Fusion of Global and Local Side Information using Support Vector Machine in Transform-Domain DVC

Abdel-Bassir Abou-Elailah, Frederic Dufaux, J. Farah, M. Cagnazzo

FUSION OF GLOBAL AND LOCAL SIDE INFORMATION USING SUPPORT VECTOR MACHINE IN TRANSFORM-DOMAIN DVC

Abdalbassir Abou-Elailah [1], Frederic Dufaux [1], Joumana Farah [2], and Marco Cagnazzo [1]

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

ABSTRACT

Side information has a strong impact on the performance of Distributed Video Coding. Commonly, side information is generated using motion compensated temporal interpolation. In this paper, we propose a new method for the fusion of global and local side information using Support Vector Machine. The global side information is generated at the decoder using global motion parameters estimated at the encoder using the Scale-Invariant Feature Transform. Experimental results show that the proposed approach can achieve a PSNR improvement of up to 1.7 dB for a GOP size of 2 and up to 3.78 dB for larger GOP sizes, with respect to the reference DISCOVER codec.

Index Terms— Distributed Video Coding, Support Vector Machine, Classification, Side Information, Rate-Distortion Performance

1. INTRODUCTION

Distributed Video Coding (DVC) is a paradigm especially fitted for emerging applications such as wireless video surveillance, multimedia sensor networks, wireless PC cameras, and mobile communications. These applications demand a low-complexity encoding process, which cannot be achieved with current standards such as MPEG-1,-2,-4 or H.264/AVC because of the computational burden of motion estimation. On the contrary, in the DVC framework, the correlation among successive frames is mainly exploited at the decoder, thus making a DVC encoder much lighter than a standard one.

The DVC foundations date back to the Slepian-Wolf theorem for lossless compression [1], which states that 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 which can be attained in the case of joint encoding and decoding. The Wyner-Ziv (WZ) theorem [2] extends the Slepian-Wolf one to the case of lossy compression of X, when Side Information (SI) Y is available at the decoder.

Based on these theoretical results, practical implementations of DVC have been proposed [3, 4]. DISCOVER codec [5, 6], based on transform domain WZ coding, is one of the most efficient and popular existing architectures. In this codec, the images of the sequence are split into two sets of frames, the key frames (KFs) and the Wyner-Ziv frames (WZFs). A Group of Pictures (GOP) of size n is a set of successive frames, one KF and $n-1$ WZFs. The KFs are independently encoded and decoded using Intra coding techniques such as H.264/AVC Intra mode. The WZFs are separately transformed using a 4×4 integer Discrete Cosine Transform (DCT). The obtained coefficients are uniformly quantized. A systematic channel code such as the Turbo code or the Low-Density Parity Check Accumulate (LDPCA) code is applied on the resulting quantized coefficients. Only the parity bits are kept, and sent to the decoder while the systematic bits are discarded.

At the decoder, the reconstructed reference frames are used to compute the SI, which is an estimation of the WZF being decoded. This estimation can be seen as a noisy version of the original WZF. Motion-Compensated Temporal Interpolation (MCTI) [7] is used to generate the SI in the DISCOVER codec. The channel decoder corrects the DCT coefficients of the SI using the parity bits requested by the decoder through the feedback channel. Finally, reconstruction and inverse 4×4 integer DCT are applied to obtain the decoded WZF.

In this paper, we propose a new fusion method to combine two SI using Support Vector Machine (SVM). The first SI is generated using MCTI as in DISCOVER codec and is referred to as MCTI SI. The second one is generated by applying global motion parameters on the decoded reference frames [8], and is referred to as Global Motion Compensation SI (GMC SI). In this context, the objective is to optionally fuse MCTI SI and GMC SI to reach the best Rate-Distortion (RD) performance. For this purpose, an SVM classifier is applied on a block basis to choose from MCTI SI and GMC SI for fusion. We further propose two approaches based on binary and linear decisions to generate SVM SI and SVMLin SI respectively.

This paper is structured as follows. First, the related work is introduced in Section 2. The combination of MCTI SI and GMC SI using SVM is depicted in Section 3. Experimental results are shown in Section 4 in order to evaluate and compare the RD performance of the proposed approach. Finally, conclusions are drawn in Section 5.

2. RELATED WORK

MCTI [7] produces an estimation of the current frame I_n by using two decoded reference frames, say I_{n-k} and I_{n+k}. It operates as follows: First, the reference frames are low-pass filtered, and a forward motion estimation between them is performed. The resulting motion vector field $V_{(n-k)\to(n+k)}$ is then split into backward and forward motion vector fields, $V_{n\to(n-k)}$ and $V_{n\to(n+k)}$. These fields are then refined (with a further block matching operation) and smoothed (using a weighted median filter). Finally they are applied to I_{n-k} and I_{n+k}, and the resulting motion-compensated images are averaged to produce the side information.

2.1. Global Motion Compensation

We proposed a new approach for GMC SI in [8]. Here, we give the main characteristics of this technique, and we improve upon it, ending up with a new SI generation algorithm. The approach in [8] is the following: First, the feature points of the original WZ and reference frames are extracted 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 to 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 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 $\hat{R}_F$.

Using this algorithm we have two SI frames for the current frame, therefore a technique for fusion is needed. In [8], 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 [8] 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 as 'SAD-Fusion'.

2.2. Improved Side Information Generation

The SI is usually generated through an interpolation of the backward and forward reference frames. The quality of the SI is poor in certain regions of the video scene, like in areas of partial occlusions, fast motion, etc. In VISNET II codec [9], the refinement process of the SI is carried out after decoding all DCT bands, and a deblocking filter is used. In [10, 11], approaches are proposed for transform-domain DVC based on the successive refinement of the SI after each decoded DCT band. High-order motion interpolation has been proposed [12] in order to cope with object motion with non-zero acceleration. In [13], a solution is proposed by sending a hash information of the current WZF. A genetic algorithm is carried out using the hash information to merge multiple SI at the decoder. A DVC scheme proposed by Dufaux *et al.* [14] 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.

On the contrary, 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 authors in [15] proposed new techniques for the fusion of SI_t and SI_v. Inspired from [15], a linear fusion of GMC SI and MCTI SI is proposed as follows:

$$\text{SI}(b) = \frac{f_{\text{MCTI}} \cdot (\text{GMC SI}) + f_{\text{GMC}} \cdot (\text{MCTI SI})}{(f_{\text{GMC}} + f_{\text{MCTI}})} \tag{3}$$

This method is referred to as 'FusLin'. Dufaux [16] proposed a solution that consists in combining SI_t and SI_v using SVM. In this paper, we extend and improve the method of [16] for monoview DVC.

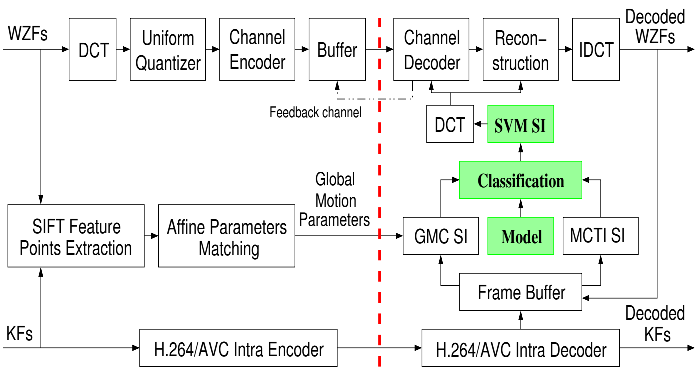

Fig. 1. Overall structure of the proposed DVC codec.

3. PROPOSED METHOD

The block diagram of our proposed codec architecture is depicted in Figure 1. It is based on the DISCOVER codec [5, 6]. The shaded (green) blocks correspond to the three new modules introduced in this paper: Model, Classification, and generating of SVM SI.

Each block in the SI can be predicted from either GMC SI or MCTI SI using the SVM classifier. In this paper, we use SVM$^{\text{Light}}$ software implementation [17]. Several kernels have been investigated, without a notable impact on performance. Therefore, a linear kernel is used hereafter.

The training stage to generate the model is described with the classification procedure in Subsection 3.1. Finally, the proposed methods for the combination of GMC SI and MCTI SI based on the predicted value by the SVM classifier is described in Subsection 3.2.

3.1. Model and Classification

First, we select the most discriminative features to be used in SVM. For this reason, three features are estimated in the proposed method as follows:

$$\begin{cases} f_1 = f_{\text{GMC}} \\ f_2 = f_{\text{MCTI}} \\ f_3 = f_{\text{GMC}} - f_{\text{MCTI}} \end{cases} \tag{4}$$

where f_{GMC} and f_{MCTI} are defined in Eq. 1. Note that different types of features have been considered but we retain in this paper the three ones (Eq. 4) which give the best results.

In the training stage, the first WZF is encoded using H.264/AVC Intra mode as the KFs. This frame is used to build the model for SVM. For each 4×4 block b, D_{GMC} and D_{MCTI} are computed according to:

$$\begin{aligned} D_{\text{GMC}} &= |\text{WZF}(p) - \text{GMC SI}(p)| \\ D_{\text{MCTI}} &= |\text{WZF}(p) - \text{MCTI SI}(p)| \end{aligned} \tag{5}$$

More precisely, D_{GMC} and D_{MCTI} are the SADs between the WZF and the GMC SI and MCTI SI for the block b respectively.

The block b is assigned to GMC SI if D_{GMC} is smaller than D_{MCTI}, and is assigned to MCTI SI otherwise. Only the N blocks which give the largest difference D ($D = |D_{\text{GMC}} - D_{\text{MCTI}}|$) are taken in the training stage. This step allows to increase the accuracy of the training stage. In our experiments, N has been empirically set to 300 blocks (about 20% of the total blocks). However, the actual value of N has a slight impact on the RD performance of the proposed method.

The features (f_1, f_2 and f_3) are computed for those selected blocks ($N = 300$ blocks), and used in the training step, in order to create the first model for the classification. Next, the classification procedure is carried out on the first WZF using this model. The blocks which are well-classified are taken into account for a second learning stage, in order to produce the final model (i.e. find the hyperplane that optimally separates the blocks of GMC SI and MCTI SI). This model will then be used in the classification procedure for all WZFs in the sequence.

In the classification, three features f_1, f_2, and f_3 are computed for each WZF using GMC SI and MCTI SI. The SVM classifier computes a predicted value for each block based on the features and the obtained model.

3.2. Proposed fusion

The SVM classifier gives a decision value d for each block. d represents the distance between this block and the separating hyperplane. Based on this value, we define two fusion algorithms. The first algorithm consists of a binary combination of GMC SI and MCTI SI. The second algorithm linearly combines the two SI.

SVM binary fusion - In this method, the value d is directly used to combine the two SI as follows:

$$\text{SI}(b) = \begin{cases} \text{GMC SI} & \text{if } d > 0 \\ \text{MCTI SI} & \text{otherwise} \end{cases} \tag{6}$$

where d represents the classification label at block b. This method is referred to as 'SVM'.

SVM linear fusion - This method aims at combining linearly GMC SI and MCTI SI. The linear combination is defined as follows:

$$\text{SI}(b) = \begin{cases} \text{GMC SI} & \text{if } d > T \\ \text{MCTI SI} & \text{if } d < (-T) \\ \frac{(T+d)\cdot\text{GMC SI}+(T-d)\cdot\text{MCTI SI}}{2\cdot T} & \text{if } |d| \le T \end{cases} \tag{7}$$

where T represents a threshold. In our experiments T has been empirically set to 3. This method is referred to as 'SVM-Lin'.

Oracle fusion - This method is impractical, but it aims at estimating the upper bound limit that can be achieved by

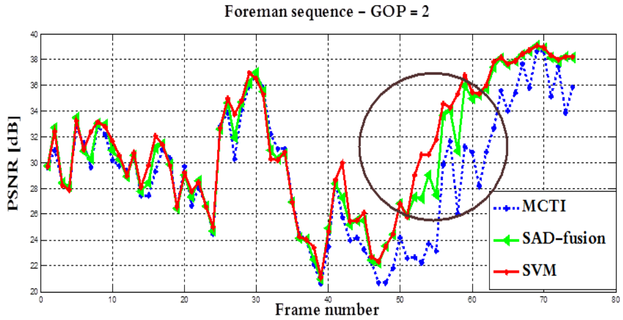

Fig. 2. PSNR of MCTI SI, SAD-fusion, and the proposed method SVM for Foreman sequence for a GOP size of 2.

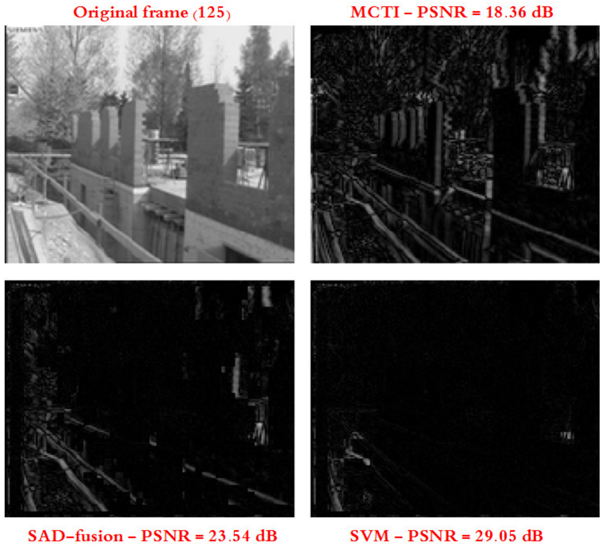

Fig. 3. Visual difference of the SI estimated by MCTI, SAD-fusion, and the proposed method SVM for frame number 125 of Foreman sequence, for a GOP size of 8 (QI = 8).

combining GMC SI and MCTI SI, using the original WZF. This fusion is defined as follows:

$$\mathrm{SI}(b) = \begin{cases} \text{GMC SI} & \text{if } D_{\mathrm{GMC}} < D_{\mathrm{MCTI}} \\ \text{MCTI SI} & \text{otherwise} \end{cases} \quad (8)$$

D_{GMC} and D_{MCTI} are introduced in Eq. 5. This method is referred to as 'Oracle'.

4. EXPERIMENTAL RESULTS

In order to evaluate the performance of the proposed methods, we performed extensive simulations, adopting the same test conditions as described in DISCOVER [5, 6], *i.e.* test video sequences are at QCIF spatial resolution and sampled at 15 frames/sec. The obtained results of the proposed methods SVM (Eq. 6) and SVMLin (Eq. 7) are compared to the DISCOVER codec, to the SAD-fusion (Eq. 2), to the linear fusion (Eq. 3), and to 'Oracle' fusion (Eq. 8).

SI Average PSNR [dB]							
Method	MCTI	GMC	SAD-F	FusLin	SVM	SVMLin	Oracle
Sequence				**GOP = 2**			
Stefan	22.57	25.88	26.27	26.19	26.45	**26.54**	27.21
Foreman	29.31	30.70	30.77	30.97	31.21	**31.30**	31.90
Bus	24.72	22.99	26.96	26.83	26.92	**27.18**	27.94
Coastguard	31.43	29.28	32.02	31.95	32.11	**32.23**	32.62
				GOP = 4			
Stefan	21.28	25.27	25.33	25.23	25.59	**25.66**	26.47
Foreman	27.58	29.62	29.24	29.47	29.77	**29.87**	30.72
Bus	23.48	22.41	25.93	25.88	25.91	**26.14**	26.91
Coastguard	29.85	28.19	30.78	30.76	30.90	**31.03**	31.46
				GOP = 8			
Stefan	20.64	24.85	24.79	24.71	25.06	**25.15**	25.99
Foreman	26.24	28.62	28.08	28.30	28.68	**28.79**	29.69
Bus	22.53	21.84	24.95	24.95	24.95	**25.17**	25.90
Coastguard	28.75	27.50	29.85	29.87	29.97	**30.10**	30.60

Table 1. SI average PSNR for a GOP size equal to 2, 4, and 8 (QI = 8).

4.1. SI performance assessment

Figure 2 shows the SI PSNR for Foreman sequence, for a GOP size of 2. The proposed method (SVM) allows a consistent improvement, compared to the previous fusion (SAD-fusion), and achieves a gain up to 4.4 dB for some frames.

Figure 3 shows the visual difference of the SI for Foreman (frame number 125), for a GOP size of 8. The SI obtained by MCTI technique is not good as shown in this figure (top-right - 18.36 dB). On the contrary, the SI obtained by the proposed method (SVM) is significantly better than the SI estimated by both MCTI and SAD-fusion. The gain is up to 5.5 dB compared to the previous SAD-fusion method for this frame.

Table 1 shows the average PSNR of the SI obtained with the different methods, for different sequences and different GOP sizes. The proposed technique (SVMLin) leads to the best SI quality for all test sequences.

4.2. Rate Distortion Performance

The RD performance is shown for the Stefan, Foreman, Bus, and Coastguard sequences in Table 2, in comparison to the DISCOVER codec, using the Bjontegaard metric [18], for a GOP size equal to 2, 4 and 8.

The proposed method SVMLin always achieves a gain compared to the other fusion methods for Foreman, Bus and Coastguard sequences, for all GOP sizes. For Stefan sequence, the proposed method SVM achieves the best performance among fusion methods, for all GOP sizes.

It is clear that the performance of the proposed fusion becomes closer to that of 'Oracle' fusion, for all test sequences. The difference between them is smaller than 0.5 dB for all GOP sizes.

The gains become even more significant for a GOP size equal to 8. In fact, for SVM, we obtain a bit reduction up to -52.46%, which corresponds to an improvement of 3.78 dB on the decoded frames w.r.t. DISCOVER codec, for Stefan sequence. For Foreman sequence, the proposed method SVM-

Method	GMC	SAD-F	FusLin	SVM	SVMLin	Oracle
	GOP = 2					
	Stefan					
Δ_R [%]	-25.59	-24.49	-21.38	**-25.70**	-25.45	-27.43
Δ_{PSNR} [dB]	1.70	1.61	1.37	**1.70**	1.68	1.84
	Foreman					
Δ_R [%]	-8.90	-7.90	-9.46	-11.31	**-12.02**	-14.30
Δ_{PSNR} [dB]	0.53	0.46	0.55	0.68	**0.72**	0.86
	Bus					
Δ_R [%]	5.02	-13.42	-10.05	-13.05	**-14.09**	-17.09
Δ_{PSNR} [dB]	-0.25	0.80	0.59	0.79	**0.84**	1.03
	Coastguard					
Δ_R [%]	9.97	-4.94	-3.71	-5.70	**-6.32**	-8.20
Δ_{PSNR} [dB]	-0.46	0.25	0.18	0.28	**0.31**	0.42
	GOP = 4					
	Stefan					
Δ_R [%]	**-45.52**	-43.12	-37.55	-45.09	-44.51	-47.81
Δ_{PSNR} [dB]	**3.16**	2.94	2.46	3.13	3.07	3.38
	Foreman					
Δ_R [%]	-22.77	-16.03	-18.58	-23.58	**-24.61**	-29.85
Δ_{PSNR} [dB]	1.33	0.90	1.05	1.38	**1.43**	1.78
	Bus					
Δ_R [%]	-2.74	-25.80	-21.74	-26.08	**-26.99**	-31.37
Δ_{PSNR} [dB]	0.16	1.52	1.26	1.54	**1.60**	1.90
	Coastguard					
Δ_R [%]	6.64	-16.34	-14.43	-18.45	**-19.28**	-24.01
Δ_{PSNR} [dB]	-0.29	0.67	0.58	0.77	**0.81**	1.04
	GOP = 8					
	Stefan					
Δ_R [%]	**-53.02**	-50.35	-44.18	-52.46	-51.99	-55.90
Δ_{PSNR} [dB]	**3.83**	3.55	2.98	3.78	3.73	4.11
	Foreman					
Δ_R [%]	-32.68	-22.77	-26.16	-32.82	**-34.20**	-39.86
Δ_{PSNR} [dB]	1.93	1.26	1.45	1.93	**2.01**	2.42
	Bus					
Δ_R [%]	-11.49	-32.33	-28.55	-32.14	**-33.24**	-38.56
Δ_{PSNR} [dB]	0.58	1.88	1.62	1.89	**1.96**	2.34
	Coastguard					
Δ_R [%]	-7.95	-28.14	-26.50	-31.64	**-32.45**	-39.02
Δ_{PSNR} [dB]	0.27	1.20	1.09	1.37	**1.41**	1.76

Table 2. 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.

Lin allows a gain of up to 2.01 dB, with a rate reduction of 34.20%, compared to the DISCOVER codec, while the SAD-fusion method allows a gain up to 1.26 dB, with a rate reduction of 22.77%, compared to the DISCOVER codec.

5. CONCLUSION

A new technique based on SVM for the fusion of global and local SI is proposed in this paper. Experimental results show that our proposed method can achieve a gain in RD performance up to 1.7 dB for a GOP size of 2 and 3.78 dB for longer GOP sizes, compared to DISCOVER codec, especially when the video sequence contains high motion.

6. REFERENCES

[1] 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.

[2] 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.

[3] 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.

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

[5] 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.

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

[7] 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.

[8] 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 (in press)*.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] T. Maugey, C. Yaacoub, J. Farah, M. Cagnazzo, and B. Pesquet-Popescu, "Side information enhancement using an adaptive hash-based genetic algorithm in a Wyner-Ziv context," in *IEEE International Workshop on Multimedia Signal Processing*, Saint-Malo, France, Oct. 2010, pp. 298 –302.

[14] 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.

[15] 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.

[16] 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.

[17] "SVM implementation," http://www.cs.cornell.edu/People/tj/svm_light/.

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