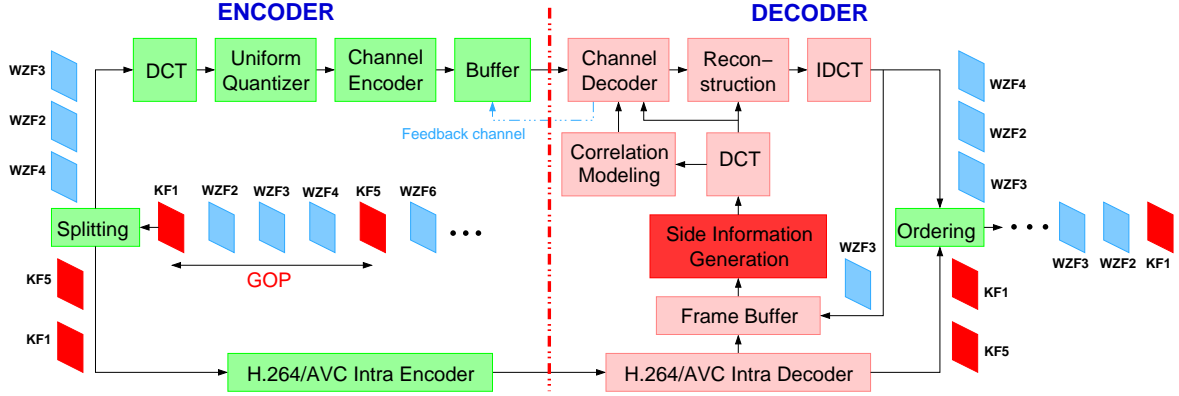


Figure 1: Stanford architecture for Distributed Video Coding (example for $\text{GOP} = 4$).

interpolating/extrapolating the already decoded frames. This estimation is called Side Information (SI) and is considered as a noisy version of the true WZF. The errors can be corrected by the parity bits, which are requested via a feedback channel.

Nevertheless, the performance gap between DVC and predictive coding remains. This is in part due to the quality of the SI, which has a strong impact on the final RD performance. This is even more evident for long duration GOPs, *i.e.* when two successive KFs are far apart. Only a small number of works propose to enhance the SI for long duration GOPs. In other works⁵⁻⁷, a hierarchical structure for estimating and decoding the WZFs for long duration GOP and for GOP sizes different from a power of two is proposed.

In our previous work⁸, we have proposed a new approach to combine the global and local motion estimation at the decoder, in order to improve the quality of the SI. In this approach, the global parameters are estimated at the encoder using SIFT features. We also introduce another approach⁹ based on support vector machine for the combination of global and local motion estimation. High-order motion interpolation has also been proposed¹⁰ in order to cope with object motion with non-zero acceleration.

In the previous work presented by Petrazzuoli et al.¹¹, when the temporal distance between the reference frames is greater than two, the SI is re-estimated for the WZFs after decoding all WZFs within the current GOP. This re-estimation consists in applying the Motion-Compensated Temporal Interpolation (MCTI)¹² technique between the neighboring decoded WZFs, without using the current decoded WZF. In our previous work¹³, we proposed solutions for improving the SI quality in transform-domain DVC, based on successive refinement of the SI after each decoded DCT band.

In this paper, we propose two new methods for SI improvement in transform-domain DVC. First, we propose a new approach based on backward and forward motion estimation, in order to generate the initial SI. It consists in selecting reliable motion vectors from the backward and forward estimations. Second, we aim at refining the SI for large GOP sizes. In this case, the central SI is of worse quality w.r.t. the lateral ones, because the reference frames used for estimating the central WZF are farther apart. The consequence is that the PSNR of the decoded frames fluctuates within the GOP. Therefore, we propose to re-estimate the SI using the already decoded WZF and the adjacent decoded frames (WZF or KF). During the re-estimation procedure, an adaptive search area and a variable block size are also used. Finally, the WZFs are reconstructed with an improved quality, using the same parity bits sent during the first step.

This paper is organized as follows: in Section II, we discuss some of the related works. In Section III, the proposed methods are presented. In Section IV, experimental results are shown and, in Section V, we draw some conclusions.

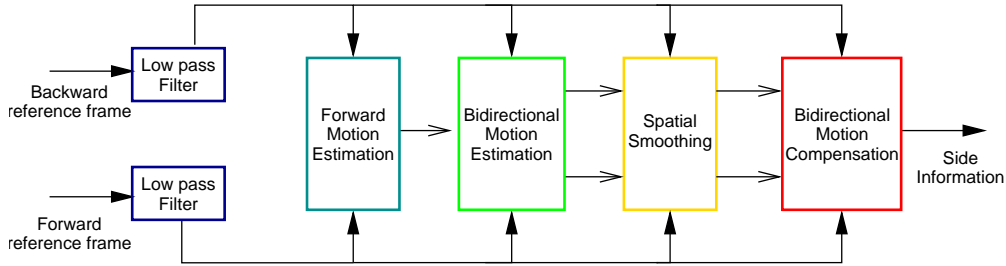


Figure 2: MCTI technique.

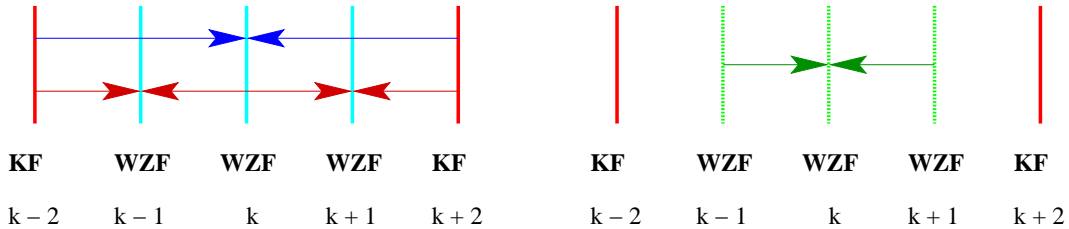


Figure 3: WZF estimation for GOP size = 4 (the already decoded WZFs are in green dashed lines.)

2. RELATED WORKS

2.1 DISCOVER codec

The DISCOVER codec¹⁴ is one of the reference architectures for DVC. In this section, the procedure of encoding and decoding the WZFs is described. The WZF is transformed by a 4×4 DCT and each coefficient is quantized with a number of levels that depends on the DCT band. For each band, the bits are grouped in bit-planes and independently encoded by a Low Density Parity Check Accumulate (LDPCA) code. Only the parity bits are sent to the decoder side. The SI (*i.e.* the estimation of the WZF at the decoder) is obtained by applying the MCTI technique between the forward and backward reference frames. Then, this SI is transformed by a 4×4 DCT: these DCT coefficients are considered as a noisy version of the true WZF DCT coefficients. The errors are corrected by using the parity bits sent by the encoder. The reconstruction is performed by using the SI DCT coefficients and the decoded DCT coefficients. Let i be the decoded quantization index and y the SI DCT coefficient. The reconstruction step¹⁵ consists in computing the expectation $\hat{x} = \mathbb{E}[x|x \in B_i, y]$, where B_i is the quantization interval corresponding to the index i . Finally, the inverse DCT is applied.

2.2 SI construction by MCTI technique

MCTI¹⁶ is commonly used in DVC to generate the SI. Fig 2 shows the architecture of MCTI technique. Backward Reference Frame (BRF) and Forward Reference Frame (FRF) are used to generate a SI for the current WZF. First, the two reference frames are filtered, in order to smooth out noise. The next step is a motion estimation from FRF to BRF (backward motion estimation) to estimate for each position \mathbf{p} , a motion vector $\mathbf{v}(\mathbf{p})$. Bidirectional motion estimation algorithm is employed to refine the motion vectors obtained in the forward motion estimation procedure. At this stage, we search for the vector that intersects the WZF in the point closest to \mathbf{p} (let it $\mathbf{v}(\mathbf{q})$): the backward and forward vectors \mathbf{u} and \mathbf{w} are defined as $\mathbf{u}(\mathbf{p}) = \frac{1}{2}\mathbf{v}(\mathbf{q})$ and $\mathbf{w}(\mathbf{p}) = -\frac{1}{2}\mathbf{v}(\mathbf{q})$. The vectors $\mathbf{u}(\mathbf{p})$ and $\mathbf{w}(\mathbf{p})$ are refined around the positions $\mathbf{p} + \mathbf{u}(\mathbf{p})$ and $\mathbf{p} + \mathbf{w}(\mathbf{p})$, using a vector $\mathbf{e} \in W = \{-1, 0, 1\} \times \{-1, 0, 1\}$: we search for \mathbf{e} that gives the best matching between the BRF and FRF. A second refinement step is performed with an halved block and search window size. In order to smooth out the two motion vector fields (MVF), a weighted median filter is then applied. Finally, the estimation of the WZF consists in taking the average of the two motion compensated reference frames.

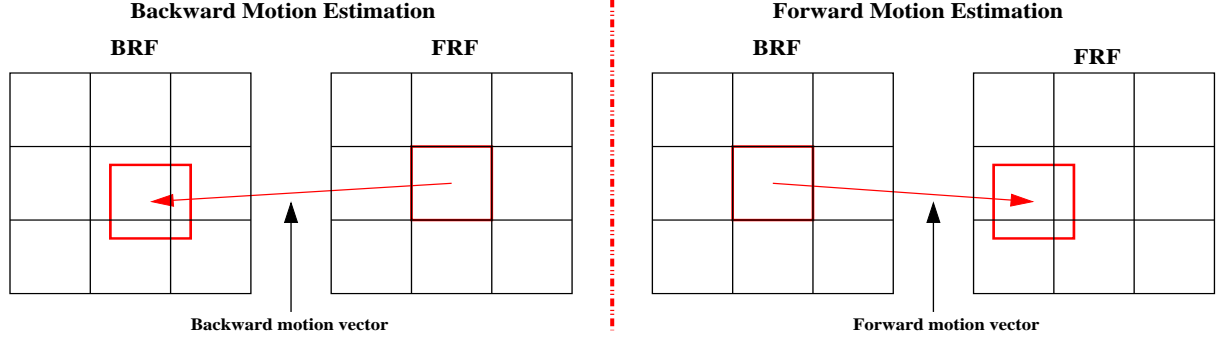


Figure 4: Backward and forward motion estimation.

2.3 SI construction for large GOP sizes

Let us start by describing the reference technique. Let I_k be the WZF that we want to estimate. When the GOP size is equal to 2, the decoding procedure is very simple. The frames used for the interpolation are the decoded KFs \hat{I}_{k-1} and \hat{I}_{k+1} . If the GOP size is equal to 4, the WZF I_k is usually estimated by using the decoded KFs \hat{I}_{k-2} and \hat{I}_{k+2} (Fig. 3 left). Then, this estimation is corrected using parity bits, thus producing the decoded WZF \hat{I}_k . The WZF I_{k-1} is estimated using \hat{I}_{k-2} and \hat{I}_k , then I_{k+1} is obtained by interpolating \hat{I}_k and \hat{I}_{k+2} . The consequence is that \hat{I}_{k-1} and \hat{I}_{k+1} are of better quality with respect to \hat{I}_k : in fact the PSNR of the decoded WZFs is varying within the GOP depending on the distance between the reference frames. In our previous work¹¹, we proposed to add another step: once I_{k-1} and I_{k+1} have been decoded, the frame I_k can be re-estimated by applying the MCTI technique between \hat{I}_{k-1} and \hat{I}_{k+1} (Fig. 3 right). The reconstruction is applied again using the new SI to obtain the final decoded WZF, without requesting any additional parity bits. This procedure can be extended for other GOP sizes.

2.4 Successive refinement techniques

SI improvement can also be obtained by using a successive refinement technique. In our previous work,¹³ after the decoding of each DCT band, three steps are performed in order to refine the SI: Suspicious vector detection, Refinement, and Mode decision and compensation.

The first step consists in detecting possible false vectors. For this purpose, the Mean Absolute Difference (MAD) is computed between the actual SI and the Partially Decoded WZF (PDWZF), for the vector associated to the current block of size 8×8 pixels. If the MAD is greater than a given threshold, this vector is classified as suspicious. In this case, the vector is re-estimated using a fixed search area of ± 16 pixels, with a precision of two pixels and using an extended block of size 12×12 pixels. Then, the obtained motion vector is refined within a search area of ± 3 pixels with a half-pixel accuracy. Otherwise, in the second step, this vector is considered as the true motion and will be only refined within a small search area. The last step consists in the choice of motion compensation modes. Let MAD_p and MAD_n be the MAD between the current block in the PDWZF and the corresponding block in the previous and next reference frames, respectively. If $|MAD_p - MAD_n| < \gamma$, we compute an average of the two frames (“bimode”); otherwise, if $MAD_p < MAD_n$, only the backward reference frame is used for motion compensation (“backward mode”), and, if $MAD_p > MAD_n$, only the forward reference is used (“forward mode”).

3. PROPOSED METHODS FOR SI IMPROVEMENT

3.1 Part I - Side Information Generation (SIG)

In DISCOVER codec, the SI is generated using MCTI technique. In this paper, we propose a new approach to generate the SI, which is based on backward and forward motion estimation. We refer to backward motion estimation when we search in the BRF to find the most similar block to the target block in FRF and forward motion estimation when the most similar block to the target block in BRF is found in FRF (see Fig. 4).

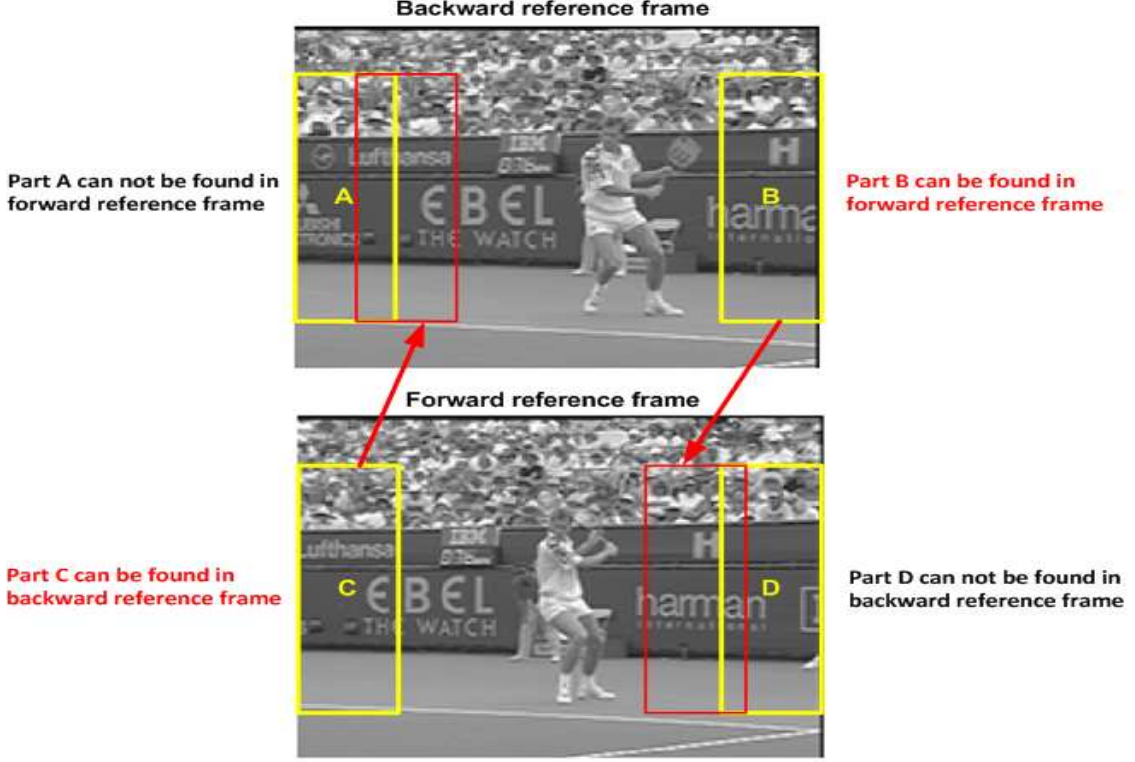


Figure 5: An example of backward and forward reference frames.

Fig. 5 shows an example of four parts A, B, C and D in the backward and forward reference frame. The parts A and B are defined in BRF, and the parts C and D are defined in FRF. As shown in the figure, the parts A and D cannot be found in FRF and BRF respectively. Conversely, the parts B and C can be found in FRF and BRF respectively. In backward motion estimation, we aim at finding the true motion vector for each block in FRF. Thus, the motion vectors for the blocks in part D cannot be reliable. However, the motion vectors for the same block positions can be reliable and accurate in forward motion estimation (the part B can be found in FRF). Similarly, the motion vectors for the blocks in part A are not reliable in forward motion estimation. In order to find the correct motion vectors for all blocks, we propose a new method which consists in using both backward and forward motion estimations. The proposed SI generation is depicted in Fig. 6. This new approach is described as follows:

- **Low-Pass Filter:** The reference frames are padded and low-pass filtered in order to improve the motion vectors reliability.
- **Backward and Forward Motion Estimation:** A block matching algorithm is applied to estimate the backward and forward motion vector fields. These motion estimations are computed with $BS_0 \times BS_0$ block size, a search area (\mathbf{S}) of $\pm SA_0$ pixels, and a step size of N_0 pixels. In the block matching algorithm, Weighted Mean Absolute Difference (WMAD) criterion is used to compute the similarity between the target block b in F_1 (F_1 can be the BRF in forward motion estimation or the FRF in backward motion estimation) and the shifted block in F_2 (F_2 can be the BRF in backward motion estimation or the FRF in forward motion estimation) by the motion vector $\mathbf{v} \equiv (v_x, v_y) \in \mathbf{S}$ as follows:

$$WMAD(b, \mathbf{v}) = \frac{1}{BS_0^2} \sum_{x=x_0}^{x_0+BS_0} \sum_{y=y_0}^{y_0+BS_0} |F_1(x, y) - F_2(x + v_x, y + v_y)| \left(1 + \lambda \sqrt{v_x^2 + v_y^2}\right), \quad (1)$$

where (x_0, y_0) is the up-left pixel of the block b and λ is a penalty factor which allows to penalize the

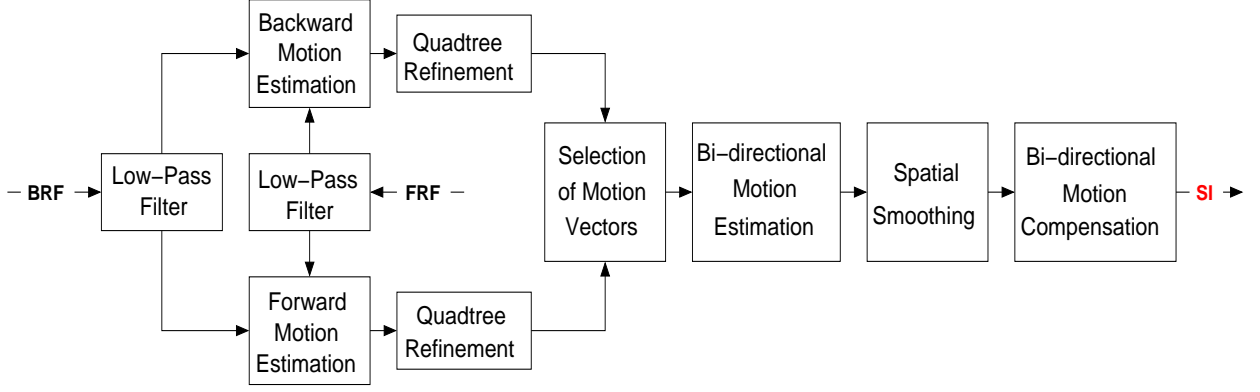


Figure 6: Proposed SI generation.

MAD by the length of the motion vector $\|\mathbf{v}\| = \sqrt{v_x^2 + v_y^2}$. λ is empirically set to 0.01. The objective of the block matching algorithm is to find the most similar block in F_2 to the target block in F_1 . In other words, the algorithm aims at obtaining the best motion vector \mathbf{V}_b for the block b by minimizing the WMAD as follows:

$$\mathbf{V}_b = \arg \min_{\mathbf{v}_i \in \mathcal{S}} \text{WMAD}(b, \mathbf{v}_i). \quad (2)$$

The obtained motion vectors are refined in N_1 ($N_1 < N_0$) pixel(s) accuracy within a search area of $\pm SA_1$ pixels ($SA_1 \ll SA_0$). The MAD criterion is used in the refinement process. Let $\mathbf{V}_b^{\text{back}}$ and $\mathbf{V}_b^{\text{for}}$ be the obtained backward and forward motion vectors for the block b respectively, and MAD_{back} and MAD_{for} the mean absolute differences corresponding to the motion vectors $\mathbf{V}_b^{\text{back}}$ and $\mathbf{V}_b^{\text{for}}$ respectively. We now aim at replacing the false motion vectors that can be obtained at the borders of the image by reliable ones (for example, the obtained motion vectors for the blocks that form the part D in backward motion estimation are not reliable). The motion vectors can be improved as follows:

$$\left\{ \begin{array}{l} \text{if } |\text{MAD}_{\text{back}} - \text{MAD}_{\text{for}}| < T_b \\ \quad \text{The motion vectors } \mathbf{V}_b^{\text{back}} \text{ and } \mathbf{V}_b^{\text{for}} \text{ are considered to be reliable} \\ \text{otherwise} \\ \quad \text{if } \text{MAD}_{\text{back}} < \text{MAD}_{\text{for}} \\ \quad \quad \text{The motion vector } \mathbf{V}_b^{\text{for}} \text{ is dropped} \\ \quad \text{otherwise} \\ \quad \quad \text{The motion vector } \mathbf{V}_b^{\text{back}} \text{ is dropped} \end{array} \right.$$

where T_b is a threshold.

- **Quad-tree Refinement:** The backward and forward motion vectors are obtained for each $BS_0 \times BS_0$ block. The objective of this step is to estimate the motion vectors for $BS_1 \times BS_1$ block ($BS_{i+1} = BS_i/2$) based on the obtained motion vectors of $BS_0 \times BS_0$ block in the previous step. Thus, each $BS_0 \times BS_0$ block is divided into four $BS_1 \times BS_1$ blocks. First, the four $BS_1 \times BS_1$ blocks inherit the motion vector of the $BS_0 \times BS_0$ block (see Fig. 7). Then, for each $BS_1 \times BS_1$ block b , the motion vectors of the neighboring blocks are taken into account to select the most accurate one. As shown in Fig. 7, one motion vector among the four different candidates \mathbf{v}_n ($n = 1, \dots, 4$) is selected for the block b according to:

$$\mathbf{v}_b = \arg \min_{\mathbf{v}_n} \text{MAD}(b, \mathbf{v}_n). \quad (3)$$

The motion vector for each $BS_1 \times BS_1$ block is computed using Eq. 3. Then, the $BS_1 \times BS_1$ block is split into four $BS_2 \times BS_2$ blocks. The motion vectors for $BS_2 \times BS_2$ blocks are computed using the same procedure. Finally, the same procedure can be repeated until obtaining the motion vectors for $BS_M \times BS_M$ blocks ($M > 1$).

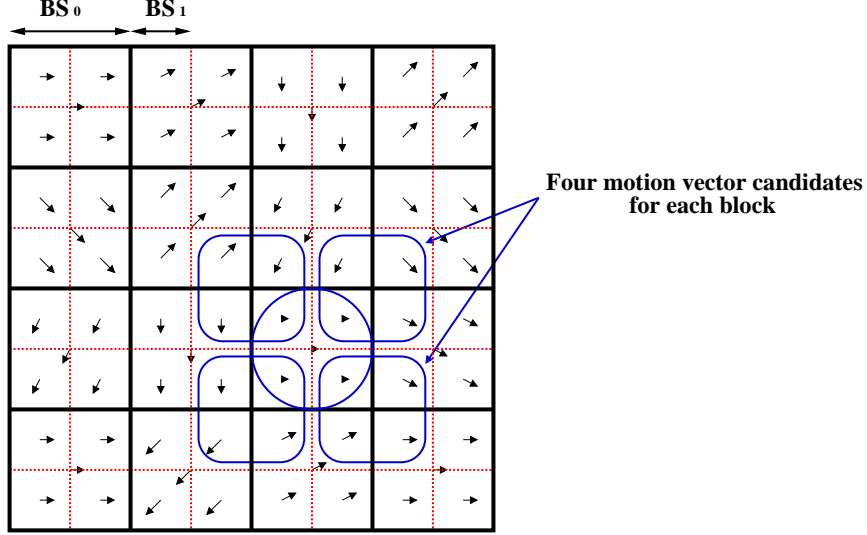


Figure 7: Motion vectors candidates for each $BS_1 \times BS_1$ block.

- **Selection of Motion Vectors:** This step consists in selecting the best motion vector $\tilde{\mathbf{v}}_b$ (backward or forward motion vector) for each block b ($BS_M \times BS_M$). Let MAD'_{back} and MAD'_{for} be the mean absolute difference corresponding to the motion vectors $\mathbf{V}_b^{\text{back}}$ and $\mathbf{V}_b^{\text{for}}$ respectively. The selection of the best motion vector is done as follows:

$$\left\{ \begin{array}{l} \text{if } MAD'_{\text{back}} < MAD'_{\text{for}} \\ \quad \tilde{\mathbf{v}}_b = \mathbf{V}_b^{\text{back}} \\ \text{otherwise} \\ \quad \tilde{\mathbf{v}}_b = -\mathbf{V}_b^{\text{for}} \end{array} \right.$$

- **Bi-directional Motion Estimation:** First, we aim at splitting the obtained motion vectors to estimate bi-directional motion vectors the blocks in WZF. For each block w in WZF, the distances between the center of the block w and the center of each obtained motion vector are computed. The closest motion vector to the block w is selected. Then, the selected motion vector is associated to the center of the block w , and divided by symmetry to obtain the bidirectional motion vector. Second, the bi-directional motion vectors are refined within a small search area \mathbf{S}' of $\pm SSR$ pixels in half-pixel accuracy. Let $(\mathbf{r}_b, -\mathbf{r}_b)$ be the bidirectional motion vector for the block b in WZF. A small displacement $\mathbf{d}_b^{\text{ref}}$ is added to the bidirectional motion vector during the refinement process. This displacement is obtained as follows:

$$\mathbf{d}_b^{\text{ref}} = \arg \min_{\mathbf{d} \in \mathbf{S}'} \text{MAD}(b, \mathbf{r}_b, \mathbf{d}), \quad (4)$$

with

$$\text{MAD}(b, \mathbf{r}, \mathbf{d}) = \frac{1}{L^2} \sum_{x=x_0}^{x_0+L} \sum_{y=y_0}^{y_0+L} |\text{BRF}(x + r_{bx} + d_x, y + r_{by} + d_y) - \text{FRF}(x - r_{bx} - d_x, y - r_{by} - d_y)| \quad (5)$$

where $\mathbf{d} \equiv (d_x, d_y)$ and $\mathbf{r}_b \equiv (r_{bx}, r_{by})$. Even though the size of the block is $BS_M \times BS_M$, an extended block of $L \times L$ ($L = BS_M + \text{EX}$) is used to compute the MAD in the refinement process.

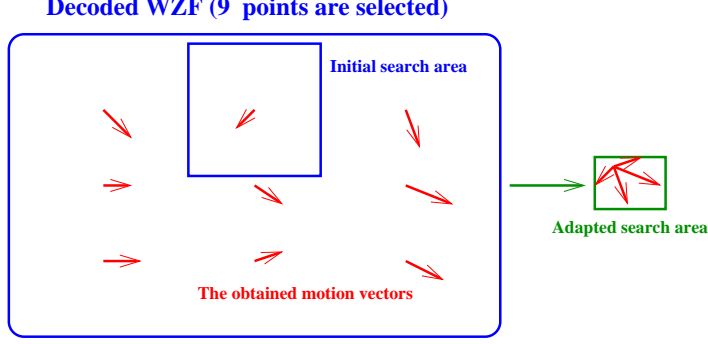


Figure 8: The motion vectors used to adapt the search area.

- **Spatial Smoothing:** The obtained bidirectional motion vectors may sometimes present low spatial coherence. The spatial smoothing algorithm aims at achieving higher motion field spatial coherence, by reducing the number of suspicious bidirectional motion vectors. For each block b , the spatial motion smoothing algorithm considers the neighboring bidirectional motion vectors as candidates for the block b . The weighted median vector field¹⁷ is used to select the bidirectional motion vector from the candidate bidirectional motion vectors as follows:

$$\mathbf{s}_b = \arg \min_{k=1,2,\dots,Nb} \left(\sum_{i=1}^{Nb} a_i \|\mathbf{r}_k - \mathbf{r}_i\| \right), \quad (6)$$

with

$$a_i = \frac{1}{L^2} \sum_{x=x_0}^{x_0+L} \sum_{y=y_0}^{y_0+L} |\text{BRF}(x + r_{ix}, y + r_{iy}) - \text{FRF}(x - r_{ix}, y - r_{iy})|, \quad (7)$$

Nb is the number of the neighboring blocks, and $L = BS_M + EX$.

- **Bi-directional Motion Compensation:** Once the final bidirectional motion vectors are estimated, the SI can be interpolated using bidirectional motion compensation as follows:

$$\text{SI}(\mathbf{p}) = \frac{1}{2} (\text{BRF}(\mathbf{p} + \mathbf{s}_b) + \text{FRF}(\mathbf{p} - \mathbf{s}_b)), \quad (8)$$

where \mathbf{s}_b and $-\mathbf{s}_b$ are the bidirectional motion vectors, associated to the position $\mathbf{p} = (x, y)$ toward the BRF and FRF respectively.

3.2 Part II - Side Information Refinement

We now propose a new method that consists in improving the decoded WZFs, already obtained by DISCOVER or by the method proposed by Abou-Elailah,¹³ when the GOP size is larger than 2. This improvement is achieved by re-estimating the SI using the neighboring decoded frames, with an adaptive search area and a variable block size.

In the method proposed by Petrazzuoli et al.,¹¹ the central WZF I_k is re-estimated without using the available decoded \hat{I}_k . In this paper, we re-estimate I_k using the already decoded WZF \hat{I}_k , along with the neighboring decoded frames. In particular, we propose an approach to adapt the search area to the real motion between the decoded WZF \hat{I}_k and the previous (or next) decoded frame. This procedure achieves a better estimation for high motion regions. Moreover, it generates more homogeneous motion vectors and reduces the estimation complexity for slow motion regions. First, nine large blocks are selected (using uniform sampling) in the decoded WZF. The matching for those blocks is performed in order to determine the corresponding blocks in the previous (or next) decoded frame. When a selected block belongs to a homogeneous region,

SI Average PSNR [dB]						
sequence	Stefan	Foreman	Bus	Coastguard	Soccer	Hall
GOP = 2						
MCTI	22.57	29.31	24.72	31.43	22.05	35.66
SIG	23.83	29.97	27.14	32.35	22.75	36.22
GOP = 4						
MCTI	21.28	27.58	23.48	29.85	20.81	34.51
SIG	22.24	28.10	25.68	30.75	21.42	35.03
GOP = 8						
MCTI	20.64	26.24	22.53	28.75	20.15	33.69
SIG	21.47	26.69	24.61	29.59	20.70	34.04

Table 1: Average PSNR of the SI obtained with the proposed method and the MCTI technique.

the MAD is almost the same for all candidate blocks in this region. In order to avoid obtaining false large motion vectors in these homogeneous areas, the MAD computed during the matching procedure is penalized (MAD_{pen}) by the length of the motion vector $\mathbf{m} = (m_x, m_y)$ using:

$$MAD_{pen} = MAD \times \left(1 + \lambda \sqrt{m_x^2 + m_y^2}\right), \quad (9)$$

as was done in equation 1. Here, λ is empirically set to 0.02.

Thus, nine motion vectors associated to those blocks are obtained. These motion vectors are used to adapt the initial search area. In this paper, we define a search area by four parameters SA_r , SA_l , SA_t and SA_b , which represent the distance between the center and the right, left, top, and bottom points, respectively, attained by the search area. The initial search area is set to ± 32 pixels (in this case, $SA_r = SA_l = SA_t = SA_b = 32$). These parameters are adapted according to the obtained motion vectors $\mathbf{m}_i = (m_{ix}, m_{iy})$ ($i = 1, 2, \dots, 9$) as follows:

$$\begin{cases} SA_r &= \max_i(m_{ix}), \text{ if } m_{ix} > 0 \\ SA_l &= -\min_i(m_{ix}), \text{ if } m_{ix} < 0 \\ SA_t &= \max_i(m_{iy}), \text{ if } m_{iy} > 0 \\ SA_b &= -\min_i(m_{iy}), \text{ if } m_{iy} < 0 \end{cases}$$

The obtained parameters are used to construct the adapted search area (see Fig. 8).

We also introduce a variable block size in the re-estimation procedure. We start by dividing the frame I_k into 16×16 blocks. Then, the block matching is carried out, based on the previous and next decoded frames, using the adapted search area. If the obtained MAD is greater than a threshold, this block is divided into 8×8 blocks, and the motion vectors are computed for these four 8×8 blocks. The procedure is stopped at a block size of 4×4 pixels. Finally, the bidirectional motion compensation is applied to obtain the new SI.

4. EXPERIMENTAL RESULTS

With the aim of evaluating the performance of the proposed methods, we perform extensive simulations, adopting the same test conditions as described in DISCOVER¹⁴, *i.e.* test video sequences are at QCIF spatial resolution and sampled at 15 frames/sec.

4.1 Part I - Side Information Generation (SIG)

The parameters of the proposed method are set as follows in the experiments: $BS_0 = 32$ pixels, $BS_M = 4$ pixels, $SA_0 = 48$ pixels, $SA_1 = 3$ pixels, $SSR = 1.5$ pixels, $N_0 = 2$ pixels, $N_1 = 1$ pixel, and $EX = 4$ pixels. The results of the proposed method are compared to DISCOVER codec.

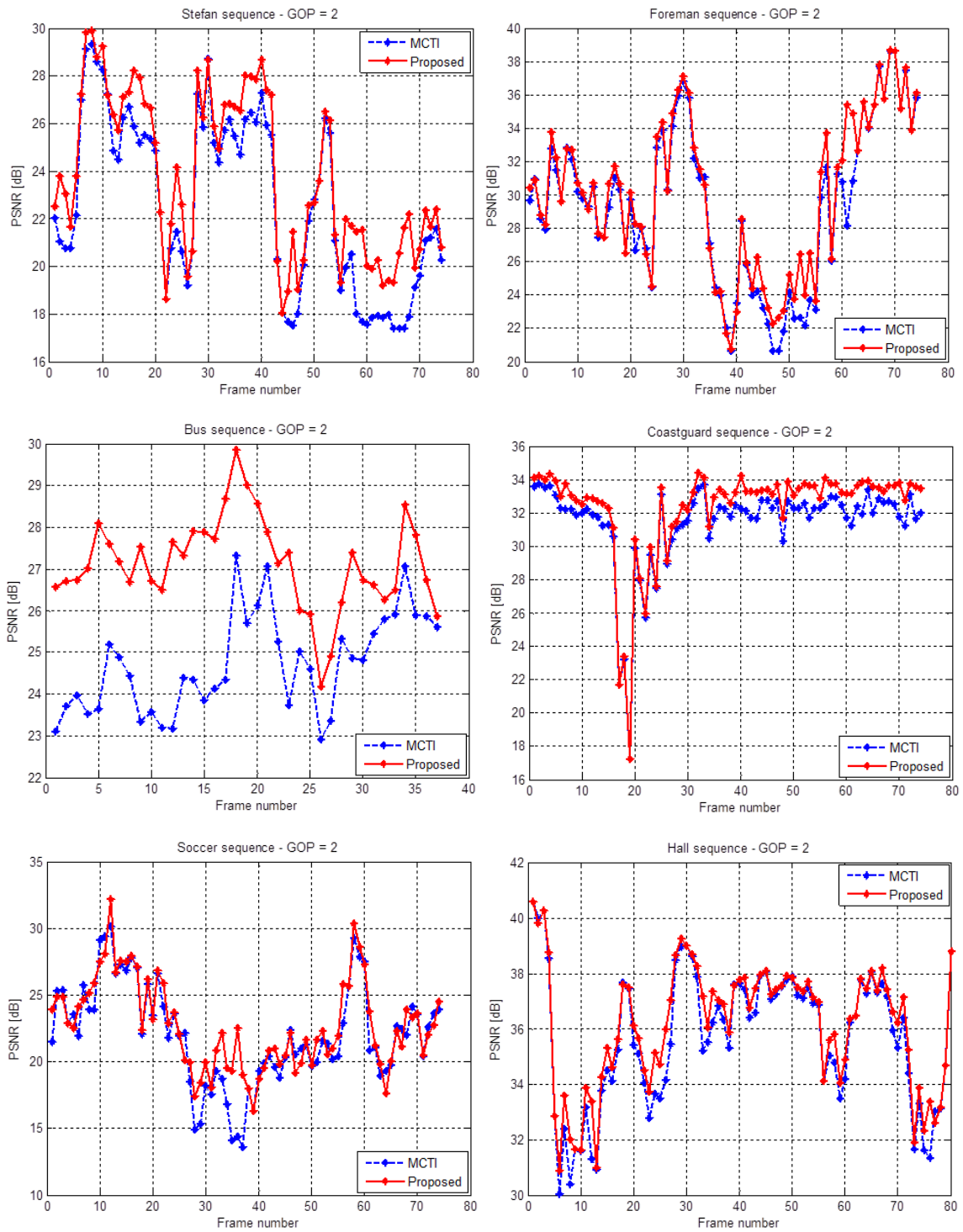


Figure 9: PSNR of the proposed SI generation (SIG) and the MCTI SI generation for Stefan, Foreman, Bus, Coastguard, Soccer and Hall sequences for a GOP size of 2.

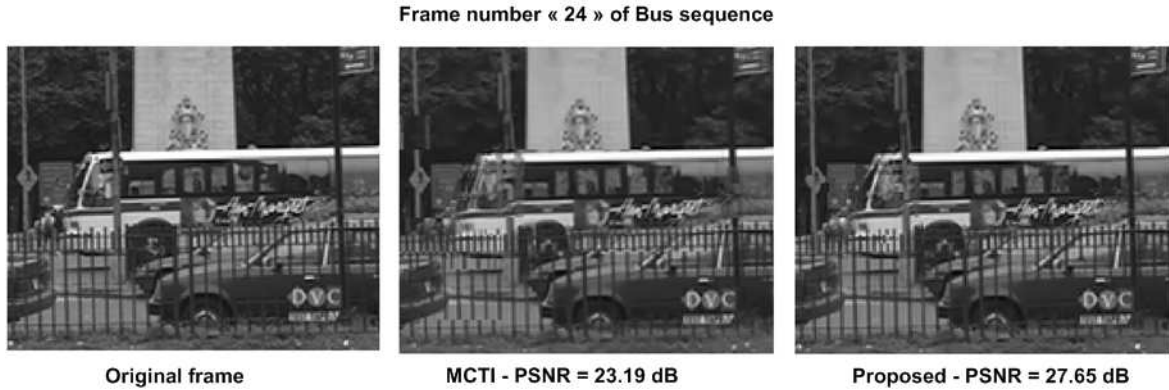


Figure 10: Visual result of the SI generated by the proposed method and the MCTI technique, for frame number 24 of Bus sequence, with a GOP size of 2.

sequence	Stefan	Foreman	Bus	Coastguard	Soccer	Hall
GOP = 2						
Δ_R [%]	-8.32	-3.17	-13.50	-4.22	-4.95	-1.55
Δ_{PSNR} [dB]	0.50	0.17	0.79	0.21	0.27	0.12
GOP = 4						
Δ_R [%]	-9.28	-2.66	-21.16	-11.70	-6.28	-2.76
Δ_{PSNR} [dB]	0.54	0.14	1.24	0.47	0.34	0.19
GOP = 8						
Δ_R [%]	-9.06	-3.58	-22.10	-16.93	-5.67	-6.07
Δ_{PSNR} [dB]	0.54	0.16	1.31	0.71	0.33	0.32

Table 2: RD performance gain for Stefan, Foreman, Bus, Coastguard, Soccer and Hall sequences towards DISCOVER codec, using Bjontegaard metric.

4.1.1 SI performance assessment

The quality of the generated SI, estimated in terms of the PSNR obtained with the proposed method and with the MCTI technique, is shown in Fig. 9, for Stefan, Foreman, Bus, Coastguard, Soccer and Hall sequences, for a GOP size of 2. As shown in this figure, the proposed method consistently achieves a gain compared to MCTI technique.

The average PSNR of the SI is shown in Table 1, for the proposed method and the MCTI technique, for all sequences and different GOP sizes. A significant gain is observed with the proposed method for all test sequences and all GOP sizes. The gain reaches 1.26 dB and 2.42 dB for Stefan and Bus sequences respectively, for a GOP size of 2.

Fig. 10 shows the visual result of the SI estimated by the proposed and the MCTI techniques, compared to the original frame, for frame number 24 of Bus sequence, with a GOP size of 2. The SI generated using the MCTI technique contains many block artifacts (PSNR = 23.19 dB), whereas the proposed method presents a much better quality, with a gain of 4.45 dB compared to MCTI.

4.1.2 Rate-Distortion performance

Fig. 11 shows the RD performance of the proposed method and DISCOVER codec for Stefan, Foreman, Bus, Coastguard, Soccer and Hall sequences, for all GOP sizes. The proposed method clearly outperforms the MCTI technique.

In Table 2, we show the RD performance of the proposed method compared to DISCOVER codec, using the Bjontegaard metric.¹⁸ At the decoder side, the computational complexity is increased with respect to the MCTI technique: in particular, two motion estimations are needed for the first step. This is justified, as the proposed method achieves a significant rate reduction. For instance, for Bus sequence, we reach a PSNR improvement of 1.31 dB and a bit reduction of 22.1% w.r.t MCTI, for a GOP size equal to 8.

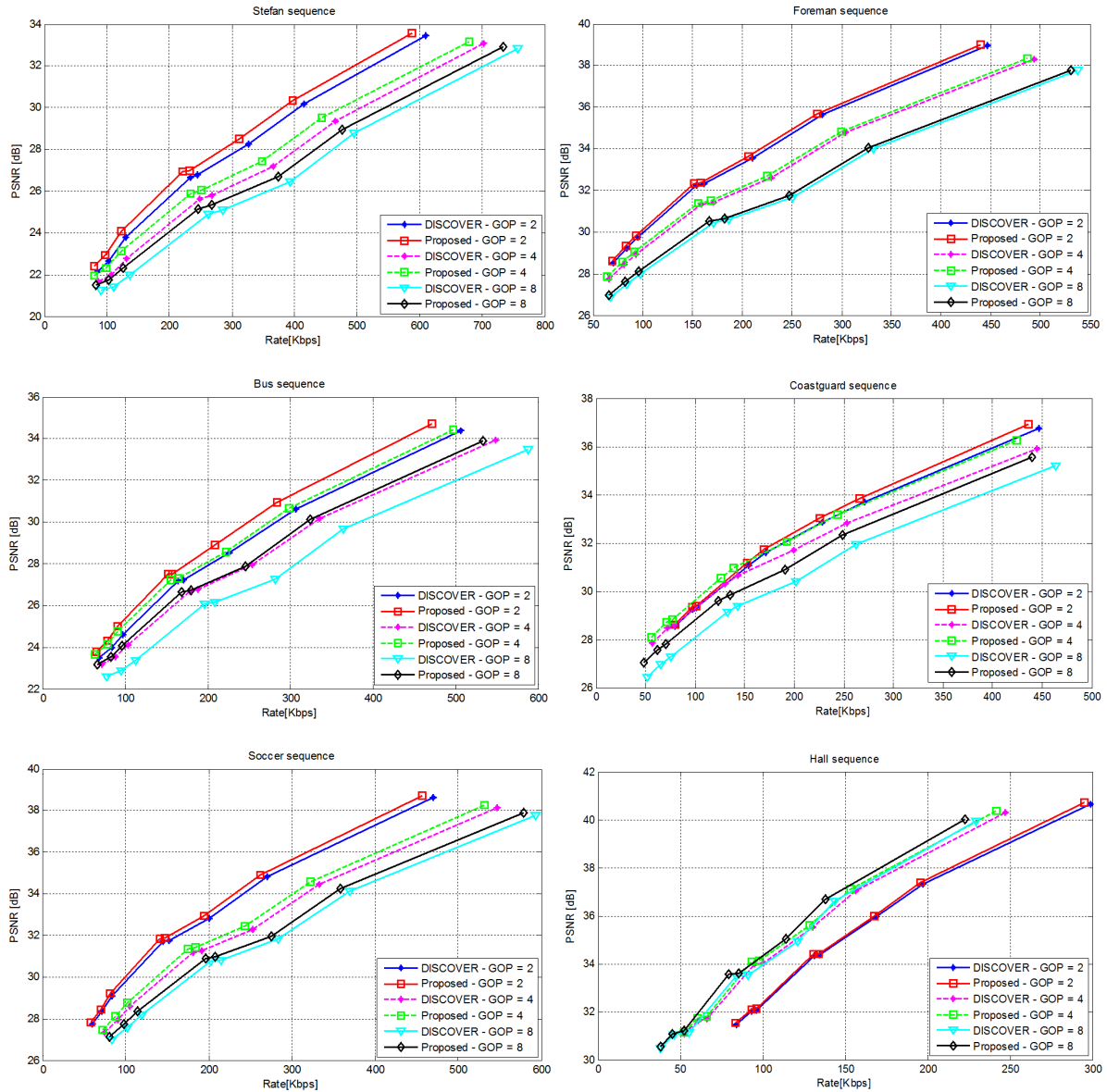


Figure 11: RD performance comparison between the proposed method SIG and DISCOVER codec for Stefan, Foreman, Bus, Coastguard, Soccer and Hall sequences for all GOP sizes.

sequence	Final SI Average PSNR [dB]					
	Stefan	Foreman	Bus	Coastguard	Soccer	Hall
DISCOVER	20.64	26.24	22.53	28.75	20.15	33.69
Petrazzuoli et al. ¹¹	22.54	29.12	24.80	31.24	22.01	35.76
PropA	28.69	34.86	28.66	32.39	30.55	37.56
Abou-Elailah et al. ¹³	25.65	32.94	26.72	30.89	28.14	35.47
PropB	28.91	35.14	28.75	32.54	30.60	37.65

Table 3: Final SI average PSNR for a GOP size equal to 8 (QI = 8).

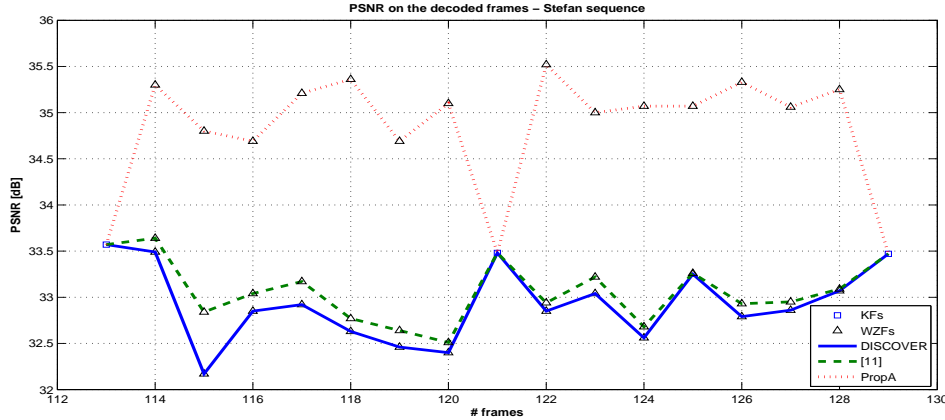


Figure 12: PSNR of the decoded frames for two GOPs beginning at frame number 113 of the *Stefan* sequence for a GOP size = 8.

4.2 Part II - Side Information Refinement

The proposed SI refinement method is applied in two cases. When the enhancement is applied on the decoded WZFs obtained by DISCOVER codec, the method is called PropA, and PropB when the improvement is carried out on the decoded WZFs obtained by the algorithm proposed by Abou-Elailah et al.¹³ Both methods are compared w.r.t. DISCOVER codec. Also the results obtained by Petrazzuoli et al.¹¹ and by Abou-Elailah et al.¹³ are shown in order to have a complete comparative analysis of the methods.

4.2.1 SI performance assessment

Table 3 shows the average PSNR of the final SI, obtained after the refinement process, with the different methods. The proposed techniques lead to a significant improvement in the SI quality for all test sequences. Both proposed PropA and PropB methods show very significant gains compared to DISCOVER, to the method proposed by Petrazzuoli et al.¹¹ and to the one proposed by Abou-Elailah et al.¹³

4.2.2 Rate-Distortion performance

Figure 12 shows the PSNR of the decoded frames of Stefan sequence for DISCOVER codec, the method proposed by Petrazzuoli et al.¹¹, and the proposed technique PropA. We can see that the proposed method achieves a significant gain compared to both DISCOVER and Petrazzuoli et al.¹¹.

The RD performance is shown for the Stefan, Foreman, Bus, Coastguard, Soccer and Hall sequences in Table 4, in comparison to the DISCOVER codec, using the Bjontegaard metric¹⁸, for a GOP size equal to 4 and 8. We represent the performance of the method proposed by Petrazzuoli et al.¹¹, the proposed method PropA, the method proposed by Abou-Elailah et al.¹³, the proposed method PropB, H.264/AVC No motion and H.264/AVC Intra.

The proposed method PropA, as well as the technique by Petrazzuoli et al.¹¹, consists in a refinement of the frames decoded by DISCOVER, while PropB is a refinement of the frames obtained by the method proposed by Abou-Elailah et al.¹³ It can be observed that PropA and PropB always achieve a gain w.r.t. the previous works^{11,13} especially for sequences containing high motion.

sequence	Stefan	Foreman	Bus	Coastguard	Soccer	Hall
GOP = 4						
Petrazzuoli et al. ¹¹						
Δ_R [%]	-4.32	-5.92	-5.50	-4.68	-2.92	-0.84
Δ_{PSNR} [dB]	0.23	0.28	0.29	0.20	0.14	0.05
PropA						
Δ_R [%]	-22.79	-23.28	-13.17	-9.13	-19.65	-3.96
Δ_{PSNR} [dB]	1.32	1.19	0.68	0.38	0.96	0.24
Abou-Elailah et al. ¹³						
Δ_R [%]	-30.36	-35.96	-19.95	-13.49	-27.89	-4.72
Δ_{PSNR} [dB]	1.98	2.21	1.12	0.56	1.62	0.30
PropB						
Δ_R [%]	-37.95	-40.33	-23.67	-16.67	-33.13	-6.17
Δ_{PSNR} [dB]	2.54	2.49	1.33	0.71	1.92	0.38
H.264/AVC No Motion						
Δ_R [%]	-33.67	-37.88	-22.79	10.10	-54.92	-40.82
Δ_{PSNR} [dB]	2.38	2.38	1.53	-0.61	5.53	3.23
H.264/AVC Intra						
Δ_R [%]	-32.59	-22.31	-24.63	19.99	-56.48	79.51
Δ_{PSNR} [dB]	2.14	1.04	1.17	-1.18	4.79	-5.02
GOP = 8						
Petrazzuoli et al. ¹¹						
Δ_R [%]	-8.38	-12.43	-12.15	-9.51	-6.57	-2.20
Δ_{PSNR} [dB]	0.44	0.56	0.58	0.40	0.28	0.08
PropA						
Δ_R [%]	-27.92	-30.72	-20.12	-15.50	-24.49	-6.79
Δ_{PSNR} [dB]	1.60	1.55	1.00	0.65	1.18	0.31
Abou-Elailah et al. ¹³						
Δ_R [%]	-37.77	-47.62	-28.91	-27.14	-32.47	-12.77
Δ_{PSNR} [dB]	2.53	3.04	1.65	1.20	1.89	0.69
PropB						
Δ_R [%]	-47.33	-53.88	-35.73	-33.09	-39.97	-15.75
Δ_{PSNR} [dB]	3.32	3.53	2.08	1.50	2.36	0.85
H.264/AVC No Motion						
Δ_R [%]	-43.79	-47.56	-37.83	-9.04	-62.09	-58.26
Δ_{PSNR} [dB]	3.33	3.34	2.62	0.12	7.18	4.57
H.264/AVC Intra						
Δ_R [%]	-45.19	-42.06	-41.76	-14.30	-64.91	77.42
Δ_{PSNR} [dB]	3.21	2.39	2.31	-0.01	6.36	-5.47

Table 4: Rate distortion performance comparison for a GOP size equal to 4 and 8, w.r.t. DISCOVER codec, using Bjontegaard metric.

The gains become even more significant for a GOP size equal to 8. In fact, for PropA, we obtain a bit reduction up to -30.72% , which corresponds to an improvement of 1.55 dB on the decoded frames w.r.t. DISCOVER codec. The proposed method PropB allows a significant gain of up to 3.53 dB, with a rate reduction of 53.88%, compared to the DISCOVER codec. These maxima of performance gain are obtained for the Foreman sequence.

The performance of the proposed method PropB is superior to that of H.264/AVC Intra for all test sequences, except for the Soccer sequence with a GOP size of 4, and Bus and Soccer sequences with a GOP size of 8. The proposed method can beat H.264/AVC No motion for Coastguard and Foreman sequences.

5. CONCLUSIONS

In this paper, we propose two new methods for SI improvement in transform-domain DVC. The first method consists in generating an initial SI using backward and forward motion estimation. This new approach allows a gain of up to 1.31 dB compared to DISCOVER codec. In the second method, we propose a new approach for SI refinement for long duration GOPs. Each decoded WZF is used to adapt the initial motion search area. The adaptive search area is then used in order to re-estimate the SI, along with the previous and next decoded frames, using a variable block size. Experimental results show that our proposed method can achieve a gain in RD performance of up to 0.79 dB for a GOP size of 8, compared to the method proposed by Abou-Elailah et al.,¹³ 2.97 dB compared to the one proposed by Petrazzuoli et al.,¹¹ and 3.53 dB compared to DISCOVER codec, especially when the video sequence contains high motion.

REFERENCES

- [1] Slepian, D. and Wolf, J. K., “Noiseless coding of correlated information sources,” *IEEE Trans. Inform. Theory* **19**, 471–480 (July 1973).
- [2] Wyner, A. and Ziv, J., “The rate-distortion function for source coding with side information at the receiver,” *IEEE Trans. Inform. Theory* **22**, 1–11 (Jan. 1976).
- [3] Guillemot, C., Pereira, F., Torres, L., Ebrahimi, T., Leonardi, R., and Ostermann, J., “Distributed monoview and multiview video coding: Basics, problems and recent advances,” *IEEE Signal Processing Mag.* , 67–76 (Sept. 2007).
- [4] Girod, B., Aaron, A., Rane, S., and Rebollo-Monedero, D., “Distributed video coding,” *Proc. IEEE* **93**, 71–83 (Jan. 2005).
- [5] Aaron, A., Setton, E., and Girod, B., “Towards practical Wyner-Ziv coding of video,” in [*IEEE International Conf. on Image Processing*], (2003).
- [6] Ascenso, J., Brites, C., and Pereira, F., “Content adaptive Wyner-Ziv video coding driven by motion activity,” in [*IEEE International Conf. on Image Processing*], (2006).
- [7] Yaacoub, C., Farah, J., and Pesquet-Popescu, B., “New adaptive algorithms for GOP size control with return channel suppression in Wyner-Ziv video coding,” *International Journal of Digital Multimedia Broadcasting* (2009).
- [8] Abou-Elailah, A., Dufaux, F., Cagnazzo, M., Pesquet-Popescu, B., and Farah, J., “Fusion of global and local motion estimation for distributed video coding,” *IEEE Transactions on Circuits and Systems for Video Technology (In press)* .
- [9] Abou-Elailah, A., Dufaux, F., Farah, J., and Cagnazzo, M., “Fusion of global and local side information using support vector machine in distributed video coding,” in [*20th European Signal Processing Conference (EUSIPCO)*], (Aug. 2012).
- [10] Petrazzuoli, G., Cagnazzo, M., and Pesquet-Popescu, B., “High order motion interpolation for side information improvement in DVC,” in [*IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*], 2342 – 2345 (June 2010).
- [11] Petrazzuoli, G., Maugey, T., Cagnazzo, M., and Pesquet-Popescu, B., “Side information refinement for long duration GOPs in DVC,” in [*IEEE Workshop on Multimedia Signal Processing*], **1** (2010).
- [12] Brites, C., Ascenso, J., and Pereira, F., “Improving transform domain Wyner-Ziv video coding performance,” in [*Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*], **2**, 525–528 (May 2006).
- [13] Abou-Elailah, A., Farah, J., Cagnazzo, M., Pesquet-Popescu, B., and Dufaux, F., “Improved side information for distributed video coding,” in [*3rd European Workshop on Visual Information Processing (EUVIP)*], 42 – 49 (July 2011).
- [14] Artigas, X., Ascenso, J., Dalai, M., Klomp, S., Kubasov, D., and Oualet, M., “The DISCOVER codec: Architecture, techniques and evaluation, picture coding symposium,” in [*Picture Coding Symposium (PCS)*], (2007).
- [15] Kubasov, D., Nayak, J., and Guillemot, C., “Optimal reconstruction in wyner-ziv video coding with multiple side information,” in [*IEEE 9th Workshop on Multimedia Signal Processing (MMSP)*], 183–186 (2007).
- [16] Ascenso, J., Brites, C., and Pereira, F., “Improving frame interpolation with spatial motion smoothing for pixel domain distributed video coding,” in [*5th EURASIP Conference on Speech and Image Processing, Multimedia Communications and Services*], (July 2005).
- [17] Alparone, J., Barni, M., Bartolini, F., and Cappellini, V., “Adaptively weighted vector-median filters for motion fields smoothing,” in [*IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*], **4**, 2267–2270 (May 1996).
- [18] Bjontegaard, G., “Calculation of average PSNR differences between RD-curves,” in [*VCEG Meeting*], (Apr. 2001).