FAST AND EFFICIENT SIDE INFORMATION GENERATION IN DISTRIBUTED VIDEO CODING BY USING DENSE MOTION REPRESENTATIONS

Giovanni Petrazzuoli, Marco Cagnazzo, Béatrice Pesquet-Popescu

Institut Telecom – Telecom ParisTech – CNRS LTCI
46 rue Barrault, F-75634 Paris Cedex 13, FRANCE
phone: +33 (0)1 45 81 77 77, fax: +33 (0)1 45 89 79 06
email: {petrazzu, cagnazzo, pesquet}@telecom-paristech.fr

ABSTRACT

Distributed video coding (DVC) does not demand motion estimation (ME) and compensation (MC) at the encoder, but only at the decoder and so it is more suitable for applications that require a simple encoder, like wireless sensor networks. In DVC the video sequence is split into Key Frames (KFs) and Wyner-Ziv Frames (WZFs): the first are intra-coded and the latter are coded by a channel code and only the parity bits are sent to the decoder. The KFs are available at the decoder, while we need to estimate the WZF and correct this estimation with parity bits. One critical step is the estimation of the WZF. The method of the state of the art, with which we compare, is given by DISCOVER. It estimates the WZF by linear interpolating the two adjacent KFs. We propose a higher order motion interpolation for WZF estimation by using four KFs. Due to the high computational efforts, we propose also a fast algorithm that halves the complexity of the previous method. We note that the results of the fast method are comparable with the original one. An other proposal is to increase the density of the motion vector field in order to improve the estimation of the WZF.

Keywords: Distributed video coding, image interpolation

1. INTRODUCTION

Let X and Y be two correlated sources. If we encode them jointly, we can decode them without loss of information if $R_X + R_Y \geq H(X,Y)$,[1] but if we encode separately X and Y, we expect that we can decode them without loss of information if $R_X \geq H(X)$ and $R_Y \geq H(Y)$, according to the first Shannon theorem. Indeed, according to the Slepian-Wolf Theorem [5], a total rate $R_X + R_Y \geq H(X,Y)$ is sufficient, even for separated encoding of dependent sources, provided that we decode them jointly.

A particular case of distributed source coding is the source coding with side information: the variable X, generated by the source, is statistically dependent of Y. The variable Y is available at the decoder, but not at the encoder. Y is called side information. Then, if $R_Y = H(Y)$, a rate $R_X \geq H(X|Y)$ is sufficient for recovering X without loss of information, according to the Slepian-Wolf theorem.

While Slepian-Wolf theorem is about lossless coding, the problem of lossy coding was solved by Wyner and Ziv [6]. They proved that there is no rate increase in the source coding with side information with respect to the joint coding for a given quality, and conversely, there is no quality loss for a given rate, subjected to some mild constraints.

[1] Let $H(X)$ be the entropy of X and $H(X,Y)$ the joint entropy of X and Y.

These results can be applied to video coding in order to simplify the encoder structure. In Distributed Video Coding (DVC) the video sequence is divided in Group of Pictures (GOP). Each GOP consists of one Key Frame (KF) (usually the first one), that is intra-coded (i.e. it is coded independently of the other frames) and Wyner-Ziv Frames (WZFs) that are coded by a systematic channel code. We send to the decoder only the redundance bits (i.e. the parity bits). Even if the KFs and the WZFs are correlated sources, we do not exploit this dependence at the encoder, but only at the decoder. We produce an estimation of WZF by interpolating the adjacent decoded KFs. This step is called Image Interpolation and the WZF estimation is called Side Information (SI). The SI can be considered a noisy version of the WZF. Then, we correct the errors of the WZF estimation with the parity bits sent by the channel encoder. The weak point of this scheme is a feedback channel is needed to adjust the number of parity bits made by the encoder.

On the other hand, the advantage of this structure is that we move the complexity in terms of computation (the motion estimation) and in terms of memory (the exigence to storage the previous frames) from the encoder to the decoder. This is desirable if we need a low-complexity encoder, as in wireless sensor network, but it is not well suited for broadcast transmission.

The reference technique for WZF estimation is given by DISCOVER, that performs a linear interpolation between the two more adjacent KFs. In [4] we propose a higher order motion interpolation (HOMI) method by using four KFs, instead of two such as in DISCOVER. In this paper we improve those results by increasing the density of the motion vector field (MVF) for the motion estimation, and we propose also a new method (Fast HOMI) in order to reduce the complexity of our algorithm.

The rest of the paper is organized as follows. In Section 2 we describe the DISCOVER motion interpolation method. In Section 3 we illustrate the method proposed in [4] and after we propose a variant to our method in order to reduce the complexity. Experimental results are reported in Section 4, while conclusions and future work are in Section 5.

2. STATE OF THE ART: DISCOVER MOTION INTERPOLATION ALGORITHM

One of the most popular methods for image interpolation is the method proposed in the DISCOVER project [3]: it consists in a linear interpolation between two adjacent KFs. For example, if the GOP size is equal to 2, we use the KFs I_{k-1} and I_{k+1} for the estimation of the WZF I_k. The DISCOVER method consists of four steps:

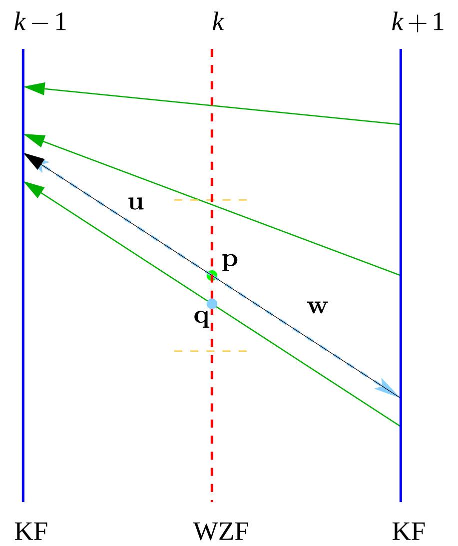

Figure 1: Bidirectional motion estimation in DISCOVER. The distance between $\mathbf{q}$ and $\mathbf{p}$ is small, such that the motion in $\mathbf{p}$ can be approximated with the motion in $\mathbf{q}$.

1. **Low Pass Filter.** The two KFs, I_{k-1} and I_{k+1}, are spatially filtered in order to reduce noise.
2. **Forward Motion Estimation.** A block matching is performed from I_{k+1} to I_{k-1}. Let $\mathbf{v}$ be the motion vector field (MVF) calculated at this step (green arrows in Fig. 1).
3. **Bidirectional motion estimation.** For each macroblock (let $\mathbf{p}$ be its center) we search the vector $\mathbf{v}$ that intercepts the frame I_k in the point closest to $\mathbf{p}$. Let $\mathbf{q}$ be this intersection. Then, the movement in $\mathbf{p}$ can be approximated as the movement in $\mathbf{q}$ (black vector in Fig.1). Afterwards, the vector is split into $\mathbf{w}$ (backward MVF) and $\mathbf{u}$ (forward MVF) (light-blue vectors in Fig. 1)
4. **Refinement and median filter.** The vectors $\mathbf{w}$ and $\mathbf{u}$ are refined around their initial positions in order to minimize the SAD (or SSD) between the MB pointed by $\mathbf{w}$ in I_{k+1} and the one pointed by $\mathbf{u}$ in I_{k-1}. Afterwards, a median filter is applied to the two MVFs in order to smooth them.

Finally, I_{k+1} is motion-compensated with $\mathbf{w}$ and I_{k-1} with $\mathbf{u}$. The average of these two predictions is the estimation of the WZF.

3. PROPOSED METHODS

In [4] we proposed a higher order motion interpolation method in order to increase the results given by DISCOVER. For the sake of clarity, we report here the basic ideas of this algorithm and after we propose some variants to this algorithm.

3.1 Higher order motion interpolation (HOMI)

While DISCOVER motion interpolation method performs a linear interpolation between the two adjacent KFs I_{k-1} and I_{k+1} (by supposing the GOP size equal to 2), we proposed in [4] a higher motion interpolation by using four KFs: I_{k-3}, I_{k-1}, I_{k+1} and I_{k+3}. Our method consists into four steps:

1. **Initialization.** By using only I_{k+1} and I_{k-1} we compute the backward and the forward MVF, respectively $\mathbf{u}$ and $\mathbf{w}$, as in DISCOVER algorithm (black dashed vectors in Fig. 2(a)).
2. **Motion estimation from $I_{k\pm1}$ to $I_{k\pm3}$.** We perform a block matching motion estimation from $I_{k\pm1}$ to $I_{k\pm3}$. Let $\mathbf{p}$ be the center of the MB in the frame I_k that we want to estimate. Let $B_k^{\mathbf{p}}$ be the MB in the frame I_k, centered in $\mathbf{p}$. Then, we search for the vector $\tilde{\mathbf{u}}$ such that the following functional is minimized:

$$J(\tilde{\mathbf{u}}) = \sum_{\mathbf{q}} \left| B_{k-1}^{\mathbf{p}+\mathbf{u}(\mathbf{p})}(\mathbf{q}) - B_{k-3}^{\mathbf{p}+\tilde{\mathbf{u}}}(\mathbf{q}) \right| + \lambda \|\tilde{\mathbf{u}} - 3\mathbf{u}\|, \quad (1)$$

where $\lambda \geq 0$ is a regularization constant. The regularization term is added for penalizing too large deviations of $\tilde{\mathbf{u}}$ from $3\mathbf{u}$, i.e. the solution that we would have supposing a linear motion along the four frames.
3. **Interpolation.** Now, we can trace the trajectory of the object along the various KFs by interpolating the positions $\mathbf{p}+\tilde{\mathbf{u}}(\mathbf{p})$, $\mathbf{p}+\mathbf{u}(\mathbf{p})$, $\mathbf{p}+\mathbf{w}(\mathbf{p})$, and $\mathbf{p}+\tilde{\mathbf{w}}(\mathbf{p})$, respectively at the instants $k-3$, $k-1$, $k+1$ and $k+3$. Then, by sampling it in k, we obtain the estimated position of the object in the frame I_k. Let $\hat{\mathbf{p}}$ be this position. We can then estimate the motion vectors centered in $\hat{\mathbf{p}}$ (red vectors in Fig. 2(a)).
4. **Vector adjustment.** We suppose that the distance between $\hat{\mathbf{p}}$ and $\mathbf{p}$ is so small, such that the motion in $\mathbf{p}$ is the same as in $\hat{\mathbf{p}}$, that we can approximate the vectors in $\mathbf{p}$ with the vectors estimated in $\hat{\mathbf{p}}$ (green dashed vectors in Fig. 2(a)).

Afterward, we repeat this for each macroblock. The average of the two compensations will be the WZF estimation.

We repeat this procedure for each macroblock. Afterward, we motion-compensate the frame I_{k+1} by the backward MVF and I_{k-1} by the forward MVF. The average of the two compensations will be the WZF estimation.

3.2 Fast HOMI

The complexity of the interpolation procedure described in the previous section can be reduced, because at the instant k, we have already estimated the forward motion vector field $\mathbf{v}$, from the frame I_{k-2} to the frame I_{k-3} and the backward motion vector field $\mathbf{z}$ from I_{k-2} to I_{k-1}. In this way, it is only necessary the motion vector field from I_{k+1} to I_{k+3} while it is not necessary to perform the motion estimation from I_{k-1} to I_{k-3}: we can exploit the MVF computed at instant $k-2$. This method is less robust than the first one, because instead of the motion estimation from I_{k-1} to I_{k-3}, we just exploit the estimation of two MVFs $\mathbf{v}$ and $\mathbf{z}$. This gives us a less accurate motion estimation. On the other hand, thanks to $\mathbf{v}$ and $\mathbf{z}$, we can find another macroblock in the frame I_{k-2}: then we can interpolate the data by using five points (not regularly spaced). The procedure consists therefore of the following five steps (see Fig. 2(b)):

1. **Initialization.** - We estimate $\mathbf{u}$ from I_k to I_{k-1} and $\mathbf{w}$ from I_k to I_{k+1} by DISCOVER method.
2. **Motion estimation from I_{k+1} to I_{k+3}.** We perform a block matching motion estimation from I_{k+1} to I_{k+3} and we find the position $\mathbf{p}+\tilde{\mathbf{w}}$.
3. **Motion estimation from I_{k-1} to I_{k-2} and from I_{k-2} to I_{k-3}.** We search for the vector $\mathbf{z}(\mathbf{q})$ that points in I_{k-1} to the position closest to $\mathbf{p}+\mathbf{u}(\mathbf{p})$. Then, we estimate the intersection of the trajectory in the frame I_{k-2} as the

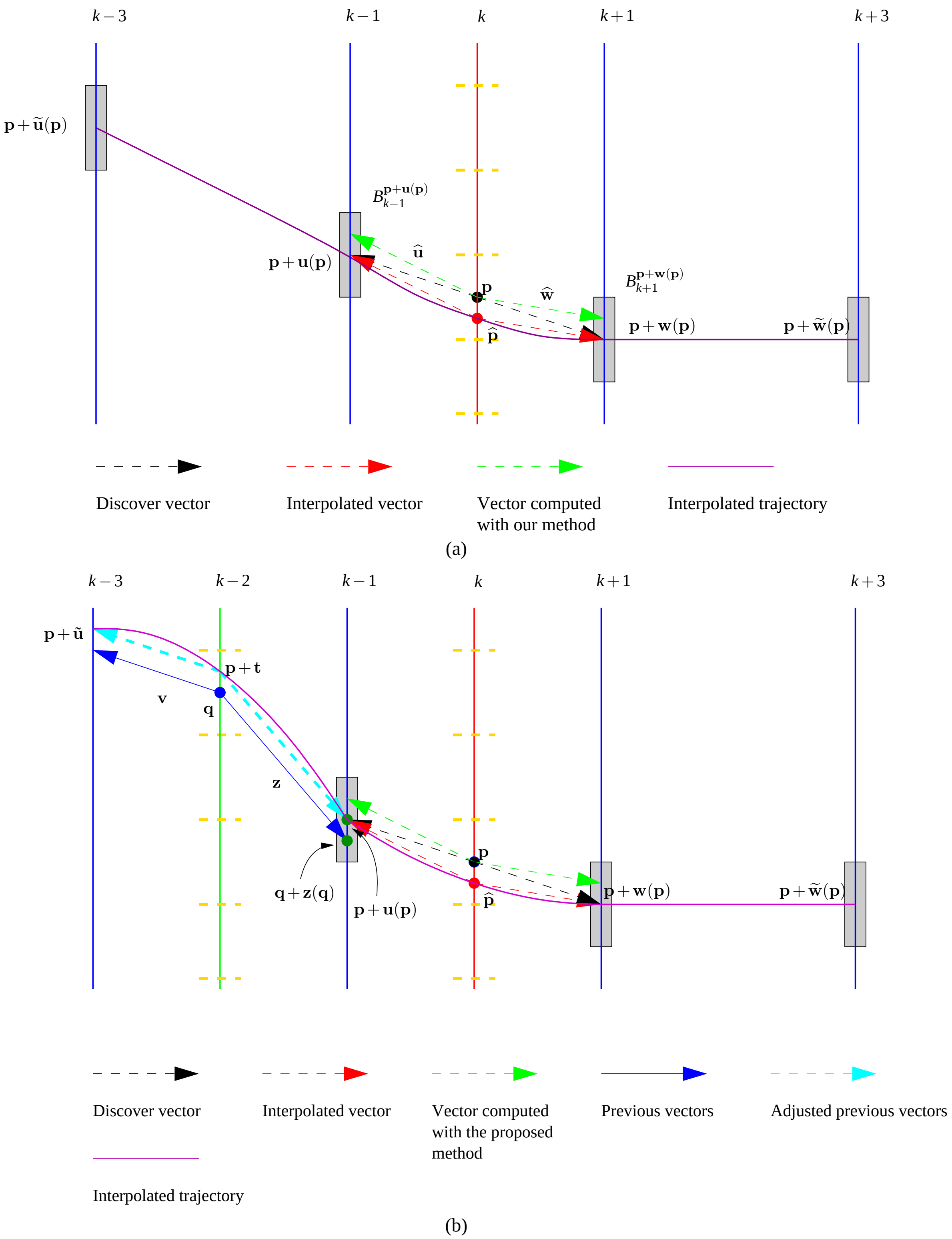

Figure 2: (a) HOMI method for motion estimation. (b) Proposed interpolation method (Fast HOMI) for WZF estimation by exploiting the previous estimated MVF.

point $\mathbf{p}+\mathbf{t}$, with $\mathbf{t}=\mathbf{u}(\mathbf{p})-\mathbf{z}(\mathbf{q})$. For the estimation of the intersection of the trajectory in I_{k-3}, we use the vector $\mathbf{v}(\mathbf{q})$. The intersection point will be $\mathbf{p}+\tilde{\mathbf{u}}$, with $\tilde{\mathbf{u}}(\mathbf{p})=\mathbf{t}+\mathbf{v}(\mathbf{q})$.

4. **Interpolation.** Finally, we interpolate a vector function with the five values $\mathbf{p}+\tilde{\mathbf{u}}$, $\mathbf{p}+\mathbf{t}$, $\mathbf{p}+\mathbf{u}(\mathbf{p})$, $\mathbf{p}+\mathbf{w}(\mathbf{p})$ and $\mathbf{p}+\tilde{\mathbf{w}}(\mathbf{p})$ respectively, at the instants $k-3$, $k-2$, $k-1$, $k+1$ and $k+3$, in order to find its value at the instant k,

GOP size	2	4	8
λ	50	20	0

Table 1: Values of λ for different GOP sizes

which will be denoted by $\widehat{\mathbf{p}}$.

5. **Vector adjustment.** It is done as in the previous section.

We observe that the complexity of Fast HOMI is about the half of the original algorithm.

3.3 Increasing the density of the MVF

When we perform the block matching, we consider a block $B_k^{\mathbf{p}}$, centered in $\mathbf{p}$ and whose size is $N \times N$. Then, we let $\mathbf{p}$ vary in $\{(nM, mM)\}_{(n,m)\in C}$, where $C \subset \mathbb{Z}^2$ is such that we compute a vector for each $M \times M$ block in the image.

Although the condition $M = N$ is usually chosen, in order to increase the SI quality we can use denser MVF by selecting a value for M smaller than N. We observe that, while we can increase the density by reducing M without particular constraint (excepted for the computational complexity), we cannot allow arbitrary variations of N (the block size) since if the blocks are too small, the matching may suffer from noise. On the contrary, thanks to the block overlap, we can densify the MVF without sacrificing robustness. So, we will consider $N = 8$ in the following, but we will use denser MVF with $M < N$.

We can therefore modify both the HOMI and the Fast HOMI technique, ending up with 4 SI generation technique: HOMI with $N = M = 8$ (that is, the one proposed in [4]), referred to as HOMI8 from now on; HOMI with $N = 8$ and $M = 4$, which we will indicate as HOMI4; and the fast version of these techniques, referred to as FastHOMI8 and FastHOMI4. The three new techniques (along with HOMI8 for completeness), will be compared with the reference method, DISCOVER, which does not use the overlap.

4. EXPERIMENTAL RESULTS

In order to use the proposed SI generation techniques, we need to tune the regularization parameter λ for different GOP sizes. At this end we can use the results reported in [4] for HOMI8, since we consider that λ depends mainly on the block size N. For the sake of completeness, we report the optimal values of λ as a function of the GOP size in Tab. 1. We found that the optimal value of λ decreases when the KFs are farther apart. This is reasonable since in this case we must allow larger vector deviations to take into account the movement.

Now we can compare the different methods. We used the test sequences *book arrival*, *ballet*, *jungle* and *breakdancer* at a resolution of 384×512 pixels. We encoded the KFs by the INTRA mode of H.264, using four quantization step values, namely 31, 34, 37 and 40. At this stage, we use as evaluation metric the PSNR of the SI with respect to the original WZF. More precisely for each of the four methods, we compute the PSNR difference (averaged along each sequence) with respect to DISCOVER. This quantity is called Δ_{PSNR}.

The results of this test are reported in Tab. 2 to 5. We observe that in almost all cases, the quality of the side information is improved w.r.t. DISCOVER. The only exception is for GOP size equal to 8, when a good SI estimation is difficult, and all methods are almost equivalent (differences usually under 0.1 dB).

QP	*book arrival*	*ballet*	*jungle*	*breakdancer*
GOP size = 2				
31	0.256	0.263	0.126	0.048
34	0.202	0.214	0.105	0.048
37	0.157	0.149	0.082	0.041
40	0.106	0.112	0.055	0.033
GOP size = 4				
31	0.431	0.255	0.354	0.123
34	0.403	0.220	0.336	0.116
37	0.347	0.175	0.306	0.108
40	0.282	0.135	0.262	0.093
GOP size = 8				
31	0.226	0.042	0.027	0.039
34	0.216	0.039	0.011	0.031
37	0.201	0.028	0.001	0.025
40	0.173	0.021	0.000	0.022

Table 2: Δ_{PSNR} [dB] for HOMI 8

Then, we observe that denser MVFs improve the SI quality for GOP size equal to 2, while they do not help in the case of long-term estimation. We ascribe this behavior to the difficulty of estimating images that are quite far from the references.

Finally we observe that the fast versions of HOMI have fairly good performances, since the quality of the SI is almost unchanged in many cases, while the computational complexity is halved.

The last experiment consisted in computing end-to-end performances (*i.e.* rate reduction and PSNR improvement) of the proposed techniques when inserted into a complete DVC coder like the one in [1], by using the the Bjontegaard metric [2]. The results are shown from Tab. 7 to Tab. 9, and they are not surprising: the proposed method are in general better than the reference DISCOVER, excepted for GOP size equal to 8, where they are practically equivalent.

Moreover, even from the point of view of RD performances, denser MVF are better than sparser ones, and the fast version of HOMI are as effective as the original algorithms. We remark that globally, the better techniques is HOMI4, which allows rate reductions up to 8% w.r.t. the reference.

As a final observation, we note that increasing the quality of the side information does not mean always an increasing of the RD performances. For example, the HOMI8 method has a better side information than DISCOVER for GOP size equal to 8, but worser RD performances. This confirms the intuition that the PSNR with respect to the original WZF is not necessarily an accurate method for evaluating the SI quality, even though for the moment is the most common, since in most cases the RD performances are well correlated to the side information PSNR.

5. CONCLUSIONS AND FUTURE WORK

Based on the previous work in [4], where we propose a higher order motion interpolation, we continue to explore higher order motion interpolation techniques in order to increase the SI quality. We try to increase the density of the estimated MVF and at the same time to reduce the complexity by reusing the already estimated MVF.

The technique using dense MVFs is successful both in increasing the SI quality (up to 0.47 dB better than the reference method DISCOVER), and in improving the end-to-

QP	*book arrival*	*ballet*	*jungle*	*breakdancer*
GOP size = 2				
31	0.464	0.379	0.194	0.052
34	0.384	0.268	0.186	0.035
37	0.334	0.142	0.129	0.072
40	0.236	0.122	0.093	0.048
GOP size = 4				
31	0.472	0.204	0.319	0.134
34	0.467	0.171	0.306	0.122
37	0.422	0.149	0.265	0.106
40	0.322	0.132	0.229	0.074
GOP size = 8				
31	-0.012	-0.049	0.008	-0.016
34	0.027	-0.026	-0.007	-0.002
37	0.012	-0.050	-0.012	0.018
40	0.039	-0.002	-0.044	-0.006

Table 3: Δ_{PSNR} [dB] for HOMI 4

QP	*book arrival*	*ballet*	*jungle*	*breakdancer*
GOP size = 2				
31	0.199	0.239	0.102	0.018
34	0.139	0.199	0.086	0.015
37	0.116	0.150	0.063	0.010
40	0.080	0.114	0.045	0.009
GOP size = 4				
31	0.316	0.340	0.361	0.081
34	0.320	0.315	0.343	0.066
37	0.263	0.255	0.313	0.069
40	0.201	0.202	0.268	0.051
GOP size = 8				
31	-0.040	0.005	0.135	-0.011
34	-0.035	-0.008	0.120	0.004
37	-0.044	-0.020	0.112	-0.016
40	-0.036	-0.015	0.094	-0.011

Table 4: Δ_{PSNR} [dB] for Fast HOMI 8

end RD performances, with a rate reduction attaining 8.2% in the best case. Moreover we show that fast version of the HOMI algorithms have almost the same performance as the original one, but with an halved complexity. These good results encourage us to keep looking for efficient SI methods. A technique we intend to investigate will exploit the previous WZF which has been fully reconstructed (i.e. by using the parity bits to correct it) to produce more accurate motion vector fields.

REFERENCES

[1] 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.

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

[3] 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.*, pages 67–76, Sept. 2007.

[4] G. Petrazzuoli, M. Cagnazzo, and B. Pesquet-Popescu. High order motion interpolation for side information improvement in DVC. In *International Conference on Acoustics, Speech and Signal Processing*, Dallas, TX, 2010.

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

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

QP	*book arrival*	*ballet*	*jungle*	*breakdancer*
GOP size = 2				
31	0.408	0.384	0.174	0.063
34	0.348	0.278	0.172	0.043
37	0.306	0.149	0.121	0.058
40	0.214	0.150	0.091	0.034
GOP size = 4				
31	0.330	0.328	0.330	0.085
34	0.316	0.267	0.315	0.082
37	0.286	0.239	0.289	0.052
40	0.193	0.211	0.242	0.038
GOP size = 8				
31	-0.279	-0.060	0.120	-0.041
34	-0.229	-0.046	0.108	-0.026
37	-0.261	-0.073	0.106	-0.027
40	-0.227	-0.107	0.047	-0.058

Table 5: Δ_{PSNR} [dB] for Fast HOMI 4

	book arrival	*ballet*	*jungle*	*breakdancer*
GOP size = 2				
Δ_R (%)	-1.309	-4.815	-1.649	-2.645
Δ_{PSNR} [dB]	0.035	0.575	0.204	0.114
GOP size = 4				
Δ_R (%)	-4.328	-3.527	-5.856	-3.595
Δ_{PSNR} [dB]	0.279	0.239	0.331	0.169
GOP size = 8				
Δ_R (%)	1.521	-0.392	-1.056	-0.123
Δ_{PSNR} [dB]	-0.086	-0.053	0.060	0.029

Table 6: Rate-distortion performance for HOMI 8

	book arrival	*ballet*	*jungle*	*breakdancer*
GOP size = 2				
Δ_R (%)	-5.655	-6.080	-4.144	-4.259
Δ_{PSNR} [dB]	0.191	0.624	0.497	0.198
GOP size = 4				
Δ_R (%)	-6.211	-4.481	-8.220	-5.240
Δ_{PSNR} [dB]	0.361	0.334	0.430	0.345
GOP size = 8				
Δ_R (%)	1.958	-0.928	-2.372	-3.001
Δ_{PSNR} [dB]	-0.129	0.108	0.139	0.144

Table 7: Rate-distortion performance for HOMI 4

	book arrival	*ballet*	*jungle*	*breakdancer*
GOP size = 2				
Δ_R (%)	-0.957	-4.268	-1.283	-2.100
Δ_{PSNR} [dB]	-0.052	0.436	0.308	0.170
GOP size = 4				
Δ_R (%)	-3.814	-4.890	-6.598	-2.870
Δ_{PSNR} [dB]	0.294	0.398	0.335	0.192
GOP size = 8				
Δ_R (%)	1.922	-0.857	-1.996	-1.622
Δ_{PSNR} [dB]	-0.072	0.023	0.107	0.072

Table 8: Rate-distortion performance for Fast HOMI 8

	book arrival	*ballet*	*jungle*	*breakdancer*
GOP size = 2				
Δ_R (%)	-4.497	-5.438	-3.74	-3.315
Δ_{PSNR} [dB]	0.122	0.703	0.564	0.237
GOP size = 4				
Δ_R (%)	-4.475	-4.591	-7.830	-5.862
Δ_{PSNR} [dB]	0.270	0.423	0.383	0.316
GOP size = 8				
Δ_R (%)	3.425	0.157	-2.454	-2.191
Δ_{PSNR} [dB]	-0.228	-0.099	0.127	0.105

Table 9: Rate-distortion performance for Fast HOMI 4