INTER-BLOCK DEPENDENCIES CONSIDERATION FOR INTRA CODING IN H.264/AVC AND HEVC STANDARDS

M. Bichon[†,⋆], *J. Le Tanou*[†], *M. Ropert*[†]

[†] Ericsson TV and Media
France, 35136 Saint-Jacques-de-la-Lande
firstname.lastname@ericsson.com

W. Hamidouche[⋆], *L. Morin*[⋆], *L. Zhang*[⋆]

[⋆] INSA de Rennes (IETR - UMR6164)
France, 35708 Rennes
firstname.lastname@insa-rennes.fr

ABSTRACT

Recent MPEG video compression standards are still block-based: blocks of pixels are sequentially coded using spatial or temporal prediction schemes. For each block, a vector of coding parameters has to be selected. In order to limit the complexity of this decision, independence between blocks is assumed, and coding parameters are locally optimized to maximize the coding efficiency. Few studies have investigated the benefits of inter-block dependencies consideration using Joint Rate-Distortion Optimization (JRDO), especially in Intra coding. To the best of our knowledge, maximum achievable gains of such approaches have never been exhibited. In this paper, we propose two JRDO models performing joint optimization of multiple blocks applied to intra prediction mode decision. The proposed models have been evaluated in both H.264/AVC and HEVC standards. These two models enables a bitrate saving with respect to the classical RDO model up to -3.10% and -2.31% in H.264/AVC and HEVC, respectively.

***Index Terms*—** Keywords : Inter-Block Dependencies, Joint Rate-Distortion Optimization, Intra Coding, H.264/AVC, HEVC.

1. INTRODUCTION

Rate Distortion Optimization (RDO) [1] is a well-known technique used to decide the best set of coding parameters which maximizes coding efficiency. It consists in minimizing the distortion (D) under a bitrate (R) constraint. This minimization under constraint can be rewritten using the Lagrange multiplier method [2]. Knowing λ the Lagrangian Multiplier (LM), it leads to minimization of the R-D cost function:

$$J = D + \lambda \times R \tag{1}$$

In block-based video compression standards like H.264/AVC [3] and High Efficiency Video Coding (HEVC) [4], the RDO technique is applied sequentially to groups of pixels, defined as Coding Units (CU)s or blocks. One denotes CU_i the i_{th} coded CU, $\vec{p_i}$ the vector defining the set of coding parameters to estimate for CU_i and J_i the local R-D cost of CU_i. Then, the global solution is obtained by

$$\min_{\{\vec{p_k}\}_{k=0}^{N-1}} \sum_{i=0}^{N-1} \left(J_{i|_{\{\vec{p_k}\}_{k=0}^{N-1}}} \right) \tag{2}$$

with N the number of CU within the video sequence, and $\{\vec{p_k}\}_{k=0}^{N-1}$ representing all the set of coding parameters from CU_0 to CU_{N-1}. Without loss of generality, (2) can be rewritten as (3) by using the causality of the block-based compression scheme.

$$\min_{\{\vec{p_k}\}_{k=0}^{N-1}} \sum_{i=0}^{N-1} \left(J_{i|_{\{\vec{p_k}\}_{k=0}^{i}}} \right) \tag{3}$$

In this paper, one denotes (3) the *global-RDO*. The complexity of solving (3) exhaustively is tremendous and unrealistic. By writing K the vector space dimension for $\vec{p_i}$, K^N would be the number of configurations to be tested.

Equation (3) is usually simplified based on the common assumption of independence between CUs: it assumes that J_i is only dependent on $\vec{p_i}$. Under this hypothesis, one defines the *classical-RDO* minimization problem as

$$\min_{\{\vec{p_k}\}_{k=0}^{N-1}} \sum_{i=0}^{N-1} \left(J_{i|_{\vec{p_i}}} \right) = \left\{ \min_{\vec{p_i}} \left(J_{i|_{\vec{p_i}}} \right) \right\}_{i=0}^{N-1} \tag{4}$$

Classical-RDO reduces the number of tested configurations to $K \times N$. Although the independence assumption significantly simplifies the computational complexity, we have not found any evaluation of its impact on the coding efficiency in the literature. Some studies [5∼10] investigated joint optimization over multiple CUs, considering spatial or temporal dependencies inherent to the prediction scheme. These approaches are noted Joint Rate-Distortion Optimizations (JRDOs), which differs in this paper from the joint optimization of $\vec{p_i}$ components.

Regarding Inter coding, Yang and al. [5] proposed an inter-frame distortion propagation model using source motion estimation which, along with local delta-quantization, significantly increases the coding efficiency. Li and al. [6] formalized the temporal dependency between two blocks as a rate function, dependent of distortion made on the reference signal used for prediction, noted as $R_{i+1}(D_i)$.

The JRDO in intra coding has also been studied. In [7, 8], Pang and al. used successive convex optimizations in order to reduce block boundaries distortions and thus to improve intra coding efficiency. Sun and al. [9] achieved similar objective by providing an optimized quantization matrix, based on the assumption that coefficients distortion does not follow a uniform distribution in intra coded blocks. In [10], Qingbo and al. experimentally estimated a linear distortion propagation model used for λ computation, coupled with off-line learning and multiple LM framework.

Nevertheless, these studies are using coarse assumptions resulting in simplified dependency models. In addition, they often use a two-in-one algorithm which makes difficult to exhibit gains brought by a JRDO standalone strategy. Finally, to the best of our knowledge, there is no reference proving the maximum achievable gain of a coding decision model that considers intrinsic inter-block dependencies.

In this paper, we propose to evaluate the maximum achievable gain of exhaustive joint optimization of multiple CUs applied to intra prediction mode decision. A brief review of intra coding is first given in Section 2. Section 3 identifies dependencies inherent to intra coding scheme and introduces two JRDO models: *Dual-JRDO* and *Quad-JRDO*. Experimental results and bitrate savings of the proposed JRDO approaches are presented and discussed in Section 4 for both H.264/AVC and HEVC. Finally, Section 5 concludes this paper.

2. INTRA CODING REVIEW

MPEG intra coding scheme can be decomposed into two main steps: mode prediction decision and block partitioning decision. The two decisions steps, for HEVC, are summarized in the following subsections. For a complete explanation on intra coding in HEVC standard, the reader is referred to [11].

2.1. Intra Mode Prediction

Intra prediction consists in sequentially predicting the source signal from the previously reconstructed pixels within the same frame, used as reference. Prediction is built by copying or interpolating reference pixels onto target pixels, according to a rule specified by the predictor. In the case of intra coding, reference pixels used to predict the current CU are depicted in Figure 1.

In HEVC, 35 possible predictors for intra coding are available and given in Figure 1. The DC-mode uses average value of reference pixels and the Planar-mode is a bilinear interpolation designed to preserve continuities along block boundaries. The 33 angular modes represent a direction of projection for reference pixels.

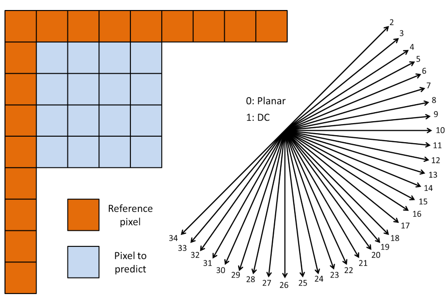


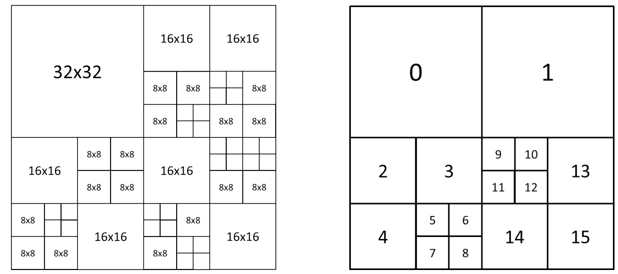
Fig. 1. Intra Prediction in HEVC

The H.264/AVC standard uses the same prediction process, but the number of spatial predictors is limited to 9, or 4 in case of 16x16 size blocks.

2.2. Block Partitioning

In HEVC, each frame is uniformly partitioned in Coding Tree Units (CTU), equivalent to Macroblocks (MBs) in H.264/AVC. CTUs are sequentially compressed in a raster scan order. Then, each CTU can recursively be further sub-divided in multiple CUs, following a *QuadTree* structure. Figure 2 (a) shows an example of the partitioning of a 64x64 CTU in HEVC.

In HEVC, each CU at a given depth of the *QuadTree* can be compressed as a 2Nx2N or NxN partition, with $N \in \{32, 16, 8, 4\}$. The block partitioning decision can be summarized to decide for each CU whether to split or not, based on its R-D cost. The *QuadTree* structure implies a Z-scan order to process successive CUs, an example is shown in Figure 2 (b).



(a) CTU QuadTree Example (b) Coding order in QuadTree

Fig. 2. Intra Partitioning in HEVC

In comparison, H.264/AVC partitioning only allows uniform sub-division of a MB. The only authorized MB partitions are: one 16x16 block, four 8x8 blocks and sixteen 4x4 blocks.

The next Section describes dependencies inherent to the intra coding scheme and the two proposed JRDO models.

3. PROPOSED JRDO MODELS

We define the two Inter-block dependencies inherent to the intra coding scheme. In H.264/AVC and HEVC, both predictive and contextual entropy coding are used.

The compression scheme implies a first dependency between the prediction efficiency on the current CU_i and the distortion made on reference samples D_i^{ref}, i.e. spatial reference neighbors previously coded.

The entropy coding step is usually based on Context Adaptive Binary Arithmetic Coding (CABAC). Each Syntax Element (SE) is binarized and lossless coded, based on context adaptation and occurrence probability updates of each SE bin. Hence, by definition, the syntax coding efficiency of the current CU_i is directly impacted by previous decisions and dependent of SE contexts updates after processing CU_{i-1}. The syntax context state before to code CU_i is noted SC_i.

Once the inter-block dependencies SC_i and D_i^{ref} are defined, we consider $J_{i|_{\vec{p_i}}}(0,0)$, the R-D cost of CU_i knowing $\vec{p_i}$, if no dependencies interfere, either in term of distortion or CABAC. Then, one writes the JRDO equation as follows:

$$J_{i|_{\vec{p_i}}} = J_{i|_{\vec{p_i}}}(0,0) + \Delta J_{i|_{\vec{p_i}}}(SC_i, D_i^{ref}) \quad (5)$$

where $\Delta J_i(SC_i, D_i^{ref})$ is the intra dependency propagation cost that [7, 8, 10] have tried to model. In order to simplify implementations and to keep the computational complexity reasonably low for the study, the joint optimization is limited to intra prediction modes. Hence, the vector of coding parameters $\vec{p_i}$ becomes the scalar p_i.

For a CU to code with a JRDO strategy, the difficulty consists in how to consider neighboring CUs: either in terms of spatial distance, coding order, or both? For example, we can refer to Figure 2 (b), we see that CU_9 has strong spatial dependencies with CU_1. Optimizing both of them jointly should significantly improve the coding efficiency of both CUs. Nevertheless, compression is applied sequentially and the CUs 2 to 8 have to be compressed first, in order to have the correct coding context SC_9 when deciding and coding CU_9.

This observation highlights the difficulty of implementing JRDO approaches in intra coding: on the one hand jointly optimizing many CUs is too complex to compute, and on the other hand ignoring seven CUs leads to strong approximations on coding context. Models proposed in Sections 3.1 and 3.2 avoid these difficulties and apply an exhaustive joint optimization of the current CU with neighbor CUs taking into account both spatial and coding order distances.

3.1. *Dual-JRDO*

Z-scan is the coding order used to sequentially encode the four sub-CUs resulting from split. Based on Z-scan and local dependencies, SC_i and D_i^{ref}, one predicts that CU_{i+1} is highly dependent of CU_i if CU_{i+1} is the spatial right neighbor of CU_i.

To confirm this, we experimentally verified that the right neighbor CU is often the most dependent on the current CU to code. In consequence, we propose the *Dual-JRDO* model that jointly optimizes the intra prediction mode of each CU with the prediction mode of its right neighbor. In order to avoid wrong syntax context states, *Dual-JRDO* handles two cases:

- If CU_i right neighbor is CU_{i+1}, apply the *Dual-JRDO*.
- Else, *classical-RDO* is applied on the current CU_i.

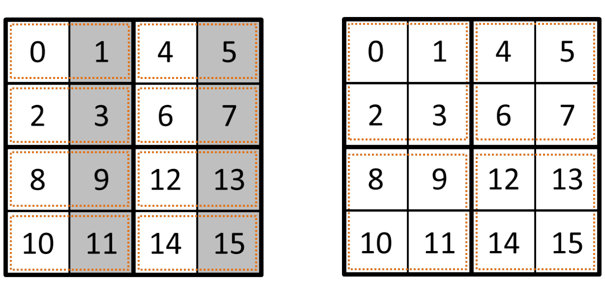
With p_i^* the chosen predictor to encode CU_i and p'_{i+1} the estimated optimal coding mode for CU_{i+1}, *Dual-JRDO* solution is defined by

$$\{p_i^*, p'_{i+1}\} = \underset{\{p_i, p_{i+1}\}}{\arg\min} \left(J_{i|_{p_i}} + J_{i+1|_{p_i, p_{i+1}}} \right) \tag{6}$$

In HEVC, the neighboring CUs can be further split, leading to $p'_{i+1} \neq p^*_{i+1}$. To overcome this problem, one considers that two CUs coming from the same split process have a high probability to have the same final partition size. Statistically, we can notice that the probability of this assumption to be true increases as CU size decreases. An example of *Dual-JRDO* is shown in Figure 3 (a), with dotted lines delimiting the optimization area and dark gray area refers to block coded independently using *classical-RDO*.

In *Dual-JRDO*, K^2 possibilities are explored for half of the CUs, whereas in *classical-RDO*, only K possibilities are explored for all CUs. We deduce that in this particular case *Dual-JRDO* multiplies the complexity of *classical-RDO* by $K/2$.

In H.264/AVC, block partitioning necessarily splits CUs into partitions of the same size, thus we have $p'_{i+1} = p^*_{i+1}$ leading to complexity reduction. Results of *Dual-JRDO* for both H.264 and HEVC standards are presented in Section 4.1.



(a) Dual-JRDO (b) *Quad-JRDO*

Fig. 3. Example of *Dual-JRDO* and *Quad-JRDO*

3.2. *Quad-JRDO*

We can see from Figure 1 that numerous spatial predictors exploit vertical spatial correlations. By definition, *Dual-JRDO* does not consider distortion propagated vertically, e.g. distortion on bottom samples of CU_i will not affect CU_{i+1} if it is the right neighbor. *Quad-JRDO* proposes to include vertical neighbors of CU_i in the joint optimization process.

Quad-JRDO optimizes all sub-CUs coming from the same split operation. At CTU level, raster scan order imposes to code the whole line of CUs before reaching bottom neighbors of CU_i. This results in unachievable computational complexity or wrong syntax context states, reason why *Quad-JRDO* is not applied at CTU nor MB level. Equation (7) depicts the optimization formulation.

$$\{p_k^*\}_{k=i}^{i+3} = \underset{\{p_k\}_{k=i}^{i+3}}{\arg\min} \sum_{k=i}^{i+3} \left(J_{k|_{\{p_l\}_{l=i}^{k}}} \right) \tag{7}$$

Quad-JRDO supposes that all sub-CUs will not be further split, which is not matching the *QuadTree* structure in HEVC. To overcome this issue, (7) is only applied to the special case of NxN mode, other cases use the *classical-RDO*. An example of *Quad-JRDO* is shown in Figure 3 (b). The use of *Quad-JRDO* for NxN analysis multiplies the complexity of *classical-RDO* by $35^3/4$.

In H.264/AVC, MBs are always split in sub-partitions of same size, either 8x8 or 4x4 partitions. In the case of 8x8 partition mode, coding parameters of the four 8x8 blocks are optimized jointly. In the case of 4x4 partition mode, each 8x8 block optimizes jointly the four 4x4 sub-blocks. Experimental configurations and R-D results for both standards are presented and discussed in Section 4.2.

4. EXPERIMENTS

The two methods exposed in Section 3 have been implemented into HEVC and H.264/AVC reference test models, HM16.6 [12] and JM19.0 [13] respectively. The set of sequences utilized is picked among JCT-VC test set [14]. Furthermore, because of the computational complexity of our methods, one encodes one frame of each sequence, in I-only configuration. Indeed, this study aims to estimate maximum gains of JRDO strategies, and not to propose low-complexity solutions.

Tables 1 and 2 present coding efficiency improvements of *Dual-JRDO* and *Quad-JRDO* against *classical-RDO*. Results use Bjøntegaard metric [15] and are expressed in BD-BR, i.e. the percentage of bitrate savings to achieve similar distortion, measured as frame PSNR. Even if initial metric is expressed using 4 different QP values, we use it with 5 QP values ($QP \in (22, 27, 32, 37, 42)$) to cover a larger range of bitrates. Besides, since the proposed solutions are mainly used to optimize luminance (Y) encoding, we focuses on Y BD-BR, nevertheless, similar gains have been obtained in YUV 4:2:0.

4.1. Results on *Dual-JRDO*

Results of *Dual-JRDO* are depicted in table 1. We observe constant gains against *classical-RDO*. Average bitrate savings are of -0.77% and of -0.71% in JM19.0 and HM16.6, respectively. *Dual-JRDO* outperforms *classical-RDO* up to more than -1.3% in both reference softwares. However, one observes that the *BasketballPass* sequence in JM19.0 is the only one to present negligible losses. *Dual-JRDO* slightly favors horizontal predictions. In few cases where vertical prediction is better than horizontal, *Dual-JRDO* can slightly penalize coding efficiency.

Test sequences		JM19.0	HM16.6
1920x1080	Kimono	-1.01%	-0.21%
	ParkScene	-0.68%	-0.48%
	Cactus	-0.80%	-0.62%
	BQTerrace	-0.58%	-0.69%
	BasketballDrive	-0.93%	-0.47%
	Average	**-0.80%**	**-0.49%**
1280x720	FourPeople	-0.77%	-0.68%
	Johnny	-0.94%	-0.41%
	KristenAndSara	-0.96%	-0.47%
	Average	**-0.89%**	**-0.52%**
832x480	RaceHorses	-0.57%	-0.50%
	BQMall	-0.75%	-0.89%
	PartyScene	-0.46%	-0.88%
	BasketballDrill	-1.37%	-1.31%
	Average	**-0.89%**	**-0.90%**
416x240	RaceHorses	-0.67%	-0.98%
	BQSquare	-0.73%	-1.10%
	BlowingBubbles	-0.66%	-0.61%
	BasketballPass	0.08%	-1.02%
	Average	**-0.50%**	**-0.93%**
All	**Average**	**-0.77%**	**-0.71%**
	Maximum	**-1.37%**	**-1.31%**
	Minimum	**0.08%**	**-0.21%**

Table 1. Y BD-Rate of *Dual-JRDO* in JM19.0 and HM16.6

Test sequences		JM19.0	HM16.6 (NxN)
1920x1080	Kimono	-2.53%	-0.04%
	ParkScene	-1.60%	-1.00%
	Cactus	-1.91%	-1.33%
	BQTerrace	-1.37%	-1.34%
	BasketballDrive	-1.79%	-0.24%
	Average	**-1.84%**	**-0.79%**
1280x720	FourPeople	-2.04%	-1.40%
	Johnny	-1.86%	-1.07%
	KristenAndSara	-1.76%	-1.53%
	Average	**-1.89%**	**-1.33%**
832x480	RaceHorses	-1.39%	-1.38%
	BQMall	-1.69%	-1.95%
	PartyScene	-1.38%	-1.94%
	BasketballDrill	-3.10%	-2.31%
	Average	**-1.89%**	**-1.90%**
416x240	RaceHorses	-1.67%	-2.04%
	BQSquare	-1.57%	-2.19%
	BlowingBubbles	-1.69%	-1.60%
	BasketballPass	-1.09%	-2.08%
	Average	**-1.51%**	**-1,98%**
All	**Average**	**-1.78%**	**-1.47%**
	Maximum	**-3.10%**	**-2.31%**
	Minimum	**-1.09%**	**-0.04%**

Table 2. Y BD-Rate of *Quad-JRDO* in JM19.0 and HM16.6

The results presented in [10] are coming from two separate solutions, the first contribution is related to a JRDO approach and the second contribution to a multiple LM framework. In their JRDO approach, similar dependencies as *Dual-JRDO* are considered and gains announced for video of 1920x1080 resolution are about -0.13%. Our study on identical test set shows that achievable gains are on average -0.80% for H.264/AVC. They estimate the distortion dependency with an off-line linear distortion propagation model, and analytically deduct the related optimal λ. Our exhaustive joint prediction optimization demonstrates there is room for improvement in modelization of dependencies or cost propagation.

4.2. Results on *Quad-JRDO*

In the case of the *Quad-JRDO* model presented in Section 3.2, the optimization is applied only to 4x4 and 8x8 blocks in JM19.0, and NxN case in HM16.6. The remaining decisions are based on *classical-RDO*. Results for both implementations are presented in Table 2.

As expected, much higher gains are observed with this second model, which is also much more complex. In average, bitrate savings over *classical-RDO* are -1.78% in JM19.0 and -1.47% in HM16.6. *BasketballPass* sequence, for which negligible losses were observed in *Dual-JRDO*, now outperforms *classical-RDO* from -1.09% in *Quad-JRDO*. Besides, we must note that the *Quad JRDO* is less efficient than *Dual JRDO* on some high resolution sequences such as *Kimono* and *BasketballDrive*. One explanation is that some HD sequences may have more homogeneous areas, where larger partitions are preferred for the prediction; the joint optimization of NxN (i.e. 4x4) partition mode is then of limited interest for these particular cases.

The significant coding efficiency improvement between Dual and *Quad-JRDO* mostly comes from the consideration of vertical predictions and 2-D spatial dependency. Based on these results, it seems relevant to assume that adding more CUs in the proposed joint optimization process, would bring much more gain. One could expect to tend towards *global-RDO* efficiency. In practice, the complexity of such process would lead to computationally intractable simulations.

5. CONCLUSION

In this paper, one demonstrates the benefits of considering inter-block dependencies in intra coding, for both H.264/AVC and HEVC standards. For intra coding, we identify two inter-block dependencies coming respectively from the distortion of reference pixels D_i^{ref} and the CABAC coding context SC_i. Then, we propose an exhaustive JRDO scheme for intra prediction of multiple CUs in order to exhibit maximum achievable gains of dependencies consideration. For a first model, focused on dependencies with the right-CU only, one observes on average -0.77% (H.264/AVC) and -0.71% (HEVC) improvements in BD-rate against *classical-RDO*. In a second model, one includes up to four CUs into the joint optimization process. This last model outperforms on average *classical-RDO* by -1.78% (H.264/AVC) and -1.47% (HEVC). Despite complexity of proposed JRDO methods, we demonstrate that significant gains can be achieved with such a strategy.

Our future work will focus on the modelization of the two identified dependencies in order to propose a low-complexity JRDO scheme for intra prediction of multiple CUs, in the particular context of HEVC. Results of this study will be used for comparison. The *QuadTree* structure and the number of possible partitioning allowed within a CTU will be the main difficulties to overcome in HEVC. Besides, extension to this work to temporal dependencies across frame will be also envisaged.

6. REFERENCES

[1] G. J. Sullivan and T. Wiegand, “Rate-distortion optimization for video compression,” *Signal Processing Magazine, IEEE*, vol. 15, no. 6, pp. 74–90, 1998.

[2] T. Wiegand and B. Girod, “Lagrange multiplier selection in hybrid video coder control,” in *International Conference on Image Processing (ICIP)*, Thessaloniki, Oct. 2001, pp. 542–545, IEEE.

[3] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, “Overview of the H.264/AVC video coding standard,” *IEEE*

Transactions on Circuits and Systems for Video Technology (TCSVT), vol. 13, no. 7, pp. 560–576, July 2003.

[4] G.-J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Transactions on Circuits and Systems for Video Technology (TCSVT)*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[5] T. Yang, C. Zhu, X. Fan, and Q. Peng, "Source Distortion Temporal Propagation Model for Motion Compensated Video Coding Optimization," in *International Conference on Multimedia and Expo (ICME)*. July 2012, pp. 85–90, IEEE.

[6] S. Li, C. Zhu, Y. Gao, Y. Zhou, F. Dufaux, and M.-T. Sun, "Inter-frame dependent rate-distortion optimization using lagrangian multiplier adaption," in *International Conference on Multimedia and Expo (ICME)*. 2015, pp. 1–6, IEEE.

[7] C. Pang, O. C. Au, F. Zou, J. Dai, and R. Cha, "Optimal distortion redistribution in block-based image coding using successive convex optimization," in *International Conference on Multimedia and Expo (ICME)*. 2011, pp. 1–5, IEEE.

[8] C. Pang, O. C. Au, F. Zou, X. Zhang, W. Hu, and P. Wan, "Optimal dependent bit allocation for AVS intra-frame coding via successive convex approximation," in *International Conference on Image Processing (ICIP)*. 2013, IEEE.

[9] L. Sun, O. C. Au, W. Dai, R. Zou, and S. Li, "Modified distortion redistribution problem for High Efficiency Video Coding (HEVC)," in *International Workshop on Multimedia Signal Processing (MMSP)*. 2012, pp. 278–282, IEEE.

[10] W. U. Qingbo, J. Xiong, L. U. O. Bing, C. Huang, and X. U. Linfeng, "A Novel Joint Rate Distortion Optimization Scheme for Intra Prediction Coding in H. 264/AVC," *IEICE Transactions on Information and Systems*, vol. 97, no. 4, pp. 989–992, 2014.

[11] J. Lainema, F. Bossen, W.-J. Han, J. Min, and K. Ugur, "Intra Coding of the HEVC Standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1792–1801, Dec. 2012.

[12] K. McCann, C. Rosewarne, B. Bross, M. Naccari, K. Sharman, and G. Sullivan, "JCTVC-R1002: High Efficiency Video Coding (HEVC) Test Model 16 (HM 16) Encoder Description," July 2014.

[13] A. M. Tourapis, A. Leontaris, K. Shring, and G. Sullivan, "JVT-AE010: H.264/14496-10 AVC Reference Software Manual," July 2009.

[14] F. Bossen, "Common test conditions and software reference configurations," *Tech. Rep. JCTVC-L1100, Joint Collaborative Team on Video Coding (JCT-VC)*, Jan. 2013.

[15] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," in *ITU-T VCEG, Texas, USA, Proposal VCEG-M33*, Austin, Texas, Apr. 2001.