Title: Compensating for motion estimation inaccuracies in distributed video coding

Authors: Jürgen Slowack, Jozef Škorupa, Stefaan Mys, Nikos Deligiannis, Peter Lambert, Adrian Munteanu, and Rik Van de Walle

In: International Conference on Image and Signal Processing (ICISP), pp. 324-332, 2010

Compensating for motion estimation inaccuracies in distributed video coding

Jürgen Slowack[1], Jozef Škorupa[1], Stefaan Mys[1], Nikos Deligiannis[2], Peter Lambert[1], Adrian Munteanu[2], and Rik Van de Walle[1]

[1] Ghent University – IBBT, Dept. of Electronics and Information Systems (ELIS) Multimedia Lab, Gaston Crommenlaan 8 bus 201, B-9000 Ghent, Belgium
[2] Vrije Universiteit Brussel – IBBT, Electronics and Informatics Dept. (ETRO) Pleinlaan 2, B-1050 Brussels, Belgium

Abstract. Distributed video coding is a relatively new video coding approach, where compression is achieved by performing motion estimation at the decoder. Current techniques for decoder-side motion estimation make use of assumptions such as linear motion between the reference frames. It is only after the frame is partially decoded that some of the errors are corrected. In this paper, we propose a new approach with multiple predictors, accounting for inaccuracies in the decoder-side motion estimation process during the decoding. Each of the predictors is assigned a weight, and the correlation between the original frame at the encoder and the set of predictors at the decoder is modeled at the decoder. This correlation information is then used during the decoding process. Results indicate average quality gains up to 0.4 dB.

1 Introduction

Video compression is achieved by exploiting redundancies in the frame sequence. In the temporal direction, these redundancies are often exploited through a process called *motion estimation*. In conventional video compression schemes, motion estimation is performed at the encoder. Each frame is partitioned into non-overlapping blocks, and the goal of the motion estimation process is to find, for each of these blocks, the closest matching block in a set of reference frames. Next, the residual between the block and its prediction is entropy coded, along with the motion vectors.

In distributed video coding (DVC), on the other hand, motion estimation is performed by the decoder instead of the encoder. As a result, the complexity of the encoder is low compared to the complexity of the decoder. However, performing motion estimation at the decoder is difficult, since in DVC motion estimation is performed without having the original frame available. Hence, the original frame available at the encoder is predicted at the decoder using reference frames only. This prediction is called *side information*. Since the side information is but a prediction of the original, additional information is sent by the encoder allowing the decoder to correct the side information. In this process, the correlation between the original frame X and the side information Y is often estimated

at the decoder, for efficient use of the error correcting information (e.g. LDPC or turbo codes).

Complex motion characteristics of video cause a significant amount of errors in the side information. Typically, a motion vector is calculated for each block in the side information by comparing blocks in the reference frames. For example, techniques have been proposed by Aaron et al. [1] and in the context of DISCOVER [2]. In the latter, block-based motion estimation is performed between a past and a future reference frame. This motion field is then interpolated to obtain a motion vector for each block in the side information. Next, the motion vector is further refined. Other researchers have made contributions as well, for example, Kubasov et al. [3] use a mesh-based approach for generating the side information, as well as a combination of mesh and block-based techniques. The problem with these techniques is that motion is assumed linear between the past and future reference frames. This assumption becomes less valid if the distance between the reference frames is large. As a result, sequences with irregular motion such as non-linear motion and occlusion are not predicted very accurately. This is illustrated with an example further on.

The most recent techniques for side information generation use a refinement approach. Decoding is performed partially and the partially decoded frame is used to improve the side information. The improved side information is then used for further decoding. Some interesting techniques in this context are proposed by Martins et al. [4], as well as by Ye et al. [5], and Fan et al. [6], for example. While these techniques show good results, they need to decode some information first before they can compensate for any mistakes made during the side information generation process.

Therefore, in this paper, we propose a technique where some of the motion estimation inaccuracies are taken into account during the decoding, by using a combination of weighted predictors (Sect. 2). The weights are updated using an online procedure. Evaluating our technique indicates average PSNR gains up to 0.4 dB (Sect. 3). Conclusions and future work are provided in Sect. 4, and Sect. 5, respectively.

2 Proposed technique

We first illustrate the problems associated with side information generation in the case of complex motion. Side information has been generated using the techniques employed in DISCOVER [2], for the 5th frame of the Foreman sequence, using the first frame as a past reference, and the 9th frame as a future reference. The side information is corrected using a turbo decoding procedure. When analyzing the residual between the side information and the decoded frame in Fig. 1, it is clear that a lot of errors have been corrected. Judging from the side information itself, it could already be expected that the accuracy of estimating the face is low. However, the residual between the side information and the decoded frame also reveals that errors have been corrected in the background.

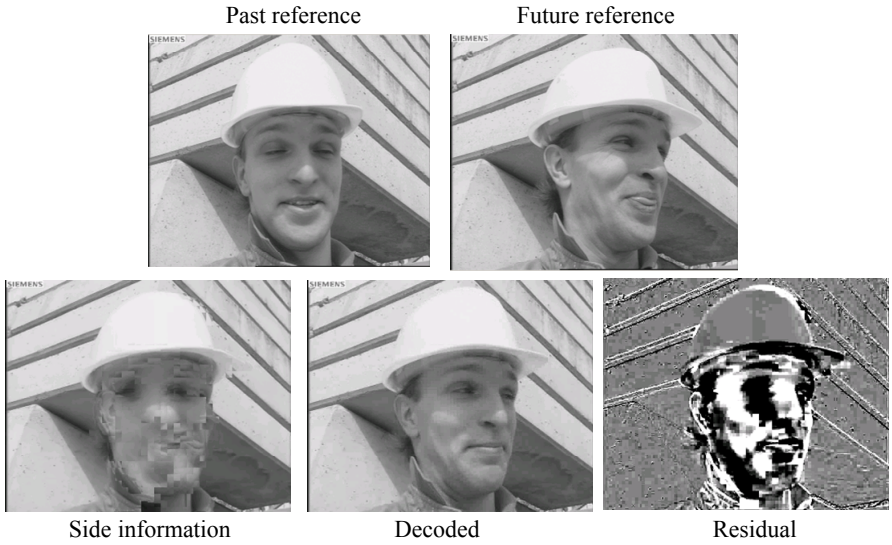

Fig. 1. A lot of errors need to be corrected in the side information if the distance between the reference frames is large, as shown by the residual between the side information and the decoded frame.

More specifically, we can see that edges in the side information are not predicted accurately. This is due to non-linear camera motion.

We can compensate for some of these inaccuracies by using more than one prediction for each block. This is explained using Fig. 2. As input we use the side information called Y generated as in DISCOVER [2]. As such, a particular block in the side information is generated by averaging past and future reference blocks P and F, using a linear motion vector. However, if the motion is non-linear, then the prediction should appear on a different spatial position in the side information. Hence, to predict a block at position (x_0, y_0), we can use the block at position (x_0, y_0) in Y, together with some of the surrounding blocks in Y. This strategy can also be beneficial in other cases with complex motion such as occlusion and deformation.

Before explaining this method in detail, a description of the codec is provided in the following section.

2.1 Codec description

The proposed codec is based on the work of Aaron et al. [1], with some important extensions adopted from DISCOVER [2], and from our previous work [7]. The codec is depicted in Fig. 3, highlighting the extensions proposed in this paper.

The frame sequence is partitioned into key frames I and Wyner-Ziv (WZ) frames W. At the encoder, key frames are intra coded using H.264/AVC intra coding. The intra decoded key frames I' and the original key frames I are used

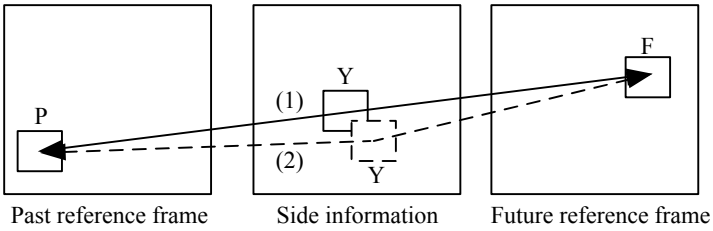

Fig. 2. The linear motion vector (1) could be inaccurate, so that the interpolation between P and F is located on a different spatial position (2).

to calculate the quantization noise, which is needed at the decoder for accurate correlation noise estimation, as in [7]. WZ frames are partitioned into 4-by-4 non-overlapping blocks, and each block is transformed using a DCT. Coefficients at the same index k (e.g. all DC coefficients) are grouped into so-called coefficient bands, and each coefficient band is quantized using a quantizer having 2^{M_k} levels. For each quantized band, bits at the same position (e.g. all most significant bits) are grouped into bitplanes, which are fed to a turbo coder calculating parity bits. These parity bits are stored in a buffer, and sent in portions to the decoder upon request.

At the decoder, key frames are decoded into I'. For each WZ frame, side information is generated using already decoded frames I', and W' (as discussed below). We adopt the techniques for side information generation as used in DISCOVER [2]. The output of this process is the side information frame Y, and for each block the (linear) motion vector MV_{SI}, as well as the residual R_{SI} between the past and future reference blocks. This information is used as input for the extensions provided in this paper. First, for each block, multiple predictors are generated (Sect. 2.2), denoted $\{Y_n\}$. Next, each of these predictors is assigned a weight (Sect. 2.3), and the correlation between the predictors and the original is modeled through the conditional distribution $f_{X|\{Y_n\}}$ (Sect. 2.4). This distribution is used by the turbo decoder, which requests bits until the decoded result is sufficiently reliable[3]. Finally, the quantized coefficients are reconstructed (Sect. 2.5) and inverse transformed to obtain the decoded frame W'.

2.2 Generation of predictors

A block at position (x_0, y_0) is predicted using multiple predictors, obtained from the side information frame Y. The first predictor is the predictor corresponding to linear motion, i.e., the block at the co-located position in Y. To compensate for motion inaccuracies such as non-linear motion, surrounding blocks in Y are taken into account as well. As a compromise between complexity and performance, eight additional predictors are used, namely the ones corresponding to positions $(x_0 \pm 1, y_0 \pm 1)$ in Y. This results in a total of 9 predictors per block.

[3] More specifically, the sign-difference ratio is used as a stopping criterion for turbo decoding [8].

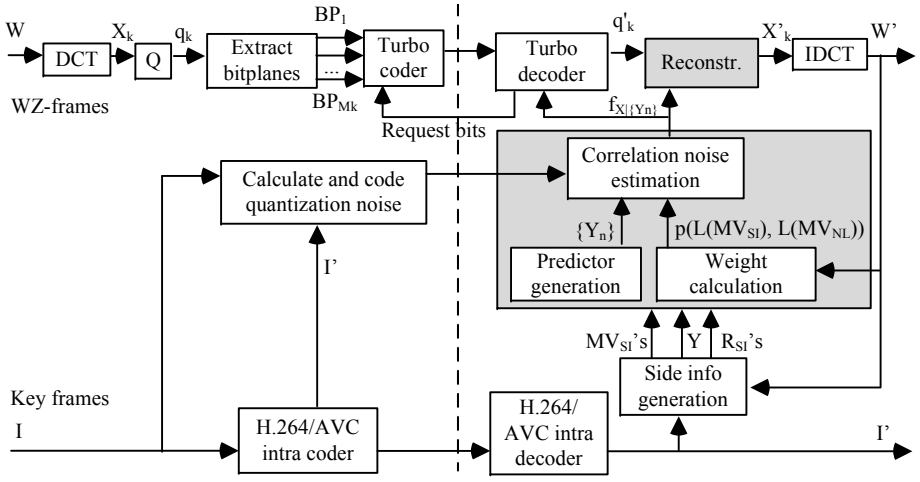

Fig. 3. Our DVC codec with the proposed modifications highlighted.

Not every predictor is equally likely, so that weights are calculated for each predictor, as explained in the following section.

2.3 Online calculation of the predictor weights

Each of the 9 predictors is assigned a weight, according to the probability that this predictor is the best one out of the set. To discriminate between different predictors, denote MV_{NL} as the predictor offset compared to the linear predictor. For example, if the linear predictor corresponds to the block at position (x_0, y_0) in Y, then the predictor at position (x_0+1, y_0-1) in Y has $MV_{NL} = (1, -1)$. As a second parameter, different weights are defined for different values of MV_{SI}. The reason is that large motion vectors MV_{SI} (delivered by the side information generation process) could be less accurate than small motion vectors.

Instead of using two-dimensional parameters MV_{NL} and MV_{SI} as input for retrieving the predictor weights, we define the following magnitude metric:

$$L((x, y)) = \max(|x|, |y|). \tag{1}$$

Using this metric, the predictor weights are defined through the distribution $p(L(MV_{SI}), L(MV_{NL}))$. This distribution is established during the decoding process, since different sequences often require different weights. For some sequences, linear motion might be a good assumption, so that the linear-motion predictor should have a significantly higher weight than the other predictors. On the other hand, for sequences with complex motion characteristics, all predictors could require similar weights. Hence, the following technique is used for updating the weights using an online procedure.

Given the side information frame Y and the decoded frame W', each block in W' is compared to each of the 9 predictors in Y. More specifically, the mean

absolute difference (MAD) is calculated between the block at a certain position (x_0, y_0) in W' and the co-located block in Y. This MAD indicates the amount of errors corrected when using the linear predictor. Likewise, the MAD for other predictors is calculated, for example, comparing the block at position (x_0, y_0) in W' to the block at position $(x_0 + 1, y_0 + 1)$ in Y etc. The predictor with the lowest MAD is then considered the best one out of the set.

However, a non-linear predictor is only considered best in case its MAD is lower than 0.9 times the MAD of the linear predictor. Otherwise, the linear predictor is considered best, nonetheless. This criterion is used to ensure that only significant improvements over the linear predictor are taken into account. For example, in a region with not much texture, one of the non-linear predictors could have a lower MAD than the linear predictor, because the quantization noise in this predictor has distorted the block in such a way that it resembles better the decoded result. To avoid these situations, the MAD of a non-linear predictor must be lower than 0.9 times the MAD of the linear predictor. The value of 0.9 has been experimentally obtained.

Given the best predictor for a block, a histogram table T is updated, where each entry $T(i, j)$ indicates the number of occurrences of the tuple $(L(MV_{SI}) = i, L(MV_{NL}) = j)$. This table only covers the statistics of the current frame. After processing all blocks in W', the histogram table is used to update the global statistics:

$$\begin{aligned} p(L(MV_{SI}) = i, L(MV_{NL}) = j) =& K \cdot p(L(MV_{SI}) = i, L(MV_{NL}) = j) \\ &+ (1 - K) \cdot \frac{T(i, j)}{\sum_k T(i, k)}, \end{aligned} \tag{2}$$

where the update parameter K is set to 0.8. This value – which has been obtained through experiments – remains fixed for all test sequences. A more detailed study of the update parameter is left as future work, as described in Sect. 5.

The global statistics are used for weighing the predictors in the following frame to be decoded.

2.4 The correlation model

Using the weights, a correlation model is defined. This model is used by the turbo decoder. If the model is more accurate, less bits are needed by the turbo decoder for decoding the quantized frame. In addition, the reconstruction process (as described in Sect. 2.5) will be more accurate.

The goal is to model the correlation between the original X and the set of predictors denoted $\{Y_n\}$. This is modeled in the (DCT) transform-domain. For each 4-by-4 block in the original frame, 16 distributions are generated, i.e., one for each coefficient X_k $(k = 0 \ldots 15)$. The predictors are transformed, and all coefficients at the same index are grouped. Denote the predictors for X_k as $\{Y_{k,n}\}$.

It is common in DVC to model the correlation between the original and the side information as a Laplace distribution [9]. Hence, with multiple predictors,

in this paper the conditional distribution $f_{X_k|\{Y_{k,n}\}}$ is modeled as a combination of weighted Laplace distributions, i.e.:

$$f_{X_k|\{Y_{k,n}\}}(x|\{y_{k,n}\}) = \sum_i w_i \cdot \frac{\alpha}{2} e^{-\alpha|x-y_{k,i}|}, \tag{3}$$

where $y_{k,i}$ indicates the kth coefficient of the ith predictor. The weight w_i for each predictor is given by $p(L(MV_{SI}), L(MV_{NL}))$, divided by the number of predictors having the same value for $L(MV_{NL})$. The scaling parameter α is calculated based on the reference residual of the linear predictor, using the techniques proposed in our earlier work [7].

2.5 Coefficient reconstruction

After turbo decoding, the quantization bin q'_k containing the original value (with very high probability) is known at the decoder. The following step is to choose a value in this quantization bin as the decoded coefficient X'_k. This is done through optimal centroid reconstruction [10]:

$$X'_k = \frac{\sum_i w_i \int_{q'_L}^{q'_H} x \cdot \frac{\alpha}{2} e^{-\alpha|x-y_{k,i}|} dx}{\sum_i w_i \int_{q'_L}^{q'_H} \frac{\alpha}{2} e^{-\alpha|x-y_{k,i}|} dx}, \tag{4}$$

where q'_L and q'_H indicate the low and high border of q'_k, respectively.

3 Results

Tests have been conducted on three different sequences: Foreman, Football and Coastguard. All are in CIF resolution, at a frame rate of 30 frames per second. A GOP of size 8 is used, and only the luma component is coded for comparison with the state-of-the-art DISCOVER codec[4]. Our system is also compared to our previous work [7], which applies a better correlation noise model than DISCOVER, but still uses only one predictor per block.

The results indicate improvements over both systems (Fig. 4). The gains are the largest for sequences with complex motion such as Football and Foreman, where the linear predictor does not always provide an accurate prediction. In these cases, using multiple predictors to compensate for inaccuracies shows average Bjøntegaard [11] quality gains up to 0.4 dB over our previous work, and 1.0 dB over DISCOVER (both for Football and Foreman).

For sequences with rather simple motion characteristics, such as Coastguard, less gain is observed. For such sequences, an accurate prediction is already provided by the linear motion predictor, and so using additional predictors provides less gain. Over our previous work, average quality gains of 0.1 dB are reported, while 1.4 dB over DISCOVER.

[4] Executables for the DISCOVER codec are available at http://www.discoverdvc.org/.

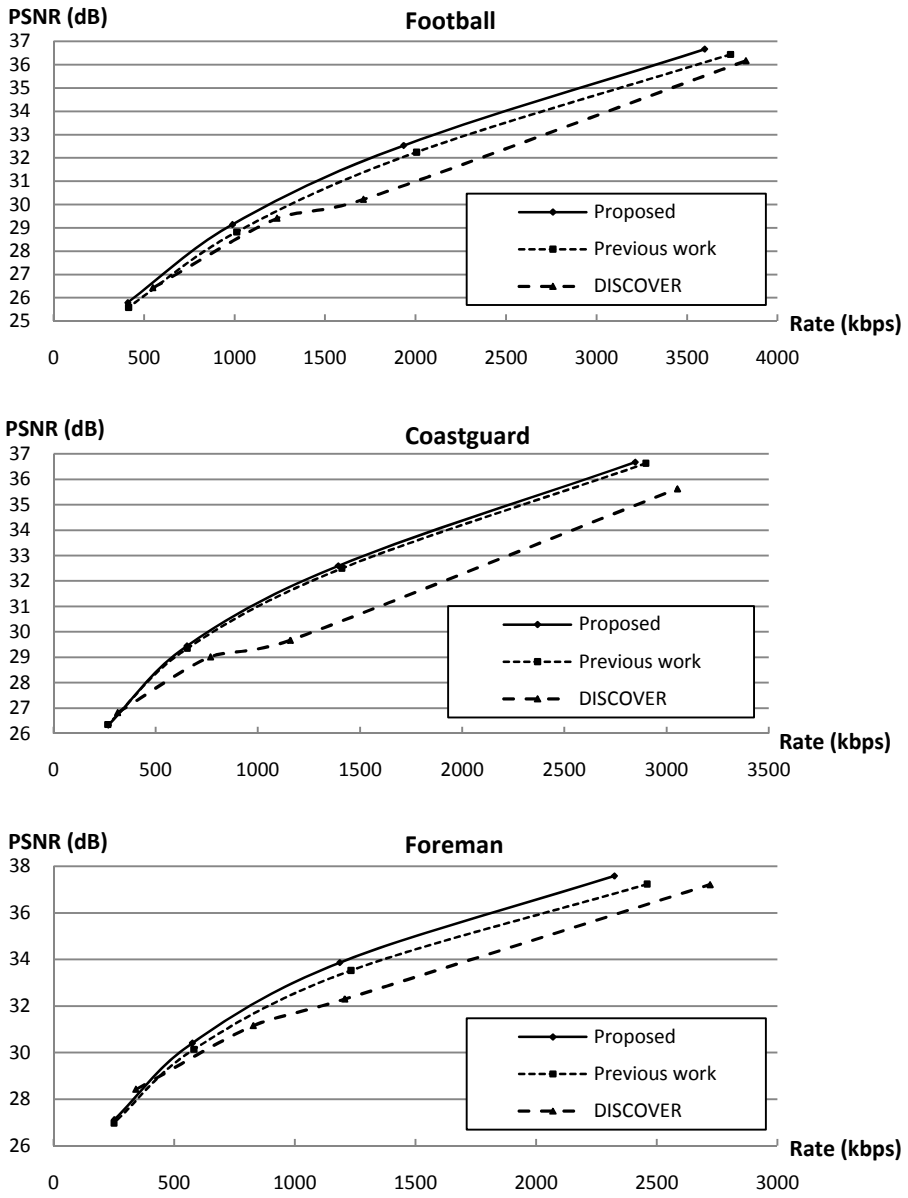

Fig. 4. Over our previous work, average quality gains of 0.4 dB are reported for Football and Foreman, and 0.1 dB for Coastguard.

4 Conclusions

The technique proposed in this paper tries to compensate for inaccuracies in the generation of the side information at the decoder. Instead of using only the predictor associated with linear motion, other predictors are taken into account as well. This technique shows good results, achieving average quality gains up to 0.4 dB over our previous work, and 1.4 dB over DISCOVER [2].These results illustrate the importance of accurate side information and correlation noise estimation in DVC.

5 Future work

Several extensions to this work can be investigated. One possible extension is to compare different techniques for updating the global statistics (used as weights in the following WZ frame to be decoded). For example, instead of using a fixed value for the update parameter K, a value for K can be determined online, i.e., during the decoding process. This could enable better tracking of the motion non-linearities in the frame sequence.

The set of predictors could be extended, for example, adding unidirectional predictors for better handling occlusion. Also, instead of using the same weights for all predictors in one WZ frame, it could be interesting to study techniques where the predictor weights are spatially adaptive.

References

1. Aaron, A., Rane, S., Setton, E., Girod, B.: Transform-domain Wyner-Ziv codec for video. In: Proc. SPIE Visual Communications and Image Processing. Volume 5308. (January 2004) 520–528
2. Artigas, X., Ascenso, J., Dalai, M., Klomp, S., Kubasov, D., Ouaret, M.: The DISCOVER codec: Architecture, techniques and evaluation. In: Proc. Picture Coding Symposium (PCS). (November 2007)
3. Kubasov, D., Guillemot, C.: Mesh-based motion-compensated interpolation for side information extraction in Distributed Video Coding. In: Proc. IEEE International Conference on Image Processing (ICIP). (October 2006)
4. Martins, R., Brites, C., Ascenso, J., Pereira, F.: Refining side information for improved transform domain wyner-ziv video coding. IEEE Transactions on Circuits and Systems for Video Technology **19**(9) (september 2009) 1327–1341
5. Ye, S., Ouaret, M., Dufaux, F., Ebrahimi, T.: Improved side information generation for distributed video coding by exploiting spatial and temporal correlations. EURASIP Journal on Image and Video Processing **2009** (2009) Article ID 683510.
6. Fan, X., Au, O.C., Cheung, N.M., Chen, Y., Zhou, J.: Successive refinement based Wyner-Ziv video compression. Signal Processing: Image Communcation (2009) doi:10.1016/j.image.2009.09.004.
7. Slowack, J., Mys, S., Škorupa, J., Lambert, P., Grecos, C., Van de Walle, R.: Accounting for quantization noise in online correlation noise estimation for distributed video coding. In: Proc. Picture Coding Symposium (PCS). (May 2009)

8. Škorupa, J., Slowack, J., Mys, S., Lambert, P., Grecos, C., Van de Walle, R.: Stopping criterions for turbo coding in a Wyner-Ziv video codec. In: Proc. Picture Coding Symposium (PCS). (May 2009)
9. Brites, C., Pereira, F.: Correlation Noise Modeling for Efficient Pixel and Transform Domain Wyner-Ziv Video Coding. IEEE Transactions on Circuits and Systems for Video Technology **18**(9) (september 2008) 1177-1190
10. Kubasov, D., Nayak, J., Guillemot, C.: Optimal reconstruction in Wyner-Ziv video coding with multiple side information. In: IEEE MultiMedia Signal Processing Workshop. (October 2007) 183–186
11. Bjøntegaard, G.: Calculation of average PSNR differences between RD-curves. Technical report, VCEG (April 2002) Contribution VCEG-M33.