Sergio Sanz-Rodríguez, Tobias Mayer, Mauricio Álvarez-Mesa, Thomas Schierl

A low-complexity parallel-friendly rate control algorithm for ultra-low delay high definition video coding

A LOW-COMPLEXITY PARALLEL-FRIENDLY RATE CONTROL ALGORITHM FOR ULTRA-LOW DELAY HIGH DEFINITION VIDEO CODING

Sergio Sanz-Rodríguez[1], *Tobias Mayer*[1], *Mauricio Álvarez-Mesa*[2,3], *Thomas Schierl*[3]

[1]Image Communication Group, Technische Universität Berlin, Berlin, Germany.
[2]Embedded Systems Architectures, Technische Universität Berlin, Berlin, Germany.
[3]Multimedia Communications, Fraunhofer HHI, Berlin, Germany.

ABSTRACT

Ultra-low delay *high definition* (HD) video coding applications such as video conferencing demand, first, low-complexity video encoders able to support multi-core framework for parallel processing and, second, *rate control algorithms* (RCAs) for successful video content delivering under delay constraints. In this paper a low-complexity parallel-friendly RCA is proposed for HD video conferencing. Specifically, it has been implemented on an optimized H.264/Scalable Video Coding (SVC) encoder, providing excellent performance in terms of buffer control, while achieving acceptable quality of compressed video under the imposed delay constraints.

Index Terms— Video coding, high definition, ultra-low delay, rate control, parallel processing, low-complexity.

1. INTRODUCTION

Recently, *high definition* (HD) video content exchange and management have increasingly been demanded. However, when using HD resolutions, parallel coding architectures are highly recommended to accomplish real-time coding. For example, in H.264/Advanced Video Coding (AVC) *slices* can be used for parallel processing [1] at the expense of noticeable loss in *rate-distortion* (R-D) efficiency [2], whereas in H.265/High Efficiency Video Coding (HEVC) some parallelization approaches are proposed for supporting real-time high-performance coding [3]. It should be noticed that, in the case of H.264/Scalable Video Coding (SVC) [4], real-time picture coding is not pursued, but real-time *access unit* coding (i.e., the union of all the representations of a picture at a given time instant), thus resulting in a challenge for encoder designers to find the best tradeoff between R-D efficiency and coding speed for a given number of dependency layers and type of scalability (quality or spatial).

One of the most popular applications requiring real-time is video conferencing. The aim of this application is to provide ultra-low end-to-end delay video and audio communication typically through a constant bit rate channel. Nevertheless, because of the variable bit rate nature of compressed video, a *rate control algorithm* (RCA) is embedded into the video encoder to avoid encoder buffer (and decoder buffer, which performs the complementary process) overflow and underflow, while providing good quality consistency. Furthermore, given that the ultra-low delay restriction necessarily entails the use of very small buffer sizes, the RCA must also ensure a tight short-term target bit rate adjustment. To achieve this, the *quantization parameter* (QP) can be adjusted for every *macroblock* (MB). For a proper selection of the QP value, the RCA has to incorporate the combination of video complexity and target bit rate as well as the *hypothetical reference decoder* (HRD) constraints [5]. Several RCAs suitable for video conferencing, such as those summarized in [6] and

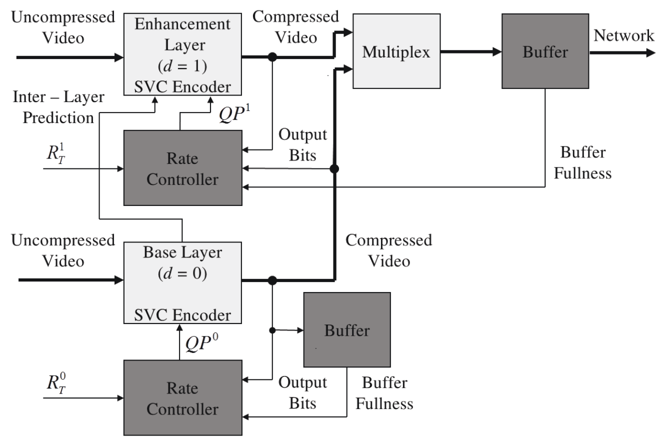

Fig. 1. Block diagram of the RCA for two dependency layers.

[7], have been proposed. However, neither of them is designed for a parallel framework, which is required to achieve real-time coding in HD sequences, unless a picture-level QP decision strategy is conducted at the expense of higher instantaneous bit rate variations [6].

In this paper we propose an efficient MB-level RCA for HD video conferencing. Specifically, it is implemented on an optimized H.264/SVC encoder supporting slice-level parallelism. Therefore, *low-complexity* and *parallel-friendly* are the two main contributions of our proposal. The former is recommended to facilitate real-time coding, whereas the latter is required to provide accurate MB QP selection within a slice and, hence, strict buffer control.

The paper is organized as follows. In Section 2 the proposed RCA is described in detail. In Section 3 the experimental setup is described and the results are presented and discussed. Finally, in Section 4 conclusions are given.

2. PROPOSED RATE CONTROL ALGORITHM

The proposed RCA is depicted in dark gray in Fig. 1 for a SVC encoder consisting of two dependency layers, base layer (BL) and enhancement layer (EL), each one containing a rate controller and associated buffer. It should be noticed that the inter-layer dependencies in SVC involve that the buffer at layer d must receive the sub-streams of layers 0 to d and, therefore, the corresponding target bit rate, R_T^d, must include that of the lower layer, R_T^{d-1}, and so on.

The rate controller at layer d is organized in four levels: *intra period level*, *picture level*, *slice level*, and *MB level*. These levels are detailed in the following subsections making special emphasis on computational simplicity and support for parallelism. Nevertheless, since the main contributions of the paper are focused on slice and MB levels, intra period and picture levels, which have already been studied extensively in the literature, are briefly described, but the reader can be referred to [8] for details.

2.1. Intra Period Level

In video coding applications that require very small buffer sizes, the preferred coding structure is IP...P with only the first picture as I-type. Notice that, since I pictures typically consume much more bit rate than P pictures, the use of periodically inserted I pictures would entail inevitable QP increases to avoid buffer overflow, thus affecting negatively the compressed video quality.

Given a time instant i, this level computes the amount $B_{R,i}^{d}$ of available bit budget to encode the remaining pictures in the intra period. In addition, the initial QP for the I picture, QP_{I}^{d}, is computed, for the BL ($d=0$), by means of a lookup table summarized in the following expression:

$$QP_{I}^{d} = 45 - (5 \cdot \Delta), \text{ if } 0.05 \cdot \Delta \leq Bpp < 0.05 \cdot (1+\Delta), \quad (1)$$

being Δ a positive integer value, and Bpp the average number of target luma and chroma bits per pixel. For the EL ($d>0$), two lookup tables, one for quality scalability (QS) and the other for spatial scalability (SS), are derived from the initial QP selection algorithms proposed in [9].

2.2. Picture Level

In this level the amount T_{i}^{d} of target bits for the ith picture is estimated by means of a weighted combination of two bit allocation methods: one taking just a portion of $B_{R,i}^{d}$ for target bit rate adjustment, and the other watching over the current buffer status for overflow and underflow prevention. Finally, T_{i}^{d} is upper and lower bounded to satisfy the HRD constraints.

2.3. Slice Level

Unlike the previously proposed RCAs, an additional level is included to guarantee slice-level parallelism, that is, several threads, one per slice, encoding sections of a picture in parallel. Within this framework, the RCA should be able to assign in advance (i.e., before encoding the picture) a suitable amount of target bits per each slice. For this purpose, two different bit allocation approaches are employed: one for the first I and P pictures, and the other for the remaining P pictures. The reason behind this separation is due to the great impact on the buffer level when the first pictures are encoded without knowing in advance the spatio-temporal complexity of the sequence.

2.3.1. For the First I and P Pictures

Given that a very short buffer size is assumed in an ultra-low delay application, the paramount goal of the slice level for these pictures is to prevent buffer overflow and underflow, regardless of whether it may influence negatively on the reconstructed picture quality.

For the I picture, the following four bit count thresholds for the buffer occupancy are used: overflow threshold (T_{OV}^{d}), upper threshold (T_{UP}^{d}), lower threshold (T_{LW}^{d}), and underflow threshold (T_{UN}^{d}). Specifically, T_{OV}^{d}, T_{UP}^{d}, T_{LW}^{d} and T_{UN}^{d} are defined as the number of bits required by the picture to reach a buffer level equal to 100%, 70%, 20% and 0% of the buffer size, respectively. The basic idea behind this threshold-based approach is to properly regulate in the next level the MB QP, so that, once the picture has been encoded, the amount of total bits is not greater than T_{UP}^{d} and not lower than T_{LW}^{d}. Otherwise, the MB QP may be changed more aggressively in order not to produce more bits than T_{OV}^{d} or less bits than T_{UN}^{d}.

Subsequently, each threshold value is divided into several equal parts, as many as the number N_{SL}^{d} of slices per picture in the dependency layer d. In particular, for the jth slice in the picture, the following set of thresholds is defined:

$$\left(T_{OV,j}^{d}, T_{UP,j}^{d}, T_{LW,j}^{d}, T_{UN,j}^{d}\right) = \frac{\left(T_{OV}^{d}, T_{UP}^{d}, T_{LW}^{d}, T_{UN}^{d}\right)}{N_{SL}^{d}}. \quad (2)$$

Notice that, although a more fair bit distribution could be carried out by, for example, using some spatial activity measurement for slice coding complexity prediction, a low-complexity bit allocation approach is pursued in this paper as remarked before.

For the first P picture, the bit range between T_{UP}^{d} and T_{LW}^{d} is reduced around the amount of bits needed to reach a target buffer level (stated in [8]) in order to achieve a stricter buffer control. It is important to notice that the QP range to be used for the I picture may not be suitable for the current one, since only buffer-based decisions are made without considering the temporal activity of the scene [9].

Next, each threshold value is split into N_{SL}^{d} portions, but using the coding complexity $C_{I,j}^{d}$ corresponding to each jth slice in the already encoded I picture, that is,

$$\left(T_{OV,j}^{d}, T_{UP,j}^{d}, T_{LW,j}^{d}, T_{UN,j}^{d}\right) = \frac{C_{I,j}^{d} \cdot \left(T_{OW}^{d}, T_{UP}^{d}, T_{LW}^{d}, T_{UN}^{d}\right)}{\sum_{u=0}^{N_{SL}^{d}-1} C_{I,u}^{d}}. \quad (3)$$

Specifically, in this paper the slice coding complexity is simply measured in terms of sum of product *TotalBits*$\times Q$ of all MBs in the slice, being Q the quantization step associated with a certain QP.

2.3.2. For the Remaining P Pictures

In this case, the amount $T_{i,j}^{d}$ of target bits for the jth slice in the ith picture is computed as

$$T_{i,j}^{d} = \frac{\widetilde{C}_{i,j}^{d}}{\sum_{u=0}^{N_{SL}^{d}-1} \widetilde{C}_{i,u}^{d}} \cdot T_{i}^{d}, \quad (4)$$

where $\widetilde{C}_{i,j}^{d}$ stands for a prediction of the slice coding complexity. More specifically, $\widetilde{C}_{i,j}^{d}$ is updated frame by frame via exponential average of the coding complexities corresponding to co-located slices in previous pictures, with a *forgetting factor* (FF) set to 0.25.

2.4. Macroblock Level

This level focuses on estimating a suitable MB QP in order to comply with the bit budget constraints above specified. As in slice level, two different strategies are also employed and described bellow.

2.4.1. For the First I and P Pictures

Three steps are followed before encoding a kth MB in the jth slice corresponding to the ith picture: 1) predict the amount $\widetilde{B}_{i,j}$ of total bits required by the slice once the $(k-1)$th MB has been encoded; 2) compare it to those thresholds specified in Eqs. (2) and (3); and 3) modify the MB QP, $QP_{i,j,k}^{d}$, accordingly. In Algorithm 1 the proposed MB-level QP estimation approach is summarized.

In this algorithm $N_{R,MB}$ denotes the number of remaining MBs in the slice, and $B_{i,j,u}^{d}$ the amount of total bits consumed by the uth MB in the slice. Notice that the prediction $\widetilde{B}_{i,j}$ is also compared to previous one, $\widetilde{B}_{i,j,prev}$, so that $QP_{i,j,k}^{d}$ can only be modified when necessary, that is, when $\widetilde{B}_{i,j}$ is still too high for the current QP, thus providing smooth QP variation within the slice.

Algorithm 1 QP estimation procedure for the first I and P pictures.

1. **if** $k = 0$ **then** {first MB?}
2. $\quad QP^d_{i,j,k} \leftarrow QP^d_I$
3. **else**
4. $\quad \widetilde{B}_{i,j} \leftarrow \left(1 + \frac{N_{R,MB}}{k}\right) \cdot \sum_{u=0}^{k-1} B^d_{i,j,u}$ {prediction}
5. $\quad$ **if** $\left(\widetilde{B}_{i,j} \geq T^d_{UP,j}\right) \wedge \left(\widetilde{B}_{i,j} \geq \widetilde{B}_{i,j,prev}\right)$ **then**
6. $\quad\quad QP^d_{i,j,k} \leftarrow QP^d_{i,j,k-1} + 1 +$ (P picture? $1 : 0$)
7. $\quad$ **else if** $\left(\widetilde{B}_{i,j} \leq T^d_{LW,j}\right) \wedge \left(\widetilde{B}_{i,j} \leq \widetilde{B}_{i,j,prev}\right)$ **then**
8. $\quad\quad QP^d_{i,j,k} \leftarrow QP^d_{i,j,k-1} - 1$
9. $\quad$ **else if** $\widetilde{B}_{i,j} \leq T^d_{OV,j}$ **then**
10. $\quad\quad QP^d_{i,j,k} \leftarrow QP^d_{i,j,k-1} + 2 +$ (P picture? $1 : 0$)
11. $\quad$ **else if** $\widetilde{B}_{i,j} \leq T^d_{UN,j}$ **then**
12. $\quad\quad QP^d_{i,j,k} \leftarrow QP^d_{i,j,k-1} - 1$
13. $\quad$ **else**
14. $\quad\quad QP^d_{i,j,k} \leftarrow QP^d_{i,j,k-1}$
15. $\quad$ **end if**
16. $\quad \widetilde{B}_{i,j,prev} \leftarrow \widetilde{B}_{i,j}$
17. **end if**

2.4.2. *For the Remaining P Pictures*

The amount $T^d_{i,j,k}$ of target bits to encode the current kth MB in the jth slice corresponding to the ith picture is computed as

$$T^d_{i,j,k} = \frac{\widetilde{C}^d_{i,j,k}}{\sum_{u=k}^{N^d_{MB}-1} \widetilde{C}^d_{i,j,u}} \cdot T^d_{R,i,j}, \qquad (5)$$

where $\widetilde{C}^d_{i,j,k}$ is a prediction of the MB coding complexity via exponential average ($FF = 0.25$) of those corresponding to co-located MBs in previous pictures, N^d_{MB} is the number of MBs in the current slice, and $T^d_{R,i,j}$ is the amount of available target bits to encode the remaining MBs in the slice.

Afterwards, $QP^d_{i,j,k}$ is computed by means of the following simple linear R-Q model:

$$T^d_{i,j,k} = \frac{\widetilde{X}^d_{i,j,k}}{Q^d_{i,j,k}} + \widetilde{H}^d_{i,j,k}, \qquad (6)$$

where $\widetilde{X}^d_{i,j,k}$ and $\widetilde{H}^d_{i,j,k}$ are, respectively, a prediction of the complexity to encode the MB transform coefficients (in terms of product *CoeffBits*$\times Q$) and a prediction of the amount of header bits. Both predictors are also updated via exponential average ($FF = 0.25$) of those corresponding to co-located MBs in previous pictures.

Finally, to ensure quality consistency within the slice and also between slices, $QP^d_{i,j,k}$ is bounded ± 1 unit respect to that of the preceding MB and ± 4 units respect to the average QP of the previous picture. However, for the first MB in the slice, the QP is set to the average QP of the co-located slice.

3. EXPERIMENTS AND RESULTS

3.1. Experimental Setup

The proposed RCA was implemented on an H.264/SVC encoder supporting slice-level parallelism and optimized with SIMD (SSE2) instructions for the most time consuming kernels. The main characteristics and coding tools of the video encoder are listed next: 2 dependency layers, 1 temporal layer, IP...P pattern, full intra prediction for I picture, only 8×8 inter prediction for P picture, diamond shaped motion estimation with search range of 16×16 pixels, adaptive residual prediction, no adaptive inter layer prediction, no adaptive motion vector prediction, R-D optimization, and CAVLC coding.

In order to assess the performance of the proposed RCA, it was compared to two methods: 1) constant QP (CQP) coding, as a reference from the R-D efficiency point of view; and 2) the same RCA without taking into account parallelism, that is, both slice and MB levels were disabled to make QP decisions at picture level using the same R-Q model as that in Eq. (6) and, then, bounding the result ± 1 unit respect to the preceding frame QP. We will refer to this picture-level scheme as reference RCA.

A total of six 10-s 720p@60fps test sequences suitable for video conferencing were selected, specifically [10]: *FourPeople*, *Johnny*, *KristenAndSara*, *Vidyo1*, *Vidyo3*, and *Vidyo4*. However, for our experiments, these sequences were then converted to 720p@30fps and 1440p@30fps, this latter to allow for SS. The SVC normative upsampling method based on a set of 4-taps filters was used.

The first task carried out was the search of the best tradeoff between R-D efficiency and number of slices required to reach real-time coding. For this purpose, a PC with 8 Intel Xeon CPUs E5-2687W@3.10GHz, OS Debian with Linux kernel 3.2.0-4 and compiler GCC 4.7.2 with optimization level O3, was used to encode the test sequences with CQP and different number of slices. In particular, the following QPs were selected [10]: 22, 27, 32 and 37 for the BL and, for the EL, 5 QP units less in case of QS (to guarantee higher reconstruction fidelity) and the same QPs as those for the BL in case of SS. The coding time results showed that real-time coding over an acceptable range of bit rates could be achieved with $\mathbf{3|3}$ and $\mathbf{3|6}$ (BL|EL) slices for QS and SS, respectively, at the expense of maximum *Bjøntegard difference* (BD)-rate[1] losses of $\mathbf{7.0|3.4}\%$ compared to 1|1-slice coding.

The output bit rates generated by CQP coding with the final slice settings were used as target bit rates for both RCAs. The results for a buffer size per layer set to **50ms** are shown and discussed bellow.

3.2. Results and Discussion

BD-rate was the measurement used to compare both RCAs from the R-D efficiency point of view. Table 1 reports the BD-rate performance of both RCAs referred to multi-slice CQP coding for the two types of scalability (notice that the results for BS-QS and BS-SS do match, since the same number of slices was employed). As can be observed, the reference RCA achieves better BD-rate performance, as expected, since picture-level RCAs generally provides better quality than MB-level RCAs for a given target bit rate [6].

Table 2 illustrates the coding performance in terms of target bit rate adjustment, by measuring the bit rate error respect to CQP coding, and buffer control, by measuring the average/maximum percentages in which overflows (#O) or underflows (#U) occurred in one coding. As can be shown, both RCAs achieve bit rate errors very close to 0%, as also expected, since a short-term target bit rate adjustment is pursued in the assessed RCAs. Nevertheless, the results in terms of #O and #U revealed that the proposed RCA was capable of considerably reducing the overflow and underflow occurrences, since MB-level RCAs generally guarantee tighter buffer control when compared to picture-level RCAs (see also Fig. 2 depicting some representative buffer occupancy time evolutions provided by the assessed RCAs for the sequence *FourPeople*). It is also worth mentioning that unavoidable overflows happened at the beginning of

[1]BD-rate compares two (reference and tested) R-D curves with only a single number. If the number is positive, the reference algorithm outperforms the tested one (see [11] for details).

Table 1. BD-rate performance [%] of both RCAs referred to CQP coding with 3|3 slices (QS) and 3|6 slices (SS).

Sequence	BL-QS(SS)		EL-QS		EL-SS	
	Ref.	**Prop.**	**Ref.**	**Prop.**	**Ref.**	**Prop.**
FourPeople	14.1	16.6	11.1	13.9	21.4	23.5
Johnny	28.1	44.3	14.2	19.7	20.9	28.1
KristenAndSara	12.6	22.1	8.5	13.3	14.8	20.1
Vidyo1	3.5	15.7	3.7	7.4	6.6	10.9
Vidyo3	5.7	9.7	5.6	11.0	7.5	12.1
Vidyo4	6.8	12.5	4.2	7.5	6.9	10.4
Overall	11.8	20.2	7.9	12.1	13.0	17.5

Table 2. Rate error, #O and #U performance [%] of both RCAs. Average/maximum results are presented.

Layer	RCA	Rate Error	#O	#U
BL-QS(SS)	**Ref.**	0.16/0.66	11.64/50.33	8.40/15.00
	Prop.	0.04/0.11	0.22/1.33	0.08/1.67
EL-QS	**Ref.**	0.05/0.13	0.00/0.00	10.68/20.00
	Prop.	0.00/0.01	0.00/0.00	0.18/2.00
EL-SS	**Ref.**	0.08/0.37	1.29/8.00	7.61/13.00
	Prop.	0.00/0.01	0.00/0.00	0.14/1.00

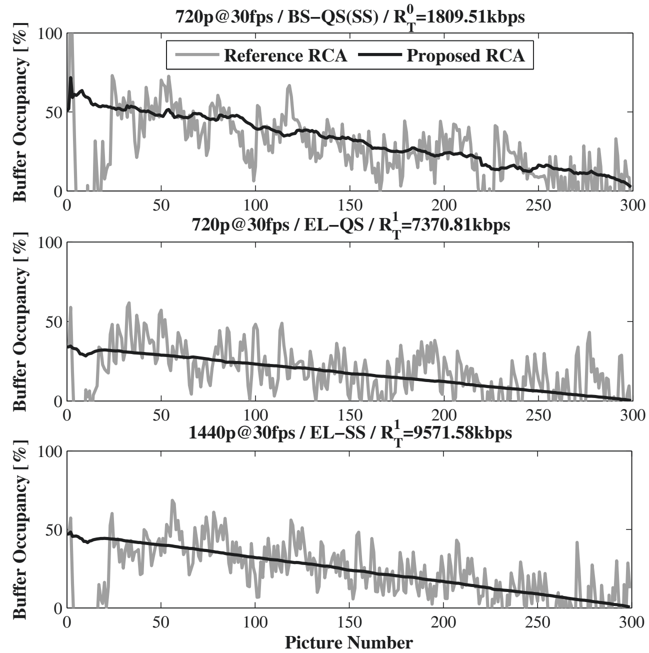

Fig. 2. Comparison of buffer occupancy time evolutions corresponding to both RCAs for the sequence *FourPeople*.

the coding process of some video sequences encoded with low target bit rates, since, for the stated video encoder, 50ms as buffer size was too short for the high bit rate consumed by the I picture even encoded with the maximum QP, thus explaining the non-zero #O values obtained by the proposed RCA at the BL (see Table 2).

Finally, the coding time results disclosed that, with the specified coding system, real-time coding was attained, for QS, in all the tested overall target bit rates and, for SS, until 10Mbps.

4. CONCLUSIONS

In this paper a low-complexity parallel-friendly MB-level RCA has been proposed for HD video conferencing. In particular, the RCA has been implemented on an optimized H.264/SVC encoder supporting slice-level parallelism to allow for real-time coding.

For a slice setting consisting of 3|3 and 3|6 slices for QS and SS, respectively, the experimental coding results demonstrated its remarkably good performance in terms of target bit rate adjustment and buffer control, with acceptable losses in R-D efficiency when compared to a picture-level RCA. Furthermore, with the specified coding system, real-time coding was achieved for a reasonable range of target bit rates.

5. REFERENCES

[1] T. Wiegand, G.J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 7, pp. 560–576, 2003.

[2] C.C. Chi, M. Álvarez-Mesa, B. Juurlink, G. Clare, F. Henry, S. Pateux, and T. Schierl, "Parallel scalability and efficiency of HEVC parallelization approaches," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1827 –1838, dec. 2012.

[3] G.J. Sullivan, J. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[4] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 9, pp. 1103–1120, Sept. 2007.

[5] J. Ribas-Corbera, P.A. Chou, and S.L. Regunathan, "A generalized hypothetical reference decoder for H.264/AVC," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 7, pp. 674–687, 2003.

[6] S. Sanz-Rodríguez, Ó. del Ama-Esteban, M. de Frutos-López, and F. Díaz de María, "Cauchy-density-based basic unit layer rate controller for H.264/AVC," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 20, no. 8, pp. 1139–1143, Aug. 2010.

[7] S. Sanz-Rodríguez and F. Díaz-de María, "In-layer multibuffer framework for rate-controlled scalable video coding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 8, pp. 1199 –1212, Aug. 2012.

[8] S. Ma, Z. Li, and F. We, "Proposed draft of adaptive rate control," *JVT-H017, 8th JVT Meeting*, Geneva, Switzerland, May 2003.

[9] S. Sanz-Rodríguez and F. Díaz-de María, "Rate control initialization algorithm for scalable video coding," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*, Sept. 2011, pp. 3497 –3500.

[10] F. Bossen, "Common test conditions and software reference configurations," *JCTVC-J1100, 10th JCTVC Meeting*, Stockholm, Sweden, July 2012.

[11] G. Bjøntegaard, "Improvements of the BD-PSNR model," *ITU-T SG16, VCEG-AI11*, Berlin, Germany, July 2008.