HIGH ORDER MOTION INTERPOLATION FOR SIDE INFORMATION IMPROVEMENT IN DVC

Giovanni Petrazzuoli, Marco Cagnazzo, Béatrice Pesquet-Popescu

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

ABSTRACT

A key step in distributed video coding is the generation of the side information (SI) *i.e.* the estimation of the Wyner-Ziv frame (WZF). This step is also frequently called image interpolation. State-of-the-art techniques perform a motion estimation between adjacent key frames (KFs) and linear interpolation in order to assess object positions in the WZF, and then the SI is produced by motion compensating the KFs. However the uniform motion model underlying this approach is not always able to produce a satisfying estimation of the motion, which can result in a low SI quality.

In this paper we propose a new method for the generation of SI, based on higher order motion interpolation. We use more than two KFs to estimate the position of the current WZF block, which allows us to correctly estimate more complex motion (such as, for example, uniform accelerated motion). We performed a number of tests for the fine tuning of the parameters of the method. Our experiments show that the new interpolation technique has a small computational cost increase with respect to state of the art, but provides remarkably better performance with up to 0.5 dB of PSNR improvement in SI quality. Moreover the proposed method performs consistently well for several GOP sizes.

Index Terms— Distributed video coding, image interpolation

1. INTRODUCTION

In recent years, distributed video coding (DVC) has raised a considerable amount of interest, since it promises to be the enabling technology for several extremely interesting applications. The most intriguing aspect of DVC is that it should allow a distributed (*i.e.* separated) encoding of correlated signals with the same compression efficiency as centralized (*i.e.* joint) compression. This means that many simple and cheap devices (*e.g.* wireless sensors) can do the same job of a single, complex centralized encoder, provided that joint decoding is possible. In fact, DVC allows to displace the complexity from the encoder to the decoder without loosing performances.

Even though the theoretical bases of DVC are well-known since long time [1, 2], practical implementation has only recently broken through. However, compression performances are still quite far from theoretical bounds. For these reasons, DVC lasts as one of the most attracting research issues in the field of digital video processing.

We consider a very popular frameworks for DVC, proposed by Aaron *et al.* [3] (see Fig. 1). The input sequence is split into key frames (KF) and Wyner-Ziv frames (WZF). KFs and WZFs are seen as different but correlated sources, and are coded independently the one from the other. In particular, the KFs are generally coded with a still image technique, such as JPEG2000, or the INTRA mode of any video coder, and they are used at the decoder to generate an estimation of the WZFs. This operation is performed by the image interpolation box shown in Fig. 1. The result, called side information (SI) is then corrected thanks to the information coming from the encoding of WZFs carried out by means of (possibly) a discrete transform (DCT or DWT), a quantizer, and a channel encoder (LDPC or turbo-coder). The coder sends only the parity bits, and at the receiver side a channel decoder corrects the unavoidable errors of SI as they were channel errors induced by noise. The turbo decoder uses a feedback channel to set the rate of parity bits.

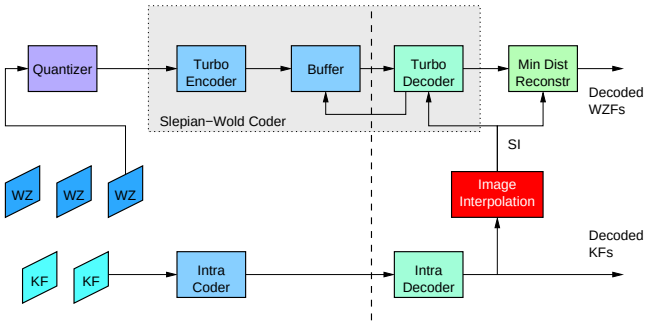

Fig. 1. DVC framework considered in this paper [3]

In this framework, the compression performances are strongly influenced by the quality of the image interpolation step. Several methods have been proposed in literature, as those using block matching (BM) motion estimation (ME) and compensation [4]. A very popular image interpolation scheme is the one proposed within the DISCOVER coder [5], which has become a popular reference. Recently, methods based on differential motion estimation [6, 7] have proved to be very effective in improving the image interpolation, improving the PSNR between SI and original WZFs, and also the end-to-end RD performance of the distributed video coder. These results prove the importance of obtaining a reliable representation of object motion in order to perform a correct image interpolation. With the present paper we continue in this direction, but we explore another tool which can achieve an even better motion description.

Generally, the current SI is obtained by motion compensation of the adjacent KFs. The key point is to find the motion vectors relating these KFs to the current frame. In previous works (in particular in [5, 6, 7]) this movement is estimated using only the same adjacent KFs. In this paper we introduce the idea of using a larger set of KFs in order to estimate this motion. Then, by means of higher order interpolation, we find the vector to be used in order to perform the motion compensation. The tests that we have performed show some remarkable improvement in SI quality, with little increment of computational cost.

The rest of the paper is organized as follows. A brief recall about

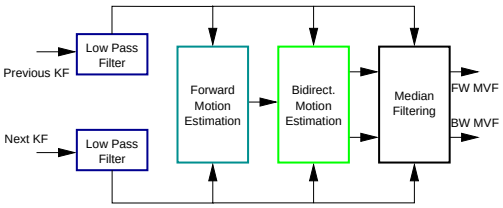

Fig. 2. DISCOVER motion interpolation method.

the DISCOVER interpolation method is given in Section 2. In this section we show that this technique is equivalent to a linear motion interpolation between object positions in the previous and following KFs. In Section 3 we introduce the novel image interpolation method, that mainly amounts to a higher order interpolation for object positions. Experimental results are reported in Section 4, while conclusions and perspectives of this work are drawn in Section 5.

2. DISCOVER AND IMAGE INTERPOLATION

The DISCOVER paradigm [5] is one of the most popular in DVC. Conceptually, its operation mode is depicted by the generic DVC encoder shown in Fig. 1. In particular, the KFs are coded using the Intra mode of H.264 with an assigned quantization step QP. As far as WZFs are concerned, they are first transformed using DCT, then quantized. The resulting coefficients are turbo-encoded, and only parity bits are sent to the decoder, where they are used to correct the side information. Of course, the better is the SI, the fewer parity bits are needed in order to achieve a certain quality for the reconstructed image. Therefore, the image interpolation step in Fig. 1 is of paramount importance. Moreover we remark that we can modify the image interpolator without affecting the encoder.

Before describing the proposed interpolation method, we illustrate the one used by DISCOVER, which is summarized in Fig. 2. Let I_k be the current WZF. The SI is an estimation of I_k produced by using the adjacent KFs, let them be I_{k-1} and I_{k+1}[1]. The KFs are first spatially filtered in order to smooth out noise and higher frequency contributions. Then, a classical block-matching motion algorithm algorithm is used to find a forward motion vector field (MVF) between images I_{k-1} and I_{k+1}. A further bidirectional ME is performed in order to find the movement between the current WZF and the KFs. This process is detailed in Fig. 3: let us consider a block of pixels centred in the position $\mathbf{p}_2$. We call $\mathbf{v}$ the MVF from I_{k+1} to I_{k-1}, $\mathbf{u}$ the one from I_k to I_{k-1}, and $\mathbf{w}$ that from I_k to I_{k+1}. The motion vector produced by the forward motion estimation is therefore $\mathbf{v}(\mathbf{p}_2)$ and it points to the position $\mathbf{p}_2 + \mathbf{v}(\mathbf{p}_2)$ in the previous KF. The basic idea is that motion is linear between I_{k+1} and I_{k-1}, so one might assume that $\mathbf{u}(\mathbf{p}_2 + \frac{1}{2}\mathbf{v}(\mathbf{p}_2)) = \frac{1}{2}\mathbf{v}(\mathbf{p}_2)$. However, in order to avoid gaps and overlaps in the motion compensated image, we need to estimate $\mathbf{u}(\mathbf{p}_2)$. We use for this position the vector passing closest to the block center. In the example in Fig. 3 this is $\mathbf{v}(\mathbf{p}_3)$, since $\|\mathbf{p}_2 - \mathbf{q}_3\| < \|\mathbf{p}_2 - \mathbf{q}_2\|$, where we defined $\mathbf{q}_i = \mathbf{p}_i + \frac{1}{2}\mathbf{v}(\mathbf{p}_i)$. In summary, the DISCOVER algorithm shall choose in this case:

$$\mathbf{u}(\mathbf{p}_2) = \frac{1}{2}\mathbf{v}(\mathbf{p}_3) \qquad \mathbf{w}(\mathbf{p}_2) = -\frac{1}{2}\mathbf{v}(\mathbf{p}_3) \tag{1}$$

Finally, DISCOVER allows a further processing of the MVFs $\mathbf{u}$ and $\mathbf{w}$: first, a refinement around the position found in Eq. (1), is possible. Finally, a weighted median filter is run over the MVFs in order to regularize them. However these latter steps are mainly a refinement of the linear interpolation, which largely affects the resulting performances. The vectors computed in this way are used for compensating the KFs, and the average of the resulting motion compensated KFs constitutes the side information.

[1]We are supposing that KFs are every second frame of the video. However, all the methods described in the following can be trivially extended to the cases of sparser KFs, *i.e.* larger GOP sizes.

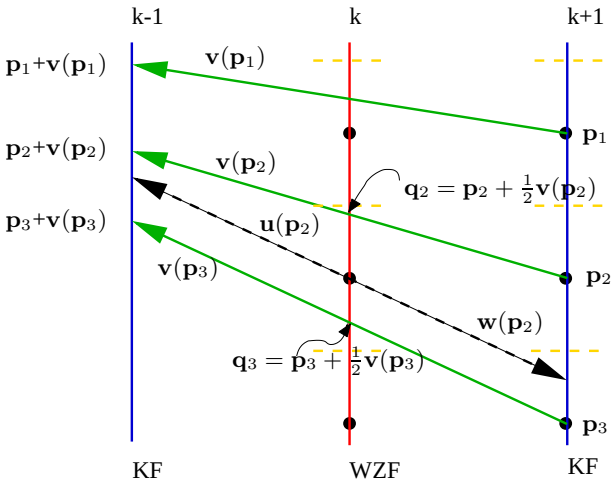

Fig. 3. Bidirectional motion estimation in DISCOVER. Green solid arrows: results of forward ME. Black dashed arrows: results of bidirectional ME for the block centred in $\mathbf{p}_2$.

3. PROPOSED METHOD

The algorithm for image interpolation in DISCOVER is quite effective but can be improved in several ways. In this paper we propose a different technique for the motion estimation. As shown in the previous section, DISCOVER performs a linear interpolation for deducing the object position in the missing WZFs. This is equivalent to assume a uniform motion model (no acceleration of objects) in the whole period between the two KFs. However, whenever the motion in the video deviates from this model we risk to end up with an unsatisfactory motion estimation. So, the idea at the basis of this paper is to perform a higher order interpolation of object positions, in order to be able to take into account more complex motion models (*e.g.* constant acceleration, non-linear trajectories, and so on).

We describe the proposed method along with an example shown in Fig. 4. It consists in three steps: a further block matching to find the positions of the current block in images I_{k-3} and I_{k+3}; the interpolation of the block positions; and finally the adjustment of the vectors in the center of the block.

Let us consider a block of the current WZF (which of course is not available at the decoder), centred on the pixel $\mathbf{p}$. The DISCOVER algorithm provides us with the blocks centred in $\mathbf{p} + \mathbf{u}(\mathbf{p})$ [resp. in $\mathbf{p} + \mathbf{w}(\mathbf{p})$] in the image I_{k-1} [resp. I_{k+1}]. Let $B_{k-1}^{\mathbf{p}+\mathbf{u}(\mathbf{p})}$ [resp. $B_{k+1}^{\mathbf{p}+\mathbf{w}(\mathbf{p})}$] be this block of pixels. Now we want to refine the movement of this block by taking into account the images I_{k+3} and I_{k-3}. Therefore, we start by looking for the position that the block $B_{k-1}^{\mathbf{p}+\mathbf{u}(\mathbf{p})}$ will have in I_{k-3}, by using a regularized block-matching search, *i.e.* the vector $\widetilde{\mathbf{u}}$ such that the following functional is minimized:

$$J(\widetilde{\mathbf{u}}) = \sum_{\mathbf{q}} \left| B_{k-1}^{\mathbf{p}+\mathbf{u}(\mathbf{p})}(\mathbf{q}) - B_{k-3}^{\mathbf{p}+\widetilde{\mathbf{u}}}(\mathbf{q}) \right|^n + \lambda \|\widetilde{\mathbf{u}} - 3\mathbf{u}\| \tag{2}$$

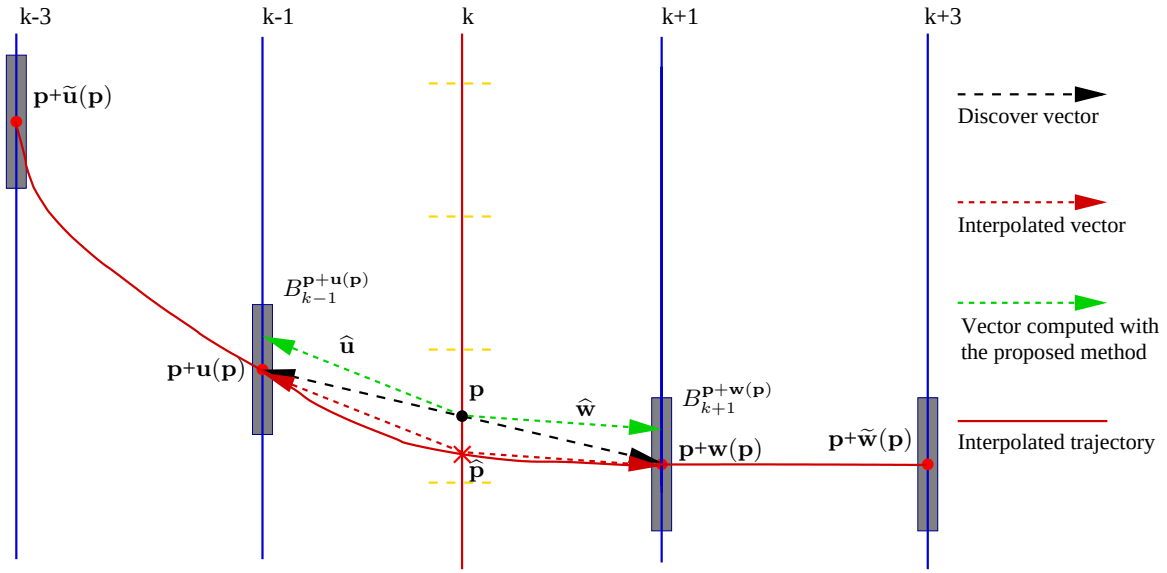

Fig. 4. Proposed interpolation method for motion estimation.

The regularization term penalizes too large deviations from the linear model: with $\lambda \to \infty$ the proposed algorithm becomes equivalent to DISCOVER. The sum of absolute differences (SAD) or the sum of squared differences (SSD) can be used in this criterion simply by setting $n = 1$ or $n = 2$ respectively. Likewise, we can find the vector $\widetilde{\mathbf{w}}$ allowing to match the block $B_{k+1}^{\mathbf{p}+\mathbf{w}(\mathbf{p})}$ with another block in I_{k+3}.

The second step of the proposed algorithm consists in interpolating the positions of the current block in the four images. In other words we interpolate a vector function with the values $\mathbf{p}+\widetilde{\mathbf{u}}$, $\mathbf{p}+\mathbf{u}$, $\mathbf{p}+\mathbf{w}$ and $\mathbf{p}+\widetilde{\mathbf{w}}$ respectively at instants $k-3$, $k-1$, $k+1$, $k+3$, in order to find its value at instant k, be it $\widehat{\mathbf{p}}$. We used a piecewise cubic Hermite interpolation to find the position. As a consequence, the interpolated motion vectors are $\mathbf{u}(\mathbf{p})+\mathbf{p}-\widehat{\mathbf{p}}$ and $\mathbf{w}(\mathbf{p})+\mathbf{p}-\widehat{\mathbf{p}}$. These vectors are shown in dark red in Fig. 4.

The last step consists simply in choosing the interpolated trajectory passing closest to the block center and in assigning the associated vector to the position $\mathbf{p}$, just as in DISCOVER. The difference is that the trajectory is interpolated using four points instead of two. With respect to Fig. 4, the new vectors, shown in green, are:

$$\widehat{\mathbf{u}}(\mathbf{p}) = \mathbf{u}(\mathbf{p}) + \mathbf{p} - \widehat{\mathbf{p}}, \qquad \widehat{\mathbf{w}}(\mathbf{p}) = \mathbf{w}(\mathbf{p}) + \mathbf{p} - \widehat{\mathbf{p}}. \quad (3)$$

We observe that the proposed method can be used as well for distributed multiview video coding, where the KFs can be images taken at the same time by different cameras. In this case, our algorithm can be applied without changes in the hypothesis of equally spaced cameras, and small changes in the general case.

4. EXPERIMENTAL RESULTS

Several experiments have been carried out in order to tune and validate the proposed method. In a first set of tests, we tuned the algorithm parameters for different configurations. In a second one, we compared the proposed method with the reference one in terms of quality of side information. Finally we implemented the proposed method within a modified version of DISCOVER in order to evaluate the impact on the end-to-end coding performance.

	ballet		*book arrival*	
Precision	Lossless	QP=31	Lossless	QP=31
Full pixel	0.318	0.277	0.210	0.211
Half pixel	0.319	0.274	0.199	0.215
Quarter pixel	0.320	0.269	0.169	0.185

Table 1. Δ_{PSNR} [dB] with respect to DISCOVER. SAD criterion, $\lambda = 0$

We performed both interpolation along the temporal axis and along the views (for multiview videos). The KFs are coded with H.264 in INTRA mode for different QP, and also in lossless mode, in order to have a more complete analysis. We evaluate performances by computing the PSNR of the SI with respect to the original WZF, and by comparing it to the PSNR achieved by the original algorithm DISCOVER. The test sequences are *ballet*, *book arrival*, *breakdancer* and *jungle* (multiview, non-rectified videos).

4.1. Parameter tuning

In a first set of experiments, we compared SAD and SSD as metrics in the criterion (2) used for the search of $\widetilde{\mathbf{u}}$ and $\widetilde{\mathbf{w}}$. We do not report results for the sake of brevity, but in all our tests the quality of SI resulting from the two metrics was very close, with slight advantage for SAD, which moreover is computationally lighter. For this reason, in the following experiments we consider only SAD.

In a second set of tests, we considered full, half, and quarter pixel precisions for the search of $\widetilde{\mathbf{u}}$ and $\widetilde{\mathbf{w}}$ with the criterion in Eq. (2). We report some results in Tab. 1. In this case we used SAD criterion and $\lambda = 0$, however similar results have been obtained for other values of λ and for other sequences. We first observe that, even before optimizing λ, the proposed method has a non negligible gain with respect to the state of the art. As far as precision is concerned, we conclude that it is not worth going beyond half pixel, so we keep this value in the following.

The last set of experiments for parameter tuning has been devoted to the λ parameter. We considered 10 values between 0 and 100 and we found that the optimal value depends on the KF distance:

GOP size	2	4	8
λ_{opt}	50	20	0

Table 2. Value of λ_{opt} for different GOP sizes

QP	*book arrival*	*ballet*	*jungle*	*breakdancer*
GOP size = 2				
lossless	0.356	0.460	0.165	0.055
31	0.326	0.348	0.150	0.053
34	0.291	0.313	0.139	0.054
37	0.252	0.238	0.123	0.049
40	0.204	0.204	0.101	0.044
GOP size = 4				
lossless	0.523	0.301	0.387	0.133
31	0.471	0.290	0.369	0.127
34	0.464	0.270	0.360	0.121
37	0.422	0.236	0.341	0.116
40	0.392	0.202	0.314	0.104
GOP size = 8				
lossless	0.226	0.060	0.037	0.036
31	0.234	0.045	0.028	0.041
34	0.230	0.045	0.010	0.032
37	0.230	0.033	0.000	0.028
40	0.198	0.027	0.000	0.025

Table 3. Δ_{PSNR} [dB] for temporal interpolation

λ_{opt} 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. The optimal values are summarized in Table 2. These values have been obtained maximizing the average PSNR over all the sequences and at all the QPs. Even if we do not report all the results, we observe that using the best value can improve SI quality up to 0.15 dB with respect to the trivial case $\lambda = 0$.

4.2. Image interpolation performance

We used the optimal configuration found in the previous section and we compared the SI quality with DISCOVER. Results for temporal interpolation are summarized in Tab. 3. We observe that the proposed method allows remarkable gains in side information quality, up to more than 0.5 dB. We also notice that the highest gains are for a GOP size equal to 4. In fact, when KF are very close (GOP size = 2), linear interpolation is not very bad, so our gains are a little smaller, while when they are too far apart, any interpolation method would have a difficult task.

We also performed image interpolation in the view direction. Results are shown in Tab. 4. We observe that the improvement is reduced with respect to the case of temporal interpolation, since the higher order interpolation better models non-linear trajectories rather than disparity of non-rectified videos. Yet, the gain is not negligible, since we registered improvements up to 0.25 dB.

Finally, we performed a complete end-to-end coding of video sequences with the DVC coder shown in Fig. 1. The rate-distortion performances, computed with the Bjontegaard metric [8], are shown in Tab. 5. We note that we obtain a rate reduction with respect to DISCOVER up to -3.33% and the improvements in received quality are up to 0.15 dB.

5. CONCLUSIONS AND FUTURE WORK

In this work we presented a new method for image interpolation in DVC. It is based on the idea the motion can be better estimated if more than two images are considered and if higher order interpolation of object positions is used to compensate the current KF. This method requires a mild increase in computational complexity, but, on the other hand, it assures a good precision in retrieving the object position. The first experimental results are encouraging: interpolation quality (measured as PSNR of SI with respect to the original WZF) is improved up to 0.52 dB in the most favorable cases. Moreover, using the proposed technique in a complete DVC system reduces the coding rate up to 3.3%. The good results push us in continuing the analysis of higher order interpolation in DVC. In particular we intend to analyze the combination of this technique with differential motion estimation [6, 7].

QP	*book arrival*	*ballet*	*jungle*	*breakdancer*
lossless	0.250	0.054	0.035	0.093
31	0.230	0.050	0.032	0.092
34	0.220	0.046	0.031	0.091
37	0.223	0.036	0.031	0.092
40	0.207	0.021	0.029	0.085

Table 4. Δ_{PSNR} [dB] for disparity estimation (views not aligned)

	book arrival	*ballet*	*jungle*	*breakdancer*
GOP size = 2				
Δ_{R} (%)	-3.218	-0.878	-1.929	-1.904
Δ_{PSNR} [dB]	0.141	0.044	0.076	0.076
GOP size = 4				
Δ_{R} (%)	-2.637	-1.072	-1.989	-2.257
Δ_{PSNR} [dB]	0.116	0.054	0.078	0.093
GOP size = 8				
Δ_{R} (%)	-3.333	-1.073	-1.953	-1.323
Δ_{PSNR} [dB]	0.054	0.150	0.077	0.059

Table 5. Rate-distortion performance comparison between DISCOVER and the proposed method, obtained with the Bjontegaard metric [8] in time domain

6. 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] 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.

[4] B. Girod, A. Aaron, S. Rane, and D. Rebollo-Monedero, "Distributed video coding," *Proc. IEEE*, vol. 93, no. 1, pp. 71–83, Jan. 2005.

[5] 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.

[6] M. Cagnazzo, T. Maugey, and B. Pesquet-Popescu, "A differential motion estimation method for image interpolation in distributed video coding," in *Proceed. of IEEE Intern. Conf. Acoust., Speech and Sign. Proc.*, Taiwan, 2009, vol. 1, pp. 1861–1864.

[7] M. Cagnazzo, W. Miled, T. Maugey, and B. Pesquet-Popescu, "Image interpolation with edge-preserving differential motion refinement," in *Proceed. of IEEE Intern. Conf. Image Proc.*, Cairo, Egypt, 2009.

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