Fusion of Global and Local Motion Estimation Using Foreground Objects for Distributed Video Coding

Abdalbassir Abou-Elailah, Frederic Dufaux, Joumana Farah, Marco Cagnazzo, Srivastava Anuj, Béatrice Pesquet-Popescu

Fusion of global and local motion estimation using foreground objects for Distributed Video Coding

Abdalbassir ABOU-ELAILAH, Frederic DUFAUX, Joumana FARAH,
Marco CAGNAZZO, Anuj Srivastava, and Beatrice PESQUET-POPESCU

Abstract

The side information in distributed video coding is estimated using the available decoded frames, and exploited for the decoding and reconstruction of other frames. The quality of the side information has a strong impact on the performance of distributed video coding. Here we propose a new approach that combines both global and local side information to improve coding performance. Since the background pixels in a frame are assigned to global estimation and the foreground objects to local estimation, one needs to estimate foreground objects in the side information using the backward and forward foreground objects, The background pixels are directly taken from the global side information. Specifically, elastic curves and local motion compensation are used to generate the foreground objects masks in the side information. Experimental results show that, as far as the rate-distortion performance is concerned, the proposed approach can achieve a PSNR improvement of up to 1.39 dB for a GOP size of 2, and up to 4.73 dB for larger GOP sizes, with respect to the reference DISCOVER codec.

Index Terms

A. ABOU-ELAILAH, F. DUFAUX, M. CAGNAZZO, and B. PESQUET-POPESCU are with the Signal and Image Processing Department, Institut T*él*com - TELECOM Paristech, 46 rue Barrault, F - 75634 Paris Cedex 13, FRANCE, e-mail: ({elailah, frederic.dufaux, marco.cagnazzo, beatrice.pesquet }@telecom-paristech.fr).

J. FARAH is with the Department of Telecommunications Engineering, Faculty of Engineering, Holy-Spirit University of Kaslik, P.O. Box 446, Jounieh, Lebanon, e-mail: joumanafarah@usek.edu.lb.

A. Srivastava is with the Department of Statistics, Florida State University, 600 W College Ave, Tallahassee, FL 32306, United States, e-mail: anuj@stat.fsu.edu

Distributed Video Coding, Wyner-Ziv Frames, Key Frames, Side Information, Global Estimation, Local Estimation, Elastic Curves, Foreground Objects, Rate-Distortion Performance.

I. INTRODUCTION

The digital video coding standards ISO/IEC MPEG-x and ITU-T H.26x are mainly based on the Discrete Cosine Transform (DCT) and inter-frame, intra-frame predictive coding. Additionally, in the High Efficiency Video Coding (HEVC) international standard, that has recently emerged as a successor to H.264/AVC, the encoder exploits the spatial and temporal redundancies existing in a video sequence. Here the encoder is significantly more complex than the decoder (with a typical factor of 5 to 10 [1]) and its architecture is well-suited for applications where the video sequence is encoded once and decoded many times, such as in broadcasting or video streaming.

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

Distributed Video Coding (DVC) is a recent paradigm in video communication that fits well in these scenarios, since it enables the exploitation of the similarities among successive frames at the decoder side, making the encoder less complex. Consequently, the complex tasks of motion estimation and compensation are shifted to the decoder. Note that the Slepian-Wolf theorem from information theory [2] states that for a lossless compression it is possible to encode correlated sources (let us call them X and Y) independently and decode them jointly, while achieving the same rate bounds that can be attained in the case of joint encoding and decoding. The case of lossy compression was subsequently dealt with by Wyner and Ziv [3]. Their popular result states that, under mild constraints, the theoretical rate-distortion bounds for distributed coding are the same as those for joint coding, provided that joint decoding is possible.

Based on these theoretical results some practical implementations of DVC have been proposed in [4], [5]. The European project DISCOVER [6], [7] resulted in one of the most efficient and popular existing architectures, where the DISCOVER codec is based on the Stanford scheme [5]. More specifically, the sequence images are split into two sets of frames: key frames (KFs) and Wyner-Ziv frames (WZFs). The Group of Pictures (GOP) of size n is defined as a set of frames

consisting of one KF and $n-1$ WZFs. The KFs are independently encoded and decoded using such Intra-coding techniques as H.264/AVC Intra mode or JPEG2000. The WZFs are separately transformed and quantized, and a systematic channel code is applied to the resulting coefficients. Only the parity bits are kept and sent to the decoder upon request. This can be seen as a Slepian-Wolf coder applied to the quantized transform coefficients. At the decoder, the reconstructed reference frames are used to compute the side information (SI), which is an estimation of the WZF being decoded. The Motion-Compensated Temporal Interpolation (MCTI) [8] is commonly used to produce SI. Finally, a channel decoder uses the parity information to correct SI, thus reconstructing the WZF. Therefore, generating an accurate SI is essential, since it would result in a reduced amount of parity information requested by the decoder through the return channel. At the same time the quality of the decoded WZF would be improved during reconstruction.

The goal in terms of compression efficiency is to achieve a coding performance similar to the best available hybrid video coding schemes. However, DVC has not reached the performance level of classical inter-frame coding yet. This is in part due to the quality of SI which has a strong impact on the final Rate-Distortion (RD) performance.

In this paper we propose new methods to enhance SI through a combination of the global and local motion estimations. The parameters of the global model are estimated at the encoder, and sent to the decoder in order to generate a SI based on Global Motion Compensation (GMC), and referred to as GMC SI. On the other hand, another SI is estimated using the MCTI technique (local motion compensation) with spatial motion smoothing, exactly as in DISCOVER codec; this SI is referred to as MCTI SI. Thus, the two estimations MCTI SI and GMC SI are generated at the decoder, using the reference frames and the global parameters.

Normally, the background pixels must be compensated using the global motion and the foreground objects using the local motion. However, the traditional motion compensation uses block-based algorithms, resulting in possible coding artifacts above all around object edges. We propose, therefore, to resort to segmentation maps in order to discriminate the background and the foreground, and to apply to each one the suitable motion model. We underline here that we are not proposing a segmentation tool, but rather a coding algorithm that is able to efficiently exploit the information provided by the segmentation. More precisely, we are able to accurately infer the segmentation maps of the WZFs given the segmentation maps of the KFs, thanks to the elastic deformation of object contours. This is the main contribution of this article. In this context,

our method could be referred as ”ideal” since we use manual segmentation maps. However, in order to validate our technique in a more realistic scenario, we also provide the experimental results using an actual yet simple automatic segmentation algorithm, showing promising results even without ideal maps.

First, we propose a new method based on elastic shape analysis of curves [9], [10] for estimating the foreground objects masks in the previously-estimated SI. Then, the pixels in the estimated masks are selected from MCTI SI, while GMC SI is used to cover all the remaining pixels in the estimated SI. More specifically, the foreground objects masks are generated using the segmented foreground objects in the reference frames. Then, the foreground objects contours are constructed from the generated masks. Furthermore, the contours are considered as closed curves and the algorithm in [10] is used to generate the curves in the estimated SI using curves from the reference frames. Finally, the objects masks are generated using these generated curves. We observe that while elastic deformations have been used earlier, the original applications were in shape analysis, face recognition, shape probabilistic models, and shape inference for pose modification. The use of elastic deformations for predicting the temporal, motion-related deformation of object boundaries is novel to this paper.

We propose two different approaches for generating foreground objects in SI, based on the local motion-compensation. In the first approach, the MCTI technique is directly applied to the backward and forward foreground objects, in order to generate the foreground objects in SI. In the second approach, a local motion estimation method is proposed to generate foreground objects in SI exploiting the backward and forward foreground objects. Here we use a local motion-estimation technique which a variation of the classical one used in Discover. The details of this method will be discussed in Sec. III-C2.

Next, a mask is generated using the estimated foreground objects in SI. Based on the mask, two approaches are proposed to combine global and local motion estimations. The first one aims at directly using the estimated foreground objects and GMC SI. The second one consists of using MCTI SI for the pixels in the object mask and GMC SI for the remaining pixels.

We clarify that the proposed technique allows to efficiently use a contour predictor in the context of compression; moreover, as we will show in the experimental section, the achieved gains are relatively immune to the segmentation process. This is partly due to the fact that the contours are estimated at the decoder and need not to be transmitted. As a consequence they

can be irregular without greatly impacting the compression performance. This is in contrast to the classical object-based compression techniques where a non-ideal segmentation, or even an ideal segmentation with complex contours, is one of the main reasons for the inferior compression performance with respect to block-based coding [11]. In other words, our method is a contour-based compression technique that consistently outperforms the block-based state-of-the-art algorithms, and this holds even when the segmentation produces imperfect or complex contours. Finally, we note that the additional complexity related to the computation of the elastic curve affects only the decoder. This perfectly fits the DVC paradigm.

The rest of this paper is structured as follows. The related work is described in Section II. Specifically, DISCOVER codec is presented in Section II-A, generation of the global SI is described is Section II-B, and relevant SI improvement techniques are presented in Section II-C. The proposed methods for the fusion of global and local motion estimations are described in Section III. More specifically, the removal of artifacts affecting the GMC SI is described in Section III-A, fusion using elastic curves in Section III-B, fusion using local motion compensation in Section III-C, and the oracle fusion in Section III-D. Experimental results are shown in Section IV in order to evaluate and compare the RD performance of the proposed approaches. Finally, conclusions and future work are presented in Section V.

II. Related work

A. DISCOVER Architecture

We start with a brief presentation of the DISCOVER codec [6], [7]. Here the input video sequence is divided into WZFs and KFs, and the latter are encoded using H.264/AVC Intra coding. The WZF encoding and decoding procedures are described below.

- **Wyner-Ziv encoder -** At the encoder side, the WZF is first transformed using a 4×4 integer Discrete Cosine Transform (DCT). The integer DCT coefficients of the whole WZF are then organized into 16 bands. Next, each integer DCT coefficient is uniformly quantized. The resulting quantized symbols are split into bit planes, which are then independently encoded using a rate-compatible Low-Density Parity Check Accumulate (LDPCA) code. The parity information is stored in a buffer and progressively sent (upon request) to the decoder, while the systematic bits are discarded.

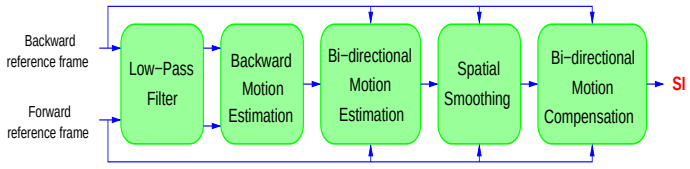

Fig. 1. MCTI technique [8].

- **Generation of side information** - In the DISCOVER scheme, the MCTI technique [8] is used to generate SI at the decoder side. Fig. 1 shows the architecture of the MCTI technique. The frame interpolation framework is composed of four modules to obtain high quality SI as follows: Both reference frames are first low-pass filtered in order to improve the motion vector reliability, followed by backward motion estimation between the backward and forward 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, and finally bi-directional motion compensation.
- **Wyner-Ziv decoder** - A block-based 4×4 integer DCT is carried out over the generated SI in order to obtain the integer DCT coefficients. Then, the LDPCA decoder corrects the bit errors in the DCT transformed SI, using the parity bits of WZF requested from the encoder through the feedback channel.
- **Reconstruction and inverse transform** - The reconstruction corresponds to the inverse of the quantization using SI DCT coefficients and the decoded Wyner-Ziv DCT coefficients. After that, the inverse 4×4 integer DCT transform is carried out, and the entire frame is restored in the pixel domain.

B. Global Motion Compensation

In [12] a new approach for generating GMC SI is proposed. Here, we give the main characteristics of this technique: First, the feature points of the original WZ and reference frames are extracted, at the encoder, using Scale Invariant Feature Transform (SIFT). Then, a matching between the feature points is carried out. Second, an efficient algorithm is proposed to estimate the affine parameters between the WZF and the backward (and forward) reference frame. Let T_B and T_F be the affine transforms between the original WZF and the backward and forward original reference frames, respectively. The parameters of those transforms are encoded and sent

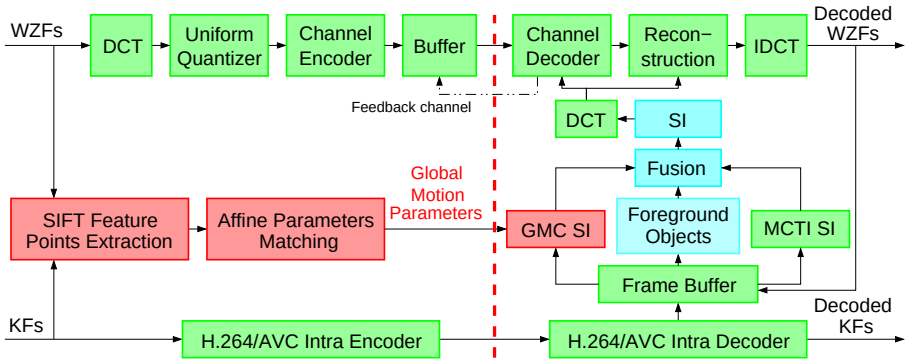

Fig. 2. Overall structure of the proposed DVC codec.

to the decoder.

Let us denote the backward and forward reference frames respectively as R_B and R_F for short. Moreover, we indicate with $\hat{R}_B$ and $\hat{R}_F$ the results of GMC transforms T_B and T_F applied to R_B and R_F. The GMC SI is simply defined as the average of the frames $\hat{R}_B$ and $\hat{R}_F$.

Consequently, we have now two SI frames (MCTI SI and GMC SI) for the current WZF, therefore a fusion technique is needed. In [12] we proposed an algorithm for the fusion, based on the residual of the compensated reference frames. Let $\tilde{R}_B$ and $\tilde{R}_F$ be the backward and forward compensated reference frames estimated by MCTI technique. For each 4×4 block b, we perform a fusion by observing pixels in a 8×8 window. Namely, we compute two sums of absolute differences (SADs), f_{GMC} and f_{MCTI}:

$$\begin{aligned} f_{\text{GMC}} &= \sum_{i=-4}^{3} \sum_{j=-4}^{3} |\hat{R}_F(X_i, Y_j) - \hat{R}_B(X_i, Y_j)| \\ f_{\text{MCTI}} &= \sum_{i=-4}^{3} \sum_{j=-4}^{3} |\tilde{R}_F(X_i, Y_j) - \tilde{R}_B(X_i, Y_j)| \end{aligned} \tag{1}$$

Here $(X_i, Y_j) = (x_0 + i, y_0 + j)$, and (x_0, y_0) is the coordinate of the center pixel of the current block b. The fusion in [12] is then given by:

$$\text{SI}(b) = \begin{cases} \text{GMC SI} & \text{if } f_{\text{GMC}} < f_{\text{MCTI}} \\ \text{MCTI SI} & \text{otherwise} \end{cases} \tag{2}$$

Hereafter, we refer to this method by ‘SADbin’.

We observe that the GMC technique demands a relatively small complexity increase, since the number of SIFT features is usually low. More precisely, the encoder complexity is higher than DISCOVER $(+30\%)$ [12] but it remains significantly smaller than Intra coding with H.264/AVC. This is perfectly compatible with a low-complexity encoder scenario.

This method for SI information fusion has quite good performance with respect to previous techniques. We have even improved it using a fusion based on support vector machine [13]. Nevertheless, the block-based motion compensation can produce some unpleasant artifacts near the object contours. In order to reduce these artifacts, we propose in the current paper to resort to image segmentation into background and foreground and to use this information to perform a suitable fusion. We propose a novel tool to efficiently estimate the object contours (and therefore, to determine the segmentation map), based on elastic deformation of curves. Finally we remark that the new technique does not require a modification in the encoder and therefore its complexity (as for [12]) remains relatively low.

C. Improved Side Information Generation

The SI is usually generated through an interpolation of the backward and forward reference frames. The quality of SI is poor in certain regions of the video scene, like in areas of partial occlusions, fast motion, etc. In VISNET II codec [14], a refinement process of SI is carried out after decoding all DCT bands in order to improve reconstruction [15]. In [16][17], approaches are proposed for transform-domain DVC based on the successive refinement of SI after each decoded DCT band. In [18], a solution is proposed based on the successive refinement of SI using an adaptive search area, for long duration GOPs, in transform-domain DVC. High-order motion interpolation has been proposed [19] in order to cope with object motion with non-zero acceleration. In [20], global motion is estimated at the decoder in order to adapt temporal inter-/extrapolation for SI generation. In [21], a SI and noise learning approach is proposed, in order to enhance SI generation and noise modeling using optical flow and clustering. The SI generation problem is very similar to the one of frame-rate up conversion. In this context, Qian and Bajic [22] have introduced a region-based interpolation technique with global, local and affine perspective motion model. In fact, region-based representation allows a more coherent motion compensation, resulting in an improved visual quality of synthesized frames.

Other solutions were proposed for SI enhancement, that require a hash information to be

transmitted to the decoder. However, the encoder needs to determine in advance the regions where the interpolation at the decoder would fail, i.e. regions corresponding to a poor SI. In [23][24], hash information is extracted from the WZF being encoded and sent only for the macroblocks where the sum of squared differences between the previous reference frame and the WZF is greater than a certain threshold.

In [25] the authors proposed a Witsenhausen-Wyner Video Coding (WWVC) that employs forward motion estimation at the encoder and sends the motion vectors to the decoder to generate SI. This WWVC scheme achieves better performance than H.264/AVC in noisy networks and suffers a limited loss (up to 0.5 dB compared to H.264/AVC) in noiseless channel. The authors in [26] proposed a novel framework that integrates the graph-based segmentation and matching to generate interview SI in Distributed Multiview Video Coding.

In [27][28][29], the authors presented DVC schemes that consist in performing the motion estimation both at the encoder and decoder. In [27], the authors propose a pixel-domain DVC scheme, which consists in combining low complexity bit plane motion estimation at the encoder side, with motion compensated frame interpolation at the decoder side. Improvements are shown for sequences containing fast and complex motion. The authors in [28] present a DVC scheme where the task of motion estimation is shared between the encoder and decoder. Results have shown that the cooperation of the encoder and decoder can reduce the overall computational complexity, while improving the coding efficiency. Finally, a DVC scheme proposed by Dufaux *et al.* [29] consists in combining the global and local motion estimations at the encoder. In this scheme, the motion estimation and compensation are performed both at the encoder and decoder.

In contrast, in this paper, both global and local SI are only generated in the decoder. It is important to note that the encoding complexity is kept low. The global parameters are sent to the decoder to estimate the GMC SI, and the combination between the GMC SI and MCTI SI is made at the decoder side.

The problem of SI fusion has been addressed in Multiview DVC, where two SI are usually generated. The first SI (SI_t) is generated from previously decoded frames in the same view, while the second one (SI_v) is estimated using previously decoded frames in adjacent views. The paper [30] proposed new techniques for the fusion of SI_t and SI_v. Dufaux [31] proposed a solution that consists in combining SI_t and SI_v using Support Vector Machine (SVM). In [13], a solution is proposed for combining global and local SI using SVM, in the context of Monoview DVC.

III. PROPOSED METHODS

The block diagram of our proposed codec architecture is depicted in Fig. 2. It is based on the DISCOVER codec [6], [7].

For the segmentation of the foreground objects, the authors in [32], [33] propose a coarse-to-fine segmentation method for extracting moving regions from compressed video. In the proposed methods, we consider that the foreground objects in the Backward Reference Frame (BRF) and Forward Reference Frame (FRF) are already segmented. Here, we are interested in the combination of global and local motion estimations.

Let F_B^i and F_F^i ($i = 1, 2, ..., N_o$, N_o is the number of foreground objects) be the foreground objects already segmented from the backward and forward reference frames, respectively. Furthermore, the foreground objects masks M_B^i and M_F^i are generated from the foreground objects according to:

$$\begin{cases} M_B^i(x,y) = \begin{cases} 0 & \text{if } F_B^i(x,y) = 0 \\ 1 & \text{otherwise} \end{cases} \\ M_F^i(x,y) = \begin{cases} 0 & \text{if } F_F^i(x,y) = 0 \\ 1 & \text{otherwise} \end{cases} \end{cases} \tag{3}$$

Then, the foreground objects contours are extracted from the foreground objects masks. The contours can be considered as closed curves. Let β_B^i and β_F^i be the representations of the backward and forward foreground objects contours. As an example, Figs. 3, 4, 5, and 6 show, respectively, the original frame, the foreground object, the foreground object mask generated from the foreground object, and the generated foreground object contour, for frame number 1 of Stefan sequence.

A. Artifact removal in GMC SI using foreground objects masks

The GMC SI is simply defined as the average of the frames $\widehat{R}_B$ and $\widehat{R}_F$ [12]. Fig. 7 shows an example of a GMC SI (top center) and the GMC SI with the object mask (bottom center), for frame number 3 of Stefan sequence. As we can see, the background around the foreground object in GMC SI is affected by the shifted foreground objects due to global motion. In this case, the background in one of the reference frames is averaged with the foreground objects

Fig. 3. Original frame number 1 of Stefan sequence.

Fig. 4. Foreground object (F) of frame number 1 of Stefan sequence.

Fig. 5. Foreground object mask (M) of frame number 1 of Stefan sequence.

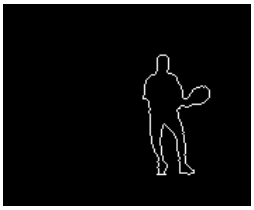

Fig. 6. Foreground object contour (β) of frame number 1 of Stefan sequence.

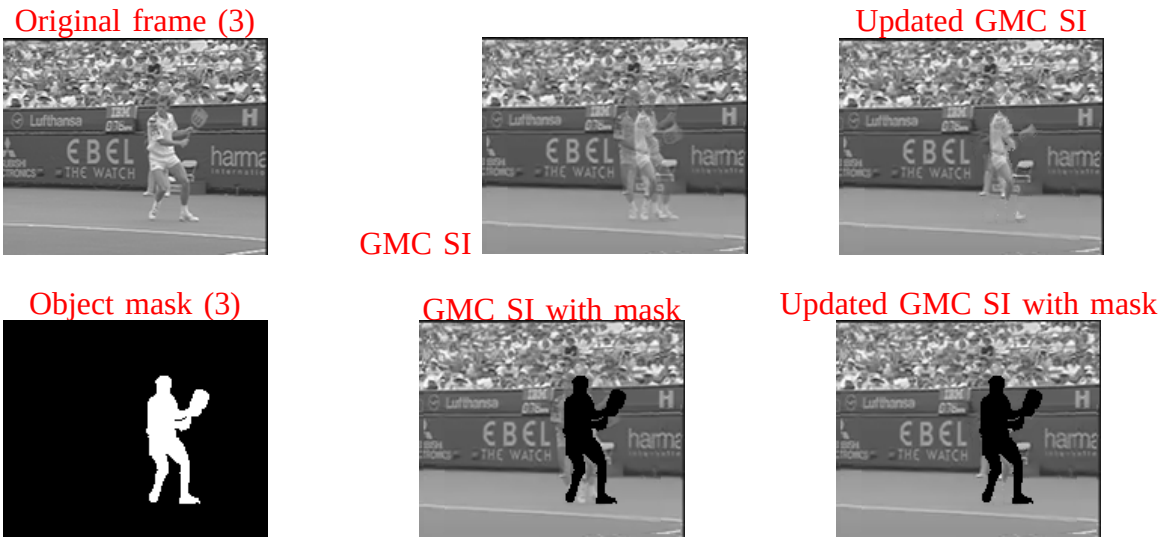

Fig. 7. Original frame, GMC SI, updated GMC SI, Object mask, GMC SI with mask, and updated GMC SI with mask for frame number 3 of Stefan sequence.

of the other reference frame. We propose to remove this artifact effect around the foreground objects using the obtained segmented foreground objects of the reference frames.

The masks M_B and M_F are defined as the union of all foreground objects masks M_B^i and

M_F^i respectively:

$$\begin{cases} M_B = \bigcup_{i=1}^{N_o} M_B^i \\ M_F = \bigcup_{i=1}^{N_o} M_F^i \end{cases} \tag{4}$$

Let $\widehat{M_B}$ and $\widehat{M_F}$ be the results of the GMC transforms T_B and T_F applied to the masks M_B and M_F respectively. $\widehat{M_B}$ and $\widehat{M_F}$ are used in order to remove the artifacts of the pixels in the background around the foreground objects. First, each pixel in the transformed frames $\widehat{R}_B$ and $\widehat{R}_F$ is assigned to either the background or the foreground objects, using $\widehat{M_B}$ and $\widehat{M_F}$. Then, in order to avoid the averaging between the background and the foreground objects, the GMC SI can be updated as follows:

$$\begin{cases} \text{if } \widehat{M_B}(x,y) = 1 \text{ and } \widehat{M_F}(x,y) = 0 \\ \quad \text{GMC SI}(x,y) = \widehat{R}_F(x,y) \\ \text{otherwise} \\ \quad \text{if } \widehat{M_B}(x,y) = 0 \text{ and } \widehat{M_F}(x,y) = 1 \\ \quad\quad \text{GMC SI}(x,y) = \widehat{R}_B(x,y) \end{cases}$$

In these situations, only the background is taken for GMC SI. Fig. 7 shows the updated GMC SI (top right) and the updated GMC SI with the object mask, for frame number 3 of Stefan sequence. It is clear that the artifact effect is removed around the foreground object, compared to the GMC SI.

B. Fusion using elastic curves

In this section our goal is to estimate the contour in SI using backward and forward contours. As described in [10], a contour can be analyzed using an elastic metric, leading up to a contour in SI. Then, the estimated contour is used to generate a mask in SI that is useful in the fusion of GMC SI and MCTI SI.

The curve β is characterized as follows:

$$\begin{aligned} \beta : \mathbb{D} &\longmapsto \mathbb{R}^2 \\ t &\longmapsto (x,y) \end{aligned} \tag{5}$$

where $t \in \mathbb{D} = [0,1]$ and (x,y) represent the coordinates of each point in the contour. For the purpose of studying the shape of β, it is represented using the Square Root Velocity (SRV)

function defined as $q : \mathbb{D} \longmapsto \mathbb{R}^2$ [10]:

$$q(t) = \frac{\dot{\beta}(t)}{\sqrt{||\dot{\beta}(t)||}} \tag{6}$$

where $||.||$ is the Euclidean norm in $\mathbb{R}^2$ and $\dot{\beta} = \frac{d\beta}{dt}$. The curve β can be obtained using q as follows:

$$\beta(t) = \int_0^t q(s)||q(s)||ds \tag{7}$$

We are given backward and forward curves β_b^i and β_f^i, treated as closed curves, and our goal is to find an estimated curve β_e^i between these two curves. The algorithm used to estimate β_e^i (Fig. 8) is described as follows (we refer the reader to [10] for the theory behind this estimation): First, the SRV representation of the curve β_b^i is computed as follows:

$$q_b^i(t) = \frac{\dot{\beta}_b^i(t)}{\sqrt{||\dot{\beta}_b^i(t)||}} \tag{8}$$

At the beginning of this algorithm, the parameters θ_{min}, δt, and k are respectively set to 2π, $\frac{1}{n}$, and zero.

Step 1 - A circular shift of $k(\delta t)$ is applied on the forward curve $\beta_f^i(t)$ as follows:

$$\tilde{\beta}_f^i(t) = \beta_f^i(t - k(\delta t)) \tag{9}$$

Then, the SRV representation of $\tilde{\beta}_f^i(t)$, denoted by $\tilde{q}_f^i(t)$, is computed using Eq. 6.

Step 2 - Rotation: The optimal rotation between q_b^i and $\tilde{q}_f^i$ is given by R_1 as follows:

$$R_1 = UIV^T \tag{10}$$

where $[U, S, V] = \text{SVD}(B)$, $B = \int_D q_b^i(t)\tilde{q}_f^i(t)^T dt$ and $I = \left(\begin{smallmatrix} 1 & 0 \\ 0 & 1 \end{smallmatrix}\right)$. Here SVD stands for the Singular Value Decomposition of a matrix. If $\det(B) < 0$, the last column of I changes sign before multiplication in Eq. 10. Then, $\tilde{q}_f^i$ is multiplied by R_1 as follows:

$$\tilde{q}_f^i(t) = R_1.\tilde{q}_f^i(t) \tag{11}$$

Following that, $\tilde{q}_f^i(t)$ is used to reconstruct $\tilde{\beta}_f^i(t)$ as follows:

$$\tilde{\beta}_f^i(t) = \int_0^t \tilde{q}_f^i(s)||\tilde{q}_f^i(s)||ds \tag{12}$$

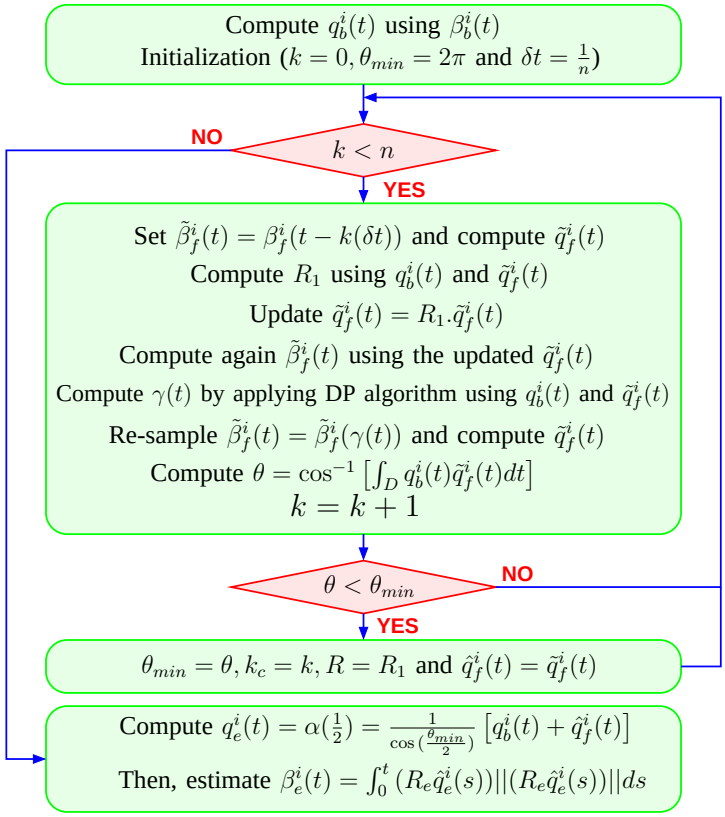

Fig. 8. Algorithm proposed in [10] for estimating $\beta_e^i(t)$.

Step 3 - Reparameterization: This step consists of using q_b^i and $\tilde{q}_f^i$ to find a function $\gamma(t)$ that is important in matching the two curves, by applying the Dynamic Programming (DP) algorithm. The obtained function $\gamma(t)$ is used to re-sample $\tilde{\beta}_f^i(t)$ as follows:

$$\tilde{\beta}_f^i(t) = \tilde{\beta}_f^i(\gamma(t)) \tag{13}$$

Consequently, $\tilde{q}_f^i(t)$ is recomputed for the updated $\tilde{\beta}_f^i(t)$ (using Eq. 6).

Step 4 - Compute the length of the geodesic θ as follows:

$$\theta = \cos^{-1}\left[\int_D q_b^i(t)\tilde{q}_f^i(t)dt\right] \tag{14}$$

If $\theta < \theta_{min}$, the parameters θ_{min}, k_c, R and $\hat{q}_f^i(t)$ are updated as follows:

$$\begin{cases} \theta_{min} = \theta \\ k_c = k \\ R = R_1 \\ \hat{q}_f^i(t) = \tilde{q}_f^i(t) \end{cases} \tag{15}$$

Then, k is set to $k+1$. If k is smaller than n, go to **Step 1**. Otherwise, go to **Step 5**

Step 5 - The geodesic $\alpha(\tau), \tau \in [0,1]$ that connects $q_b^i(t)$ and $\hat{q}_f^i(t)$, is defined as follows:

$$\begin{aligned} \alpha(\tau) &= \frac{1}{\sin(\theta_{min})} \big[\sin(\theta_{min}(1-\tau)) q_b^i(t) \\ &\quad + \sin(\theta_{min}\tau)\hat{q}_f^i(t)\big] \end{aligned} \tag{16}$$

It is clear that $\alpha(0) = q_b^i(t)$ and $\alpha(1) = \hat{q}_f^i(t)$. This equation allows predicting the curves between the backward curve β_b^i and the forward curve β_f^i at any time $\tau \in [0,1]$. Here, we aim to estimate the curve in the middle between the backward and forward curves. For this reason, we compute $\alpha(\frac{1}{2})$ to obtain $q_e^i(t)$ as follows:

$$\begin{aligned} q_e^i(t) &= \alpha(\frac{1}{2}) \\ &= \frac{1}{\sin(\theta_{min})} \left[\sin(\frac{\theta_{min}}{2}) q_b^i(t) \right. \\ &\quad \left. + \sin(\frac{\theta_{min}}{2})\hat{q}_f^i(t)\right] \\ &= \frac{1}{\cos(\frac{\theta_{min}}{2})} \big[q_b^i(t) + \hat{q}_f^i(t)\big] \end{aligned} \tag{17}$$

Then, $q_e^i(t)$ is projected [10] in $\mathbf{C}^c$ to obtain $\hat{q}_e^i(t)$ ($\mathbf{C}^c$ represents the closed curves).

Step 6 - The objective of this step is to obtain the curve $\beta_e^i(t)$ using $\hat{q}_e^i(t)$ with the rotation matrix R. The rotation matrix can be written as follow:

$$R = \begin{pmatrix} \cos(\varphi) & -\sin(\varphi) \\ \sin(\varphi) & \cos(\varphi) \end{pmatrix}$$

where φ is the angle of rotation. The rotation matrix R_e for the estimated curve can be written as follows:

$$R_e = \begin{pmatrix} \cos(\phi_e) & -\sin(\phi_e) \\ \sin(\phi_e) & \cos(\phi_e) \end{pmatrix}$$

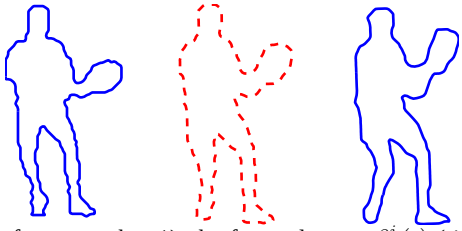

Fig. 9. The backward curve $\beta_b^i(t)$ (left, frame number 1), the forward curve $\beta_f^i(t)$ (right, frame number 3) and the estimated curve $\beta_e^i(t)$ (center, $\tau = \frac{1}{2}$) between the backward and forward curves.

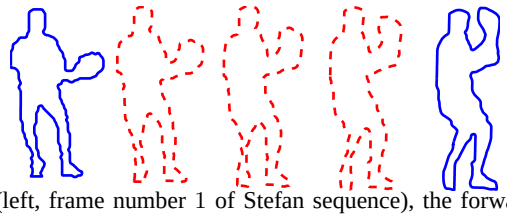

Fig. 10. The backward curve $\beta_b^i(t)$ (left, frame number 1 of Stefan sequence), the forward curve $\beta_f^i(t)$ (right, frame number 5) and the three estimated curves $\beta_e^i(t)$ for $\tau = \frac{1}{4}, \frac{2}{4}$ and $\frac{3}{4}$ (center curves).

where $\phi_e = \frac{\varphi}{2}$. The curve $\beta_e^i(t)$ can be estimated as follows:

$$\beta_e^i(t) = \int_0^t (R_e \hat{q}_e^i(s))||(R_e \hat{q}_e^i(s))||ds \tag{18}$$

Fig. 9 shows an application example of this algorithm, where we show the backward curve $\beta_b^i(t)$ (left curve) of frame number 1 of Stefan sequence, the forward curve $\beta_f^i(t)$ (right curve) of frame number 3 of this sequence, and the estimated curve $\beta_e^i(t)$ (center curve) between the backward and forward curves using this algorithm. Moreover, Fig. 10 shows the backward curve $\beta_b^i(t)$ (left) of frame number 1 of Stefan sequence, the forward curve $\beta_f^i(t)$ (right) of frame number 5 of Stefan sequence and the estimated curves $\beta_e^i(t)$ for $\tau = \frac{1}{4}, \frac{2}{4}$ and $\frac{3}{4}$(center curves).

The obtained curves $\beta_e^i(t)$ are then used to obtain the foreground objects masks M_e^i by covering all the pixels lying inside the curves. The mask M_e is defined as the union of all masks M_e^i:

$$M_e = \bigcup_{i=1}^{N_o} M_e^i \tag{19}$$

Then, to generate SI, the pixels inside the mask M_e are selected from MCTI SI and the background pixels from GMC SI:

$$\text{SI}(x,y) = \begin{cases} \text{MCTI SI}(x,y) & \text{if } M_e(x,y) = 1 \\ \text{GMC SI}(x,y) & \text{otherwise} \end{cases} \tag{20}$$

Fig. 11. Foreground objects of frames number 1 and 9 of Foreman sequence, split into 16×16 blocks.

This fusion method is referred to as 'FusElastic'.

C. Fusion using local motion compensation

In this section, we propose to apply the MCTI technique [8] to the foreground objects in order to estimate the local motion. Then, a new scheme for local motion estimation is proposed.

1) Applying MCTI on the foreground objects: In this approach, the MCTI technique is applied to the backward foreground object F_B^i and the forward foreground object F_F^i, in order to estimate the foreground object F_{MCTI}^i in SI. In this case, there are blocks entirely black, partly black, or entirely white. Fig. 11 shows foreground objects for frames number 1 and 9 of Foreman sequence, split into 16×16 blocks. In contrast, the classical MCTI SI is estimated by applying the MCTI technique to the whole (Background and Foreground) reference frames. Let F_{MCTI} be the union of all foreground objects in SI, which are estimated using the MCTI technique:

$$F_{\text{MCTI}} = \bigcup_{i=1}^{N_o} F_{\text{MCTI}}^i \tag{21}$$

The mask M_{MCTI} is generated from the estimated foreground objects F_{MCTI} as follows:

$$M_{\text{MCTI}}(x,y) = \begin{cases} 0 & \text{if } F_{\text{MCTI}}(x,y) = 0 \\ 1 & \text{otherwise} \end{cases} \tag{22}$$

Here, we propose two approaches for the combination of global and local motion estimations, based on the generated mask M_{MCTI}. The first approach consists in fusing GMC SI with the estimated foreground objects F_{MCTI} using:

$$\text{SI}(x,y) = \begin{cases} F_{\text{MCTI}}(x,y) & \text{if } M_{\text{MCTI}}(x,y) = 1 \\ \text{GMC SI}(x,y) & \text{otherwise} \end{cases} \tag{23}$$

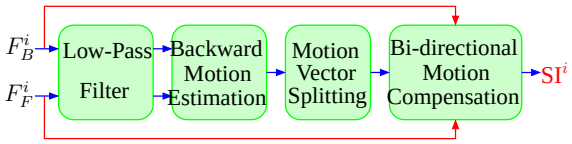

Fig. 12. Proposed method for foreground objects estimation.

This method is referred to as 'FoMCTI'.

The second approach makes the fusion of GMC SI and MCTI SI (taken within the masks) and is defined as follows:

$$\text{SI}(x,y) = \begin{cases} \text{MCTI SI}(x,y) & \text{if } M_{\text{MCTI}}(x,y) = 1 \\ \text{GMC SI}(x,y) & \text{otherwise} \end{cases} \tag{24}$$

This method is referred to as 'FoMCTI2'.

2) Proposed local motion estimation: In this section, we propose a new method for estimating the foreground objects in SI, using the backward and forward foreground objects. The proposed scheme is illustrated in Fig. 12. This technique is referred to as Foreground Object Motion Compensation (FOMC).

- **Low-Pass Filtering:** The backward F_B^i and foreground F_F^i foreground objects are low-pass filtered in order to improve the motion vectors reliability.
- **Backward Motion Estimation:** A Block Matching Algorithm (BMA) is applied to estimate the backward motion vector field. This estimation is done using a block size 16×16, a search area ($\mathbf{S}$) of ± 32 pixels, and a step size of 2 pixels. First, if all the pixels in the current block b in F_F^i and the co-located block in F_B^i are black (corresponding to non-object pixels), the motion vector is set to $\mathbf{0}$ for this block (see Fig. 11). In the case when the block b is partly black, the BMA is used to find the corresponding block (*i.e.*, BMA can find the most similar shape).

  In the BMA, the Weighted Mean Absolute Difference (WMAD) criterion is used to compute the similarity between the target block b in the forward foreground object frame F_F^i and the shifted block in the backward foreground object frame F_B^i by the motion vector $\mathbf{v} \equiv (v_x, v_y) \in \mathbf{S}$, as follows:

$$\begin{aligned} \text{WMAD}(b,\mathbf{v}) &= \frac{1}{16^2}\left(1+\lambda\sqrt{\|\mathbf{v}\|}\right) \\ &\times \sum_{\mathbf{p}\in E_B} |F_F^i(\mathbf{p}) - F_B^i(\mathbf{p}+\mathbf{v})| \end{aligned} \tag{25}$$

λ a penalty factor used to penalize the MAD by the length of the motion vector $\|\mathbf{v}\| = \sqrt{v_x^2+v_y^2}$ (it is empirically set to 0.05). An extended block E_B of $(16+2e, 16+2e)$ (e being empirically set to 8) is used in the WMAD, and $\mathbf{p}=(x,y)$ represents the coordinates of each pixel in the extended block E_B. The best backward motion vector $\mathbf{V_b}$ for the block b is obtained by minimizing the WMAD as follows:

$$\mathbf{V_b} = \arg\min_{\mathbf{v_i}\in\mathbf{S}} \text{WMAD}(b,\mathbf{v}_i). \tag{26}$$

- **Motion Vector Splitting:** Here, the obtained motion vectors are divided in such a way to obtain bi-directional motion vectors for the blocks in the estimated foreground object F^i_{FOMC}. For each block b in F^i_{FOMC}, the distances between the center of the block b and the center of each obtained motion vector are computed. The closest motion vector to the block b is selected. Then, the selected motion vector is associated to the center of the block b, and divided by symmetry to obtain the bidirectional motion field.
- **Bi-directional Motion Compensation:** Once the final bidirectional motion vectors are estimated, the F^i_{FOMC} can be interpolated using bidirectional motion compensation as follows:

$$F^i_{\text{FOMC}}(\mathbf{p}) = \frac{1}{2}(F_B^i(\mathbf{p}+\mathbf{s}_b) + F_F^i(\mathbf{p}-\mathbf{s}_b)), \tag{27}$$

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

The F^i_{FOMC} is estimated for each foreground object i ($i = 1, 2, ..., N_o$). Then, all F^i_{FOMC} are combined to form F_{FOMC} using:

$$\text{F}_{\text{FOMC}} = \bigcup_{i=1}^{N_o} F^i_{\text{FOMC}} \tag{28}$$

Furthermore, the mask M_{FOMC} is generated using F_{FOMC} as follows:

$$M_{\text{FOMC}}(x,y) = \begin{cases} 0 & \text{if } F_{\text{FOMC}}(x,y) = 0 \\ 1 & \text{otherwise} \end{cases} \tag{29}$$

Here, two approaches are proposed to combine the global and local motion estimations using M_{FOMC}. The first one aims at combining GMC SI and F_{FOMC} using:

$$\text{SI}(x,y) = \begin{cases} F_{\text{FOMC}}(x,y) & \text{if } M_{\text{FOMC}}(x,y) = 1 \\ \text{GMC SI}(x,y) & \text{otherwise} \end{cases} \tag{30}$$

This method is referred to as 'BmEst'.

The second approach consists in combining GMC SI and MCTI SI as follows:

$$\text{SI}(x,y) = \begin{cases} \text{MCTI SI}(x,y) & \text{if } M_{\text{FOMC}}(x,y) = 1 \\ \text{GMC SI}(x,y) & \text{otherwise} \end{cases} \tag{31}$$

This method is referred to as 'BmMCTI'.

D. Oracle fusion method

In this section, we describe the oracle fusion method which consists in fusing GMC SI and MCTI SI using the foreground objects masks of the original WZFs. Let M_{WZF} be the union of all foreground objects masks in the original WZF :

$$M_{\text{WZF}} = \bigcup_{i=1}^{N_o} M^i_{\text{WZF}} \tag{32}$$

M^i_{WZF} is the i^{th} foreground object mask in the WZF. The oracle fusion method combines GMC SI and MCTI SI as follows:

$$\text{SI}(x,y) = \begin{cases} \text{MCTI SI}(x,y) & \text{if } M_{\text{WZF}}(x,y) = 1 \\ \text{GMC SI}(x,y) & \text{otherwise} \end{cases} \tag{33}$$

This method is of course impractical, but it allows us to estimate the ideal upper bound limit that can be achieved by combining GMC SI and MCTI SI, using the foreground objects masks of the original WZF.

IV. Experimental results

Here, the segmentation masks for the reference frames are assumed to be known. The performance of the proposed methods are assessed using extensive simulations under the same test conditions as in DISCOVER [6], [7]. An example is illustrated in Fig 13 for several test sequences with the corresponding foreground objects: Stefan (one object, 45 frames), Foreman (one object, 150 frames), Bus (three objects, 75 frames), and Coastguard (two objects, 150 frames). The obtained results of the proposed methods are compared to the DISCOVER codec, VISNET II, GMC technique, and to our previous fusion technique SADbin.

Fig. 13. The foreground objects in the test sequences: Stefan (one object), Foreman (one object), Bus (three objects), and Coastguard (two objects).

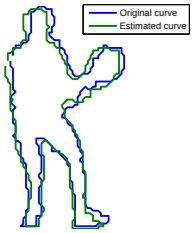

Fig. 14. Comparison between the original curve and the estimated curve using the elastic curve [10] for frame number 2 of Stefan sequence.

1) SI performance assessment: Fig. 14 shows the original curve and the estimated curve using the elastic curve algorithm [10], for frame number 2 of Stefan sequence, for a GOP size of 2. It is clear that the difference between the two curves is small.

We performed a first set of experiments in order to assess the effectiveness of the elastic deformation tool in providing an accurate segmentation map of the WZFs. Since we use the contours to classify the pixels as background or foreground, a relevant metric is the confusion matrix [34]. More precisely, we consider the ground-truth classification and we compare it to the classification obtained with the elastic curves. The classification results (averaged over all the data set images) are given in terms of "true positives" (*i.e.* the foreground pixels correctly classified as foreground), "false negatives" (foreground pixels classified as background), "false positives" (background classified as foreground) and "true negatives". Finally, we compute the foreground accuracy as the number of true foreground pixels over the number of actual foreground pixels, and similarly for the background. These results are reported in Tab. I, for all GOP sizes. We

TABLE I

CONFUSION MATRIX (PER-IMAGE AVERAGE) FOR THE BACKGROUND/FOREGROUND CLASSIFICATION USED THE ELASTIC DEFORMATION OF OBJECT CONTOURS, FOR ALL GOP SIZES

	Foreground (Predicted)	Background (Predicted)	Accuracy(%)
GOP = 2			
Foreground (Actual)	2718	122	93.52
Background (Actual)	200	22302	98.96
Overall Accuracy (%)			98.73
GOP = 4			
Foreground (Actual)	2708	147	92.45
Background (Actual)	228	22259	98.81
Overall Accuracy (%)			98.52
GOP = 8			
Foreground (Actual)	2690	179	90.66
Background (Actual)	249	22224	98.72
Overall Accuracy (%)			98.31

TABLE II

SI AVERAGE PSNR FOR A GOP SIZE EQUAL TO 2, 4, AND 8 (QI = 8).

SI Average PSNR [dB]									
Method	**MCTI**	**GMC**	**SADbin**	**FusElastic**	**BmEst**	**BmMCTI**	**FoMCTI**	**FoMCTI2**	**Oracle fusion**
GOP = 2									
Stefan	25.17	27.70	28.16	28.43	**28.72**	28.53	28.69	28.49	28.71
Foreman	29.38	30.70	30.82	31.09	30.97	31.11	30.99	**31.13**	31.15
Bus	25.37	23.10	27.30	27.30	26.92	**27.56**	27.30	27.48	27.90
Coastguard	31.47	29.28	32.00	31.80	31.91	31.91	**32.03**	31.89	32.07
GOP = 4									
Stefan	23.49	27.22	27.18	27.72	**27.95**	27.86	27.87	27.79	28.14
Foreman	27.64	29.62	29.27	29.79	29.71	29.82	29.71	**29.83**	29.88
Bus	24.00	22.53	26.27	26.29	26.02	**26.54**	26.28	26.39	26.91
Coastguard	29.91	28.19	30.76	30.68	30.77	30.73	**30.88**	30.72	30.88
GOP = 8									
Stefan	22.84	27.06	26.91	27.35	**27.67**	27.55	27.55	27.46	27.80
Foreman	26.29	28.62	28.09	28.74	28.64	28.75	28.65	**28.77**	28.83
Bus	22.95	21.95	25.26	25.33	25.13	**25.55**	25.36	25.45	25.94
Coastguard	28.82	27.50	29.85	29.77	29.88	29.83	**29.96**	29.82	30.00

observe that the classification produced with the elastic deformation is quite accurate, and this explains the good rate-distortion performance of our technique and we can observe that the

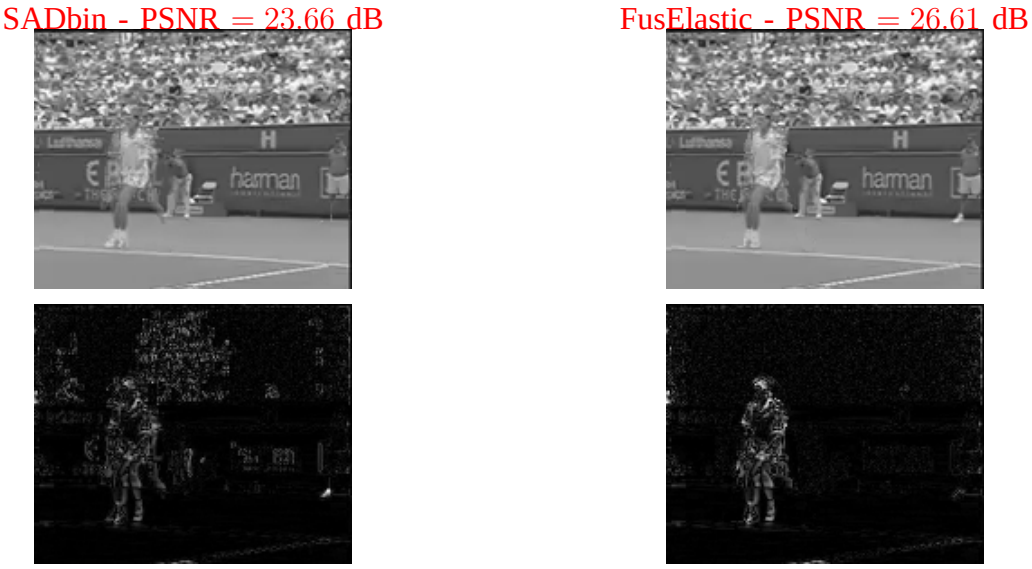

Fig. 15. Visual result of SI estimated by SADbin (PSNR = 23.66 dB) and FusElastic (PSNR = 26.61 dB), for frame number 27 of Stefan sequence, for a GOP size of 4 (QI = 8). The bottom images represents the visual differences of these SI frames.

accuracy is decreased with the GOP size.

Table II shows the average PSNR of SI obtained with MCTI, GMC, SADbin, FusElastic, BmEst, BmMCTI, FoMCTI, FoMCTI2, and Oracle fusion for Stefan, Foreman, Bus, and Coastguard sequences, for GOP sizes of 2, 4, and 8. The average PSNR of the KFs (QI = 8) is up to 33.45 dB, 39.25 dB, 34.41 dB, and 37.11 dB for Stefan, Foreman, Bus, and Coastguard sequences respectively. It is clear that the proposed fusion methods can improve the quality of SI compared to MCTI and GMC for all test sequences and all GOP sizes. The proposed method FusElastic can achieve a gain compared to the previous fusion SADbin for Stefan and Foreman sequences. For Bus sequence, the PSNR average of the two approaches SADbin and FusElastic is almost the same. For Coastguard sequence, the SADbin can achieve a slight gain compared to FusElastic.

Concerning BmEst and BmMCTI fusion methods, BmEst can achieve a gain compared to BmMCTI for Stefan and Coastguard sequences, while BmMCTI outperforms BmEst for Foreman and Bus sequences. According to this comparison, we can say that the estimation of the foreground objects in MCTI SI is better than the estimation of the foreground objects using our FOMC method for Foreman and Bus sequences. However, FOMC is better than MCTI in the

estimation of the foreground objects for Stefan and Coastguard sequences.

Concerning FoMCTI and FoMCTI2, we can see the same comparison as between BmEst and BmMCTI. Therefore, when the MCTI technique is only applied on the foreground objects, the quality of the estimated foreground objects is better than the quality of MCTI SI, for Stefan and Coastguard sequences. For Foreman and Bus sequences, the estimation of the foreground objects in MCTI SI is better than the quality of the generated foreground objects by applying MCTI only on the foreground objects.

It is important to note that the oracle fusion method represents the fusion of GMC SI and MCTI SI using the foreground objects of the original WZF. However, BmEst and FoMCTI methods represent the fusion of GMC SI and the estimated foreground objects. Thus, the oracle fusion represents the upper bound limit that can be achieved by the proposed fusion methods excluding BmEst and FoMCTI. For this reason, the average PSNR obtained by BmEst (28.72 dB) is slightly better than that the average PSNR of the oracle fusion (28.71 dB), for Stefan sequence, with a GOP size of 2.

Fig. 15 shows the visual results and the visual differences of SI for frame number of 27 of Stefan sequence, for a GOP size of 4. The SI obtained by SADbin fusion may contain block artifacts (top-left - 23.66 dB). The proposed fusion FusElastic can improve the quality of SI for this frame (top-right - 26.61 dB), with a gain of 2.95 dB compared to SADbin.

The RD performance of the proposed methods GMC, SADbin, FusElastic, BmEst, BmMCTI, FoMCTI, and FoMCTI2 is shown along with VISNET II and the Oracle fusion, for Stefan, Bus, Foreman, and Coastguard sequences in Table III, in comparison to the DISCOVER codec, using the Bjontegaard metric [35], for GOP sizes of 2, 4, and 8.

All the fusion methods can achieve a gain compared to DISCOVER codec. The proposed method FusElastic allows a gain compared to SADbin for Stefan and Foreman sequences for a GOP size of 2, and for all test sequences for a GOP size of 8. The gain is up to 4.6 dB compared to DISCOVER codec and 0.55 dB compared to SADbin, for a GOP size of 8. The loss is up to 0.04 dB compared to SADbin for Bus sequence with a GOP size of 2.

The remaining fusion methods almost achieve the same gains compared to DISCOVER. The gain is up to 4.73 dB compared to DISCOVER codec for Stefan sequence, for a GOP size of 8.

Figs. 16, 17, and 18 show the RD performance curves of the DISCOVER codec, SADbin, FusElastic, and the Oracle fusion method, for Stefan, Foreman, Bus, and Coastguard sequences,

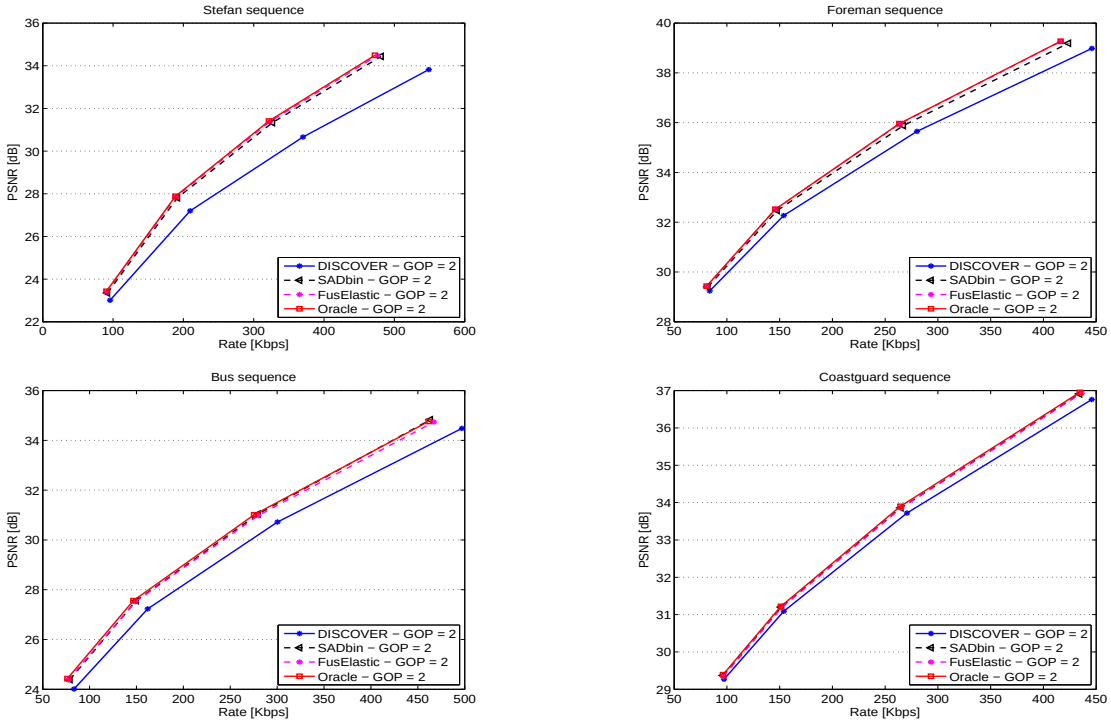

Fig. 16. RD performance comparison among DISCOVER, SADlin, FusElastic, and Oracle fusion method for Stefan, Foreman, Bus, and Coastguard sequences, for a GOP size of 2.

for GOP sizes of 2, 4, and 8 respectively. The proposed fusion methods SADbin and FusElastic always achieve a gain compared to DISCOVER codec for all test sequences. The proposed fusion FusElastic can achieve a gain up to 0.13 dB, 0.45 dB, and 0.55 dB compared to SADbin fusion for a GOP size of 2, 4, and 8 respectively, for Stefan sequence. For Foreman sequence, FusElastic fusion allows a gain up to 0.14 dB, 0.43 dB, and 0.64 dB respectively for a GOP size of 2, 4, and 8. For Bus and Coastguard sequences, the two methods SADbin and FusElastic almost achieve the same RD performance.

Finally, in order to validate our technique in a more realistic scenario, we evaluated the effect of using non-ideal segmentation maps. More precisely, we implemented a simple video segmentation algorithm, based on mathematical morphology processing of the difference between the current image and the background (the latter obtained by global motion compensation on previous frames). This algorithm gives acceptable segmentation masks, even though some inaccuracy is visible from time to time. However, using the computed segmentation maps

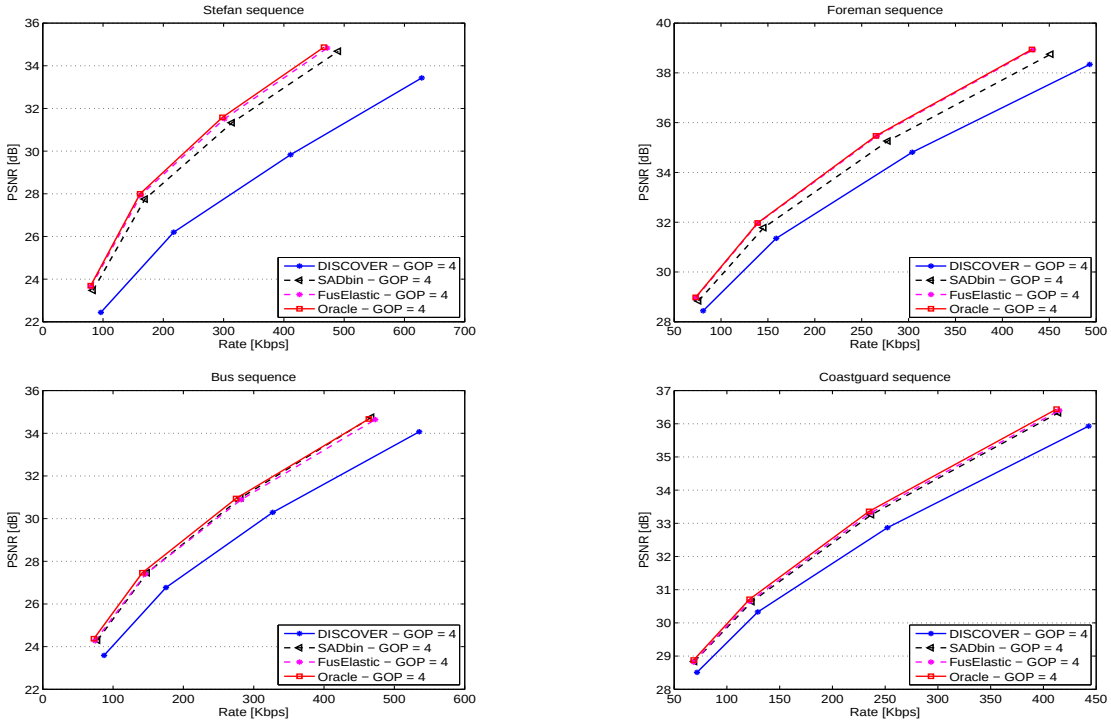

Fig. 17. RD performance comparison among DISCOVER, SADlin, FusElastic, and Oracle fusion method for Stefan, Foreman, Bus, and Coastguard sequences, for a GOP size of 4.

instead of the ideal ones in our system does not degrade too much the global rate-distortion performance: we observed a rate increase of 0.2% (GOP$=2$) to 0.8% (GOP$=8$). This preliminary experiment shows that the proposed method has the potential of good coding gains even when the segmentation is not perfect.

To measure the encoding complexity of the proposed method, we use a machine with a dual core Pentium D processor, at 3.4 GHz, with 2048 MB of RAM. We take the average of the obtained encoding times of the Coastguard and Foreman sequences. The encoding times of DISCOVER, the proposed method, H.264/AVC Intra, and H.264/AVC No motion are respectively equal to 28.4, 36.9, 49.9, and 50.4 seconds. These results prove that the increase in complexity in our proposed technique, w.r.t. DISCOVER encoder, remains moderate, and that the complexity of the new encoder is still much lower than that of H.264/AVC Intra and H.264/AVC No motion.

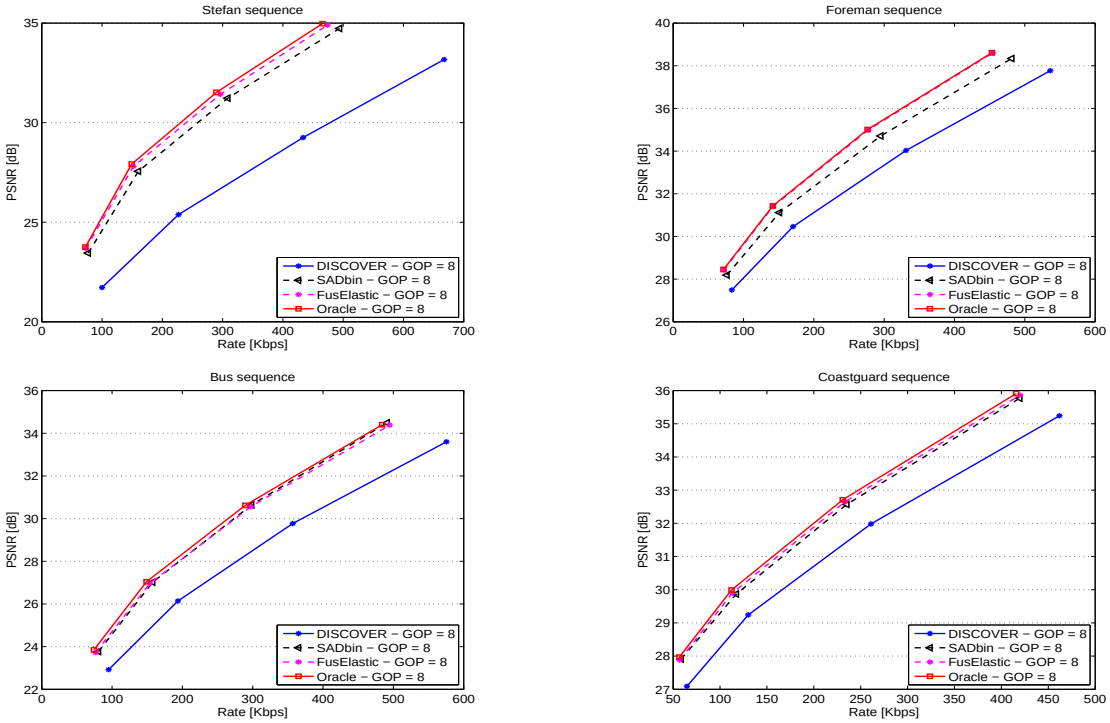

Fig. 18. RD performance comparison among DISCOVER, SADlin, FusElastic, and Oracle fusion method for Stefan, Foreman, Bus, and Coastguard sequences, for a GOP size of 8.

V. Conclusion

In this paper, new approaches have been proposed to combine the global and local motion estimations, based on the foreground objects. In the first one, elastic curves are used to estimate the contour of the foreground objects. Based on the estimated contour, the fusion of GMC SI and MCTI SI is performed. Second, the foreground objects are estimated using MCTI and FOMC techniques. In this case, for the local motion, MCTI SI and the estimated foreground objects are available. Thus, two approaches for the fusion are proposed. The first one aims at fusing GMC SI with the estimated foreground objects. The second one combines GMC SI and MCTI SI.

The proposed fusion methods allow consistent performance gains compared to DISCOVER codec and to our SADbin fusion method. The gain is up to 4.73 dB compared to DISCOVER codec, and up to 0.68 dB compared to SADbin, for a GOP size equal to 8. It is important to note that compared to SADbin, no complexity is added to the encoder, in all the proposed fusion techniques, since contours and masks generation, as well as foreground object estimations, are

all performed at the receiver side. Besides, since the quality of SI is enhanced by the new fusion techniques, a smaller number of decoder runs is generally required for the channel decoder to converge (*i.e.* less requests of parity bits through the feedback channel).

Future work will be focusing on further improvement of the fusion in order to achieve a better RD performance. We will investigate the use of the estimated contours by elastic curves in the estimation of the foreground objects. In addition, we will apply an efficient algorithm to segment the foreground objects from the decoded reference frames.

REFERENCES

[1] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding 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," in *Proc. of Picture Coding Symposium*, Lisboa, Portugal, Oct. 2007.

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

[8] J. Ascenso, C. Brites, and F. Pereira, "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*, Slovak, July 2005.

[9] S.H. Joshi, E. Klassen, A. Srivastava, and I. Jermyn, "A novel representation for Riemannian analysis of elastic curves in $\mathbb{R}^n$," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2007, pp. 1–7.

[10] A. Srivastava, E. Klassen, S.H. Joshi, and I.H. Jermyn, "Shape analysis of elastic curves in euclidean spaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 1415–1428, July 2011.

[11] M. Cagnazzo, S. Parrilli, G. Poggi, and L. Verdoliva, "Costs and advantages of object-based image coding with shape-adaptive wavelet transform," *EUR-JIVP*, vol. 2007, pp. Article ID 78323, 13 pages, 2007, doi:10.1155/2007/78323.

[12] A. Abou-Elailah, F. Dufaux, M. Cagnazzo, B. Pesquet-Popescu, and J. Farah, "Fusion of global and local motion estimation for distributed video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 23, no. 1, pp. 158–172, Jan. 2013.

[13] A. Abou-Elailah, F. Dufaux, M. Cagnazzo, and J. Farah, "Fusion of global and local side information using support vector machine in transform-domain DVC," in *European Signal Processing Conference (EUSIPCO)*, Bucharest, Romania, Aug. 2012.

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

[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] A. Abou-Elailah, J. Farah, M. Cagnazzo, B. Pesquet-Popescu, and F. Dufaux, "Improved side information for distributed video coding," in *3nd European Workshop on Visual Information Processing (EUVIP)*, Paris, France, July 2011, pp. 42 – 49.

[17] 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.

[18] A. Abou-Elailah, F. Dufaux, M. Cagnazzo, B. Pesquet-Popescu, and J. Farah, "Successive refinement of side information using adaptive search area for long duration GOPs in distributed video coding," in *19th International Conference on Telecommunications (ICT)*, Jounieh, Lebanon, Apr. 2012.

[19] G. Petrazzuoli, M. Cagnazzo, and B. Pesquet-Popescu, "High order motion interpolation for side information improvement in DVC," in *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, June 2010, pp. 2342 – 2345.

[20] R. Hansel and E. Muller, "Global motion guided adaptive temporal inter-/extrapolation for side information generation in distributed video coding," in *IEEE International Conference on Image Processing*, Brussels, Belgium, Sept. 2011, pp. 2681 – 2684.

[21] H. V. Luong, L. L. Raket, X. Huang, and S. Forchhammer, "Side information and noise learning for distributed video coding using optical flow and clustering," *IEEE Transactions on Image Processing*, vol. 21, pp. 4782–4796, Dec. 2012.

[22] C. Qian and I. V. Bajic, "Frame rate up-conversion using global and local higher-order motion," in *IEEE International Conference on Multimedia and Expo (ICME)*, San Jose, CA, 2013.

[23] A. Aaron, S. Rane, and B. Girod, "Wyner-Ziv video coding with hash-based motion compensation at the receiver," in *Proceedings of IEEE International Conference on Image Processing*, Singapore, Oct. 2004, vol. 05, pp. 3097–3100.

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

[25] M. Guo, Z. Xiong, F. Wu, D. Zhao, X. Ji, and W. Gao, "Witsenhausen-Wyner video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 21, pp. 1049 – 1060, 2011.

[26] H. Xiong, H. Lv, Y. Zhang, L. Song, Z. He, and T. Chen, "Subgraphs matching-based side information generation for distributed multiview video coding," *EURASIP Journal on Advances in Signal Processing*, p. 17 pages, 2009.

[27] T. Clercks, A. Munteanu, J. Cornelis, and P. Schelkens, "Distributed video coding with shared encoder/decoder complexity," in *Proc. IEEE International Conference on Image Processing*, San Antonio, TX, Sept. 2007.

[28] H. Chen and E. Steinbach, "Flexible distribution of computational complexity between the encoder and the decoder in distributed video coding," in *Proc. IEEE International Conference on Multimedia and Expo*, Hannover, Germany, June 2008.

[29] F. Dufaux and T. Ebrahimi, "Encoder and decoder side global and local motion estimation for distributed video coding," in *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, 2010, pp. 339 – 344.

[30] T. Maugey, W. Miled, M. Cagnazzo, and B. Pesquet-Popescu, "Fusion schemes for multiview distributed video coding,"

in *17th European Signal Processing Conference (EUSIPCO)*, Scotland, Aug. 2009.

[31] F. Dufaux, "Support vector machine based fusion for multi-view distributed video coding," in *17th International Conference on Digital Signal Processing (DSP)*, Corfu, Aug. 2011, pp. 1 –7.

[32] Y.-M. Chen, I.V. Bajic, and P. Saeedi, "Coarse-to-fine moving region segmentation in compressed video," in *10th Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS)*, May 2009, pp. 45–48.

[33] Y.-M. Chen and I.V. Bajic, "Compressed-domain moving region segmentation with pixel precision using motion integration," in *IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PacRim)*, Aug. 2009, pp. 442–447.

[34] S. V. Stehman, "Selecting and interpreting measures of thematic classification accuracy," *Remote Sensing of Environment*, vol. 62, pp. 77–89, Oct. 1997.

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

TABLE III

RATE-DISTORTION PERFORMANCE GAIN FOR *Stefan*, *Foreman*, *Bus*, AND *Coastguard* SEQUENCES TOWARDS DISCOVER CODEC, USING BJONTEGAARD METRIC, FOR A GOP SIZE OF 2, 4, AND 8.

Method	**VISNET II**	**GMC**	**SADbin**	**FusElastic**	**BmEst**	**BmMCTI**	**FoMCTI**	**FoMCTI2**	**Oracle fusion**
GOP = 2									
Stefan									
Δ_R (%)	4.02	-18.21	-17.97	-19.72	**-20.06**	-19.98	**-20.05**	-19.79	-20.38
Δ_{PSNR} [dB]	-0.26	1.25	1.23	1.36	**1.39**	1.38	**1.39**	1.37	1.41
Foreman									
Δ_R (%)	-2.87	-8.42	-7.58	**-9.65**	-8.51	**-9.67**	-8.37	**-9.70**	-10.07
Δ_{PSNR} [dB]	0.13	0.52	0.45	**0.59**	0.52	**0.59**	0.49	**0.59**	0.61
Bus									
Δ_R (%)	5.96	6.36	-12.94	-12.51	-10.25	**-13.34**	-10.75	-11.25	-14.51
Δ_{PSNR} [dB]	-0.35	-0.32	0.79	0.75	0.61	**0.80**	0.64	0.68	0.87
Coastguard									
Δ_R (%)	2.01	10.32	-4.60	-4.32	-4.34	**-4.74**	-4.40	-4.33	-5.36
Δ_{PSNR} [dB]	-0.10	-0.48	0.23	0.22	0.22	**0.24**	0.22	0.21	0.27
GOP = 4									
Stefan									
Δ_R (%)	-4.08	-44.05	-40.66	-45.18	-45.73	-45.74	-45.80	**-45.71**	-46.42
Δ_{PSNR} [dB]	0.17	3.26	2.93	3.38	3.42	3.44	3.44	**3.45**	3.51
Foreman									
Δ_R (%)	-11.68	-22.53	-15.54	**-21.72**	-20.91	-21.81	-20.34	**-21.93**	-22.41
Δ_{PSNR} [dB]	0.52	1.37	0.90	**1.33**	1.25	1.32	1.19	**1.33**	1.36
Bus									
Δ_R (%)	1.95	-1.82	-25.95	-25.97	-24.10	**-27.45**	-22.19	-23.67	-28.60
Δ_{PSNR} [dB]	-0.17	0.11	1.60	1.57	1.41	**1.67**	1.34	1.40	1.78
Coastguard									
Δ_R (%)	-0.27	8.43	-14.91	-16.48	-16.37	**-16.59**	-16.24	-15.70	-17.94
Δ_{PSNR} [dB]	-0.00	-0.35	0.61	0.68	0.68	**0.69**	0.67	0.65	0.75
GOP = 8									
Stefan									
Δ_R (%)	-8.85	-55.20	-51.56	-55.95	**-57.12**	-57.04	**-57.10**	-56.94	-57.84
Δ_{PSNR} [dB]	0.43	4.51	4.05	4.60	**4.72**	4.72	**4.73**	4.72	4.83
Foreman									
Δ_R (%)	-18.84	-31.81	-22.29	**-31.24**	-30.09	-31.01	-29.12	-30.78	-31.80
Δ_{PSNR} [dB]	0.81	2.02	1.29	**1.93**	1.84	1.92	1.76	1.91	1.97
Bus									
Δ_R (%)	-4.15	-10.33	-32.07	-32.82	-31.58	**-34.16**	-27.87	-28.53	-35.50
Δ_{PSNR} [dB]	0.06	0.58	2.04	2.07	1.97	**2.19**	1.72	1.74	2.31
Coastguard									
Δ_R (%)	-8.59	-5.57	-26.32	-29.50	**-30.37**	-29.73	-29.48	-28.19	-31.32
Δ_{PSNR} [dB]	0.33	0.15	1.10	1.24	**1.27**	1.26	1.23	1.18	1.35