A DIFFERENTIAL MOTION ESTIMATION METHOD FOR IMAGE INTERPOLATION IN DISTRIBUTED VIDEO CODING

Marco Cagnazzo, Thomas Maugey, Béatrice Pesquet-Popescu

TELECOM-ParisTech, TSI department
46 rue Barrault, F-75634 Paris Cedex 13, FRANCE

ABSTRACT

Motion estimation methods based on differential techniques proved to be very useful in the context of video analysis, but have a limited employment in classical video compression because, though accurate, the dense motion vector field they produce requires too much coding resource and computational effort. On the contrary, this kind of algorithm could be useful in the framework of distributed video coding (DVC). In this paper we propose a differential motion estimation algorithm which can run at the decoder in a DVC scheme, without requiring any increase in coding rate. This algorithm allows a performance improvement in image interpolation with respect to state-of-the-art algorithms.

Index Terms— Distributed video coding, dense motion vector field, image interpolation, differential motion estimation

1. INTRODUCTION

Distributed Video Coding (DVC) is a new promising paradigm in video communication, which refers to the compression of multiple outputs of correlated sources which do not communicate with each other. The targeted applications are numerous, such as video compression on mobile devices, multi-sensor surveillance systems, and so on. In opposition to the classical centralized coding paradigm, in the DVC case the sensors send their compressed outputs to a central point for joint decoding. One of the most challenging problems of this new paradigm is to achieve the same compression efficiency as traditional coding. In the seminal work by Slepian and Wolf [1] it was shown that separated encoding is just as efficient as joint encoding for lossless compression, if an increase in decoder complexity is allowed. Similar results were obtained by Wyner and Ziv [2] in lossy coding of jointly Gaussian sources. Unfortunately the arguments at the basis of the distributed coding theory are asymptotic and non-constructive, so many efforts have been devoted to the search of techniques able to achieve these theoretical limits.

Two main approaches have been proposed for DVC: PRISM [3] and Stanford [4]. In our work we consider the second one, characterized by the splitting of the video sequence into two subsets, the key frames (KF) and the Wyner-Ziv frames (WZF). KFs and WZFs alternate within each group of pictures (GOP) so that the two subsets can be seen as correlated sources. The number of WZFs can be fixed or adaptively chosen in order to optimize the coding performances, as in [5, 6]. The KFs are coded with a still image technique, such as JPEG2000, or the INTRA mode of H.263 or H.264. They are used at the decoder to generate an estimation of the WZF, called side information (SI). This SI is corrected by information coming from the Wyner-Ziv encoder which is constituted by (possibly) a discrete transform (DCT or DWT), a quantizer, and a channel encoder (LDPC or turbo-coder). The WZ coder sends only the parity bits, and at the receiver a channel decoder corrects the unavoidable estimation errors as they were channel errors induced by noise. The turbo decoding uses a feedback channel to set the rate of parity bits.

In such a framework, DVC performances strongly depend on the SI quality at the decoder. Though several methods have been proposed in literature, the most popular ones use block matching (BM) motion estimation (ME) and compensation [7, 8]. Though successful in classical video coding, BM algorithms are not the only approach to ME: gradient techniques, along with the derived class of pel-recursive (PR) techniques, have been developed for video analysis and solve the optical flow problem using a differential approach [9]. Frequency domain ME techniques also exist, but they proved less successful. Gradient and PR methods produce a dense motion vector field (MVF), which is unsuited for the classical video coding paradigm, since it requires huge resources to be encoded. On the contrary, it could well fit the DVC paradigm if the dense MVF is estimated only at the decoder side from the KFs, so that there is no need to send it. In this way the SI interpolation benefits from the increased accuracy of a dense MVF without having to pay the cost of an exceedingly high coding rate.

In particular, in this paper we propose a PR algorithm derived from the Cafforio-Rocca (CR) one [10], which is among the most popular PR ME techniques and can be adapted to the DVC context. Our experiments show remarkable gains in the quality of the SI and in the global RD performance, suggesting that differential ME techniques can be effectively employed in DVC.

The rest of the paper is organized as follows: after presenting the architecture of the DVC coder in Section 2, we recall the original CR algorithm in Section 3, and we describe the proposed algorithm in Section 4. Experimental results are shown in Section 5, while Section 6 draws conclusions and outlines future developments.

2. SYSTEM ARCHITECTURE

The coding scheme adopted in this work is similar to the DISCOVER one [8], also used in [11]. The KFs are coded using the Intra mode of H.264 with an assigned quantization step QP. The WZF DCT coefficients are computed and then quantized and turbo-encoded. Only parity bits are sent to the turbo decoder, where they are used to correct the side information.

In the DISCOVER method, at the decoder side, an estimation of the current WZF, let it be I_k is produced by using the adjacent KFs, let them be I_{k-1} and I_{k+1}. After a spatial smoothing of the two KFs, a BM ME is performed between them. The produced MVF is used to estimate the backward MVF $\mathbf{v}_B$ between I_{k+1} and I_k, and the forward MVF $\mathbf{v}_F$ between I_{k-1} and I_k. Finally a weighted median filter is then applied on $\mathbf{v}_B$ and $\mathbf{v}_F$ in order to eliminate the outliers and to get a smooth solution. The WZF estimation is the average
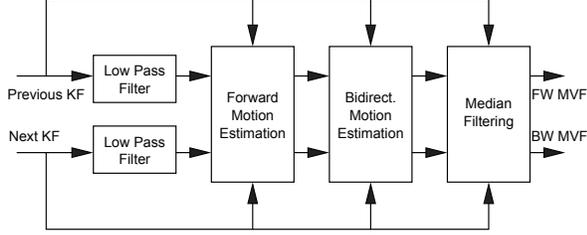
Fig. 1. DISCOVER motion interpolation method.

between I_{k+1} compensated by $\mathbf{v}_B$ and I_{k-1} compensated by $\mathbf{v}_F$.

In this paper we propose an algorithm to further improve the MVFs produced by the DISCOVER method. It takes as inputs the adjacent KFs I_{k-1} and I_{k+1}, and the related MVFs, $\mathbf{v}_B$ and $\mathbf{v}_F$, and produces a new and dense version of both the MVFs, which allows to improve the quality of the interpolated WZ frame.

3. THE CAFFORIO-ROCCA ALGORITHM

The CR ME algorithm is *pel-recursive*, meaning that the MV computed for the last pixel (or more generally, a function of the previously estimated MVs) is used as initialization for the next pixel to be processed. The pixels are not necessary scanned in raster order; rather, an order that better preserves the correlation between successively processed pixels is often preferred, *e.g.* by scanning the even lines from the left to the right and the odd ones from the right to the left.

The CR algorithm consists in applying, for each pixel $\mathbf{p}$ in the scan order, three steps so that the output vector $\mathbf{v}(\mathbf{p})$ is obtained. As mentioned before, in the first step, the motion vector is initialized with a function of the previously computed vectors. The initialization value is sometimes called *a priori* and is denoted by $\mathbf{v}^{(0)}$. The second step is called validation and consists in comparing $\mathbf{v}^{(0)}$ with the null vector. If the prediction error for $\mathbf{v}^{(0)}$ is less than the one for the null vector possibly incremented by a threshold γ, the *a priori* is retained; otherwise, the null vector is retained. A high threshold favors the regularity of the MVF (less MVs are reset to zero), a low one favors a better prediction. The validation step is important when adjacent pixels have quite different MVs (*e.g.* because they belong to different objects). In this case the previous MV must not be used to initialize the current position.

The last step consists in refining the validated vector $\mathbf{v}^{(1)}$ by adding to it a correction $\delta\mathbf{v}$. This correction is obtained by minimizing the energy of the prediction error, under a constraint on the norm of the correction vector. The Lagrangian cost function is then: $J(\delta\mathbf{v}) = [I_k(\mathbf{p}) - I_{k-1}(\mathbf{p} + \mathbf{v}^{(1)} + \delta\mathbf{v})]^2 + \lambda\|\delta\mathbf{v}\|^2$ Using a first order expansion of I_{k-1}, it turns out that the value of $\delta\mathbf{v}$ minimizing J is:

$$\delta\mathbf{v}(\mathbf{p}) = \frac{-\epsilon\boldsymbol{\varphi}}{\lambda + \|\boldsymbol{\varphi}\|^2} \qquad (1)$$

where $\epsilon = I_k(\mathbf{p}) - I_{k-1}(\mathbf{p} + \mathbf{v}^{(1)})$ is the prediction error associated to the MV $\mathbf{v}^{(1)}$, and $\boldsymbol{\varphi} = \nabla I_{k-1}(\mathbf{p} + \mathbf{v}^{(1)})$ is the spatial gradient of the motion-compensated reference image.

4. PROPOSED METHOD

We propose a new version of the CR algorithm, allowing to obtain better ME for Wyner-Ziv frames in the context of DVC. In this new version we should estimate the movement from current frametoward

I_{k-1} and I_{k+1}, without disposing of the frame I_k, but only of some encoded versions of the adjacent frames I_{k-1} and I_{k+1}. Moreover we want to exploit the block-based MVFs produced by the DISCOVER algorithm, $\mathbf{v}_B$ and $\mathbf{v}_F$.

Our ME algorithm still consists in the initialization, validation and refinement steps; but they are modified to fit the new context; moreover we use a different scanning order, based on the blocks used in the DISCOVER algorithm. A raster scan order between blocks can be used, however it is worth noting that the blocks are processed independently, so the algorithm lends itself to a parallel implementation. Within each block the pels are also scanned in raster order.

The initialization of $\mathbf{v}_B(\mathbf{p})$ and $\mathbf{v}_F(\mathbf{p})$ is different if $\mathbf{p}$ is the first position (*i.e.* top and leftmost) in the block or not. In the first case, we use the MVs estimated for the current block by the DISCOVER algorithm, $\mathbf{v}_B^{\mathrm{DISC}}$ and $\mathbf{v}_F^{\mathrm{DISC}}$; otherwise, we use a weighted average of the left, up, and up-right neighboring vectors, with different weights if the neighbors are in the same block or not. We call $\mathbf{v}_B^{(0)}(\mathbf{p})$ and $\mathbf{v}_F^{(0)}(\mathbf{p})$ (or *a priori*) the backward and forward vectors obtained from the initialization step.

The validation step amounts to compute the motion compensated errors associated to $\mathbf{v}_{B,F}^{(0)}$, to the null vector and to $\mathbf{v}_{B,F}^{\mathrm{DISC}}$, and to choose those with the least error. More precisely we compute the following quantities:

$$A = \left| I_{k+1}(\mathbf{p} + \mathbf{v}_B^{(0)}(\mathbf{p})) - I_{k-1}(\mathbf{p} + \mathbf{v}_F^{(0)}(\mathbf{p})) \right|$$
$$B = |I_{k+1}(\mathbf{p}) - I_{k-1}(\mathbf{p})| + \gamma,$$
$$C = \left| I_{k+1}(\mathbf{p} + \mathbf{v}_B^{\mathrm{DISC}}) - I_{k-1}(\mathbf{p} + \mathbf{v}_F^{\mathrm{DISC}}) \right|,$$

If A [resp. B, C] is the least quantity, we use $\mathbf{v}_{B,F}^{(0)}$ [resp. the null vector, $\mathbf{v}_{B,F}^{\mathrm{DISC}}$] as validated vectors. Note that, like for the original CR algorithm, a threshold is used to penalize the reset of the estimated vector. A high threshold causes less vector resets, producing more regular but maybe less accurate MVFs.

In the last step, we refine the validated MVs $\mathbf{v}_B^{(1)}$ and $\mathbf{v}_F^{(1)}$ by adding a correction ($\delta\mathbf{v}_B$ and $\delta\mathbf{v}_F$). So the cost function J depends on both refinements:

$$J(\delta\mathbf{v}_B, \delta\mathbf{v}_F) = [I_{k+1}(\mathbf{p} + \mathbf{v}_B^{(1)} + \delta\mathbf{v}_B) +$$
$$- I_{k-1}(\mathbf{p} + \mathbf{v}_F^{(1)} + \delta\mathbf{v}_F)]^2 + \lambda_B\|\delta\mathbf{v}_B\|^2 + \lambda_F\|\delta\mathbf{v}_F\|^2$$

Like in the original algorithm, the cost function is approximated by first order expansions; however we expand both I_{k+1} and I_{k-1}:

$$J \approx [I_{k+1}(\mathbf{p} + \mathbf{v}_B^{(1)}) + \nabla I_{k+1}(\mathbf{p} + \mathbf{v}_B^{(1)})^T\delta\mathbf{v}_B +$$
$$- I_{k-1}(\mathbf{p} + \mathbf{v}_F^{(1)}) - \nabla I_{k-1}(\mathbf{p} + \mathbf{v}_F^{(1)})^T\delta\mathbf{v}_F]^2 +$$
$$+ \lambda_B\|\delta\mathbf{v}_B\|^2 + \lambda_F\|\delta\mathbf{v}_F\|^2$$
$$= \left(\epsilon + \boldsymbol{\varphi}_B^T\delta\mathbf{v}_B - \boldsymbol{\varphi}_F^T\delta\mathbf{v}_F\right)^2 + \lambda_B\|\delta\mathbf{v}_B\|^2 + \lambda_F\|\delta\mathbf{v}_F\|^2$$

where we defined:

$$\epsilon = I_{k+1}(\mathbf{p} + \mathbf{v}_B^{(1)}) - I_{k-1}(\mathbf{p} + \mathbf{v}_F^{(1)})$$
$$\boldsymbol{\varphi}_B = \nabla I_{k+1}(\mathbf{p} + \mathbf{v}_B^{(1)})$$
$$\boldsymbol{\varphi}_F = \nabla I_{k+1}(\mathbf{p} + \mathbf{v}_F^{(1)})$$

Then, the actual refinements are defined as those minimizing the function cost and are found by setting to zero the partial derivatives

of J. Let us start with the derivative with respect to $\delta\mathbf{v}_B$.

$$\frac{\partial J}{\partial \delta\mathbf{v}_B} = \mathbf{0} \Leftrightarrow$$

$$2[\epsilon + \boldsymbol{\varphi}_B^T \delta\mathbf{v}_B - \boldsymbol{\varphi}_F^T \delta\mathbf{v}_F]\boldsymbol{\varphi}_B + 2\lambda_B \delta\mathbf{v}_B = \mathbf{0} \Leftrightarrow$$

$$[\epsilon - \boldsymbol{\varphi}_F^T \delta\mathbf{v}_F]\boldsymbol{\varphi}_B + (\lambda_B \mathbf{I}_2 + \boldsymbol{\varphi}_B \boldsymbol{\varphi}_B^T)\delta\mathbf{v}_B = \mathbf{0} \Leftrightarrow$$

$$\delta\mathbf{v}_B = \frac{\boldsymbol{\varphi}_F^T \delta\mathbf{v}_F - \epsilon}{\lambda_B + \|\boldsymbol{\varphi}_B\|^2}\boldsymbol{\varphi}_B \tag{2}$$

We use $\mathbf{I}_2$ to refer to the 2×2 identity matrix. The last equation has been obtained applying the matrix inversion lemma. Likewise, the partial derivative of J with respect to $\delta\mathbf{v}_B$ is zero iff:

$$\delta\mathbf{v}_F = \frac{\boldsymbol{\varphi}_B^T \delta\mathbf{v}_B + \epsilon}{\lambda_F + \|\boldsymbol{\varphi}_F\|^2}\boldsymbol{\varphi}_F \tag{3}$$

Substituting Eq. (3). in (2), and applying again the matrix inversion lemma, we can easily find the optimal refinements:

$$\delta\mathbf{v}_B^* = \frac{-\epsilon\boldsymbol{\varphi}_B}{\lambda_B + \|\boldsymbol{\varphi}_B\|^2 + \frac{\lambda_B}{\lambda_F}\|\boldsymbol{\varphi}_F\|^2} \tag{4}$$

$$\delta\mathbf{v}_F^* = \frac{\epsilon\boldsymbol{\varphi}_F}{\lambda_F + \|\boldsymbol{\varphi}_F\|^2 + \frac{\lambda_F}{\lambda_B}\|\boldsymbol{\varphi}_B\|^2}. \tag{5}$$

Previous equations further simplify since usually $\lambda_B = \lambda_F$ (see Section 5.1). The resulting final formulas are formally very similar to those of the original algorithm, see Eq. (1).

Finally we note that the proposed algorithm can adapt very easily to GOP sizes larger than 2. For example, for a GOP size equal to 4, we just use three times the same method: the first one in order to interpolate I_k from KFs I_{k-2} and I_{k+2}; then we use it again to interpolate I_{k-1} from I_{k-2} and I_k and I_{k+1} from I_k and I_{k+2}.

5. EXPERIMENTAL RESULTS

The proposed method has been implemented in order to improve the quality of the SI. We have performed several experiments in order to validate it, using four popular test sequences at CIF resolution and 30 fps: *foreman*, *football*, *eric* and *city*. We have run the proposed algorithm using as input the MVs and the KFs generated by the DIS-COVER algorithm for various QPs, and we have produced the new MVs and the corresponding interpolated SI. In a first set of experiments (see Sect. 5.1), we have determined the values to be used for the parameters λ_F, λ_B and γ; in a second set (see Sect. 5.2), we have compared the new interpolated WZ frames with those obtained with the reference algorithm; finally in a third one (see Sect. 5.3), we have integrated the new method in the DVC scheme described in Sect. 2 in order to evaluate the impact over global RD performances.
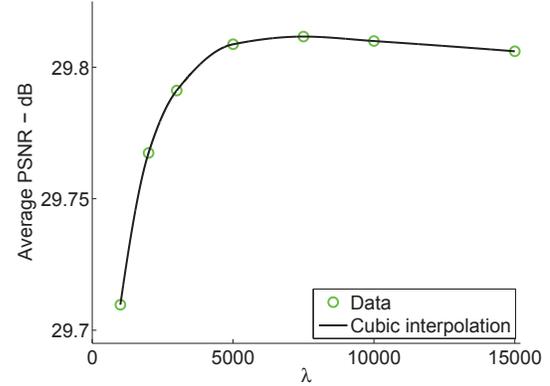
5.1. Parameter tuning

In order to determine the best value for the parameters of the proposed algorithm, we have run it over the test sequences and we have obtained the interpolation of even frames. These images were compared with the original frames by computing the PSNR.

In all our experiments, the threshold γ proved to have a small influence over the global performance, given that it is greater or equal than 50, so we used this value for the following.

Then we determined the relationship between the best λ_F and λ_B. The experiments confirmed the intuition that they should have very close values. In all the experiments we found that the best performance is obtained when $|\lambda_F - \lambda_B| < 0.1\lambda_B$; moreover, within

$\Delta\lambda$	-1000	-500	0	500	1000
eric	32.29	32.33	32.33	32.32	32.32
football	23.19	23.19	23.19	23.17	23.17
foreman	33.86	33.89	33.90	33.89	33.89
city	27.15	27.16	27.17	27.16	27.15
Average	29.12	29.14	**29.15**	29.14	29.13

Table 1. PSNR of SI images over the test sequences for different $\Delta\lambda = \lambda_F - \lambda_B$ and QP=31. Average over $\lambda_B \in [1000, 15000]$

Fig. 2. Average PSNR of side information over test sequences as a function of λ, for QP=31.
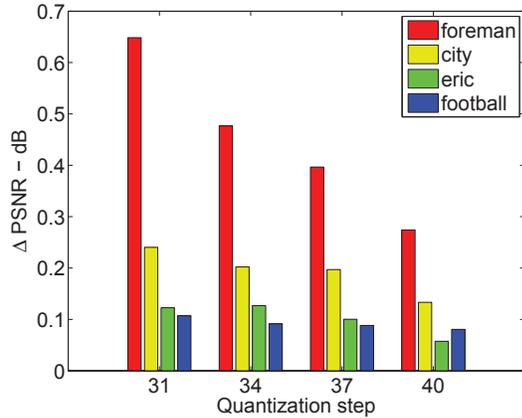
this interval the performances are very consistent, with a PSNR variation less than 0.03 dB. For the sake of brevity, we only report some of these results in Tab. 1, where for a fixed QP value, and for different $\Delta\lambda = \lambda_F - \lambda_B$ we reported the SI PSNR for each sequence, averaged over the λ_B values. As a consequence of these results, in the following we take $\lambda_F = \lambda_B$ and so we will drop the subscript.

Finally, we looked for the best value of λ. We have computed the SI PSNR over the test sequences for several values of the parameter between 1000 and 15000. As shown in Fig. 2 the average PSNR performance are quite consistent for $\lambda \geq 5000$, with a maximum around 7500, which has been used as value for λ in the following.

5.2. Side information improvement

With the values of parameters defined in the previous subsection, we have compared our algorithm with DISCOVER by running them over the same test sequences and using several QPs for the KF coding. The results are summarized in Fig. 3. We observe that the proposed method is able to improve the WZF quality, up to over 0.6 dB in the average and to over 2 dB on the single image. The best results have been obtained for the *foreman* sequence, characterized by a complex motion. The gain is still interesting for the sequence *city*, characterized by a more regular motion. Smaller gains are obtained when the movement is more irregular (*football*) and for the "head and shoulder" sequence *eric*. We observe as well that the gain is smaller for severely quantized KFs: this is reasonable since low quality KFs provide a less reliable gradient information, which is at the basis of the proposed method.

A further experiment was conducted in order to assess the efficiency of the proposed technique when larger GOP sizes are used. We performed a comparison similar to the one reported in Fig. 3, the only difference being the distance among the key frames. The re-

Fig. 3. SI PSNR improvement between reference and proposed method

GOP size	*foreman*	*city*	*eric*	*football*
2	0.65	0.24	0.12	0.11
4	0.46	0.28	0.13	0.11
8	0.39	0.28	0.12	0.12

Table 2. SI PSNR improvement for different GOP sizes [dB]

sults are reported in Tab. 2. It is interesting to observe that the PSNR improvement w.r.t. the reference method is quite consistent even for large GOP sizes.

5.3. Distributed video coding test

In the last set of experiments, we used the new SI within the global DVC scheme, and compute the global RD performance for QP=31, 34, 37, 40. This was compared with the RD performance over the test sequences of the reference DISCOVER coder, and the results are reported using the Bjontegard metric [12], recommended by the VCEG for comparing video coding methods, for the same four QPs. As shown in Tab. 3, the proposed method allows some interesting rate reductions (3.5% for *foreman* and 2.0% on the average). The PSNR improvement is less than the one we have found on the sole side information. This is reasonable since this time PSNR is computed on the KFs as well, which are identical for the two schemes.

We conclude this section with some remark on complexity. The proposed method is of course more demanding in computation than the reference DISCOVER, since we have to compute a dense motion vector field instead of a vector per macroblock. However the nature of the algorithm is fully parallel and each macroblock could be processed independently. Finally, the complexity issue is further mitigated by the fact that only the decoder is concerned, which is not critical in DVC.

6. CONCLUSIONS AND FUTURE WORK

In this work we presented a new method for motion-compensated image interpolation in the framework of DVC. This method is based on a pel-recursive ME algorithm, which can improve the couple of MVFs used to produce the interpolation. The technique was implemented upon the popular DISCOVER algorithm, which actually acts as an initialization for the proposed scheme.

	foreman	*city*	*eric*	*football*	Average
ΔRate	-3.52%	-1.97%	-1.02%	-1.53%	-2.01%
ΔPSNR	0.18	0.10	0.06	0.08	0.10

Table 3. Average RD performances improvement (Bjontegard metric [12]) with respect to the DISCOVER scheme.

The experiments showed that, after a suitable parameter tuning, the proposed technique allows remarkable gain in terms of SI quality: we have observed up to over 0.6 dB of improvement in PSNR with respect to the reference DISCOVER technique. Moreover we used the new SI in our DVC scheme and observed that gains are interesting in this case as well, up to 3.5% in rate reduction. These results show the interest of using a dense MVF in the framework of DVC. While in the classical video coding paradigm a dense MVF suffer from an exceeding coding cost and computational complexity, as far as DVC is concerned, we can obtain a good quality dense MVF at the decoder side using the key frames. This approach allows to entirely overcome the drawbacks that, in the classical video coding paradigm, have prevented a successful application of differential based ME techniques.

Future work will focus on further application of differential ME techniques and dense MVFs for DVC. Currently, studies are under way for using some differential ME in the early stage of the DISCOVER scheme as well.

7. REFERENCES

[1] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Inform. Theory*, vol. 19, pp. 471–480, July 1973.

[2] A. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the receiver," *IEEE Trans. Inform. Theory*, vol. 22, pp. 1–11, Jan. 1976.

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

[4] A. Aaron, R. Zhang, and B. Girod, "Wyner-Ziv coding of motion video," in *Asilomar Conference on Signals and Systems*, Pacific Grove, California, Nov. 2002.

[5] J. Ascenso, C. Brites, and F. Pereira, "Content adaptive Wyner-Ziv video coding driven by motion activity," in *Proceed. of IEEE Intern. Conf. Image Proc.*, Atlanta, GA, USA, July 2006, pp. 605–608.

[6] C. Yaacoub, Joumana Farah, and Batrice Pesquet-Popescu, "Feedback channel suppression in distributed video coding with adaptive rate allocation and quantization for multi-user applications," *EURASIP Journal on Wireless Communications and Networking*, 2008, to appear.

[7] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proc. IEEE*, vol. 93, no. 71, pp. 71–83, Jan. 2005.

[8] C. Guillemot, F. Pereira, L. Torres, T. Ebrahimi, R. Leonardi, and J. Ostermann, "Distributed monoview and multiview video coding: Basics, problems and recent advances," *IEEE Signal Processing Mag.*, pp. 67–76, Sept. 2007.

[9] B. Horn and B. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, pp. 185–203, 1981.

[10] C. Cafforio and F. Rocca, "The differential method for motion estimation," in *Image Sequence Processing and Dynamic Scene Analysis*, T. S. Huang, Ed., 1983, pp. 104–124.

[11] T. Maugey, T. Andre, B. Pesquet-Popescu, and J. Farah, "Analysis of error propagation due to frame losses in a distributed video coding system," in *Proceed. of Europ. Sign. Proc. Conf.*, Lausanne, Switzerland, Aug. 2008.

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