New rate adaptation method for JPEG2000-based SNR Scalable Video Coding with Integer Linear Programming models

Livio Lima #1, Renata Mansini *, Riccardo Leonardi #2

Signal Processing Group
Department of Electronics for Automation, University of Brescia
Via Branze, Brescia, Italy
[1] livio.lima@ing.unibs.it
[2] riccardo.leonardi@ing.unibs.it

* *Operations Research Group*
Department of Electronics for Automation, University of Brescia
Via Branze, Brescia, Italy
renata.mansini@ing.unibs.it

***Abstract*—In the last few years scalable video coding emerged as a promising technology for efficient distribution of videos through heterogeneous networks. In a heterogeneous environment, the video content needs to be *adapted* in order to meet different end terminal capability requirements (user adaptation) or fluctuations of the available bandwidth (network adaptation). Consequently, the adaptation problem is a critical issue in scalable video coding design. In this paper we introduce a new adaptation method for a proposed JPEG2000-based SNR scalable codec, that formulates and solves the adaptation problem as an Integer Linear Programming problem.**

***Index Terms*—Scalable video coding, rate adaptation, JPEG2000, Integer Linear Programming.**

I. Introduction

In the last years, scalable video coding emerged as a promising technology for efficient distribution of videos through heterogeneous networks, and it has been recently standardized as scalable extension of the H.264/AVC standard [1], hereafter indicated as SVC. An useful overview of the SVC extension can be found in [2]. The main advantage of SVC is that it offers coding flexibility to decode different "working points" in terms of spatial, temporal and quality resolution from a unique coded representation. In a heterogeneous environment typically the video content needs to be *adapted* in order to meet different end terminal capability requirements (user adaptation) or fluctuations of the available bandwidth (network or rate adaptation). In this work we address only the rate adaptation problem. The scalability features given by the scalable video coding offer a very flexible way to perform the *adaptation*, for example reducing the spatial resolution or the video quality. In particular, with the SVC scalability features, rate adaptation can be efficiently managed using quality scalability, i.e. coarse grain (CGS) or fine grain (FGS) scalability. Although the intrinsic support for adaptation provided with the scalability, the problem of the adaptation for scalable video content is an open research issue that is still under investigation.

An exhaustive survey on the proposed approaches for solving the adaptation problem can be found in [3]. In [3] different classification criteria and properties of the adaptation methods are presented. For the purpose of the present work, three aspects have to be considered when comparing different adaptation methods:

- the performance, evaluated in terms of decoded quality (depending on a particular metrics) of the extracted data.
- the satisfaction of additional constraints on the decoded video sequences.
- the complexity of the adaptation process.

Among the different approaches presented in [3] the first and most attractive approach proposed for rate adaptation with SVC is that proposed in [4]. Recently, another interesting approach has been proposed in [5]. In [5] the authors compare their approach to that in [4] and they show that the two approaches have similar extraction performance. Since the implementation of the approach in [4] is included in the SVC reference software (JSVM 9, version 9.14) [6] we decide to use only this approach as reference for evaluating the performance of our method, described in the following sections.

Although SVC is the reference standard for scalable video coding, other scalable video coding approaches has been proposed in literature. Most of them are based on the wavelet technology, that has native spatial scalability features. Furthermore, JPEG2000 is the state-of-the-art in still image compression and it offers very efficient spatial and SNR scalability. Inspired by the potentialities of JPEG2000 and the previous wavelet approaches we have proposed a very simple but efficient new scalable video coding solution based on JPEG2000. In Section II we define the architecture of the proposed codec and the main features of JPEG2000 that are fundamental to understand

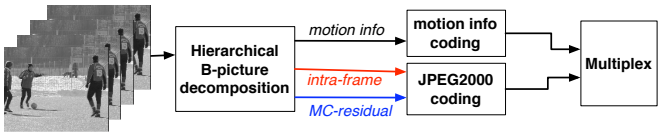


Fig. 1. Proposed JPEG2000-based codec architecture

the proposed adaptation solution. At present, the proposed architecture is still under investigation relatively to the spatial scalability. Thus, in this work, we only focus on the SNR scalability, also avoiding to consider the temporal scalability for adaptation purposes.

The main idea behind the proposed approach is that the rate adaptation problem for SNR scalable coding can be seen as an "optimal resources allocation" problem and thus formulated using Integer Linear Programming (ILP) as follows:

$$\begin{aligned} &\text{minimize} && \mathbf{c}^T\mathbf{x} \\ &\text{subject to} && \begin{cases} A\mathbf{x} \geq b \\ \mathbf{x} \geq 0 \text{ integer} \end{cases} \end{aligned} \tag{1}$$

where $\mathbf{c}^T\mathbf{x}$ is the objective function and describes the target of the adaptation problem, $A\mathbf{x} \geq b$ represents the constraints given by the coding system and additional constraints on the decoded video sequence and $\mathbf{x}$ is a vector of binary unknowns describing how resources are allocated.

In Section III we describe the proposed integer programming problem for rate adaptation and underline some of its properties. Although NP-hard, the problem (1) can be efficiently solved by means of a commercial software for mixed integer linear programming such as CPLEX and exploiting particular properties of the model. Finally, Section IV is devoted to compare our approach performance with that of the method in [4], whereas in Section V conclusions and possible future developments are drawn.

II. Proposed codec solution

The proposed scalable video coding system, that at present only offers temporal and SNR scalability features, is based on a very simple architecture shown in Figure 1. The input video sequence is temporally decomposed using a similar approach to Hierarchical B-picture decomposition proposed for SVC (see [2] and [7]), that enables closed-loop motion estimation and native temporal scalability. Hierarchical B-picture decomposition is based on the definition of key picture and group of picture (GOP). The first picture of a video sequence is an intra-coded picture, while the other key-pictures could be intra-coded or inter coded using the previous key pictures as reference. Usually key-pictures are coded at regular intervals, and the distance between two key-pictures defines the GOP length. In fact a key picture and all pictures that are temporally located between the key picture and the previous key picture are considered to build a group of pictures. Figure 2 shows a typical decomposition structure based on hierarchical B-picture for a GOP with length equal to 8. This particular GOP structure enables 4 levels of temporal scalability. In the first one, which we call level 0 and represents the sequence at its lowest available frame rate, we consider only the sequence made of key pictures, while in a generic level i ($i \geq 1$) we consider the pictures used at lower temporal levels plus the pictures indicated in Figure 2 as "B level i" pictures. For each GOP there is only 1 "B level 1" picture and it is predicted using the key picture of the previous GOP for forward prediction and the key picture of the same GOP for backward prediction. For $i > 1$ the "B level i" pictures are predicted using the pictures belonging to the lower temporal resolution. It should be noticed that the hierarchical B-picture structure enables closed-loop coding. In fact, the encoding order is different from the display order. The first encoded picture is the first frame of the sequence, then for every GOP we encode the key picture before motion estimation and compensation for the "B level 1" picture. Then every "B level i" picture is encoded before motion estimation and compensation of the "B level $i+1$" pictures. This encoding order ensures that at each temporal level the motion estimation and compensation process uses the already encoded reference pictures.

In the proposed scalable video codec the hierarchical B-picture decomposition is adopted with the only constraint that the key-picture of each GOP can be only intra-coded.

After the temporal decomposition, for each GOP, the key-picture and the moto-compensation residual for all the B-pictures within a GOP are encoded with a JPEG2000 framework as a single picture. SNR scalability is obtained generating JPEG2000 codestreams with multiple quality layers. Since the proposed work addresses only the SNR scalability, in the following we overview the main features of JPEG2000 that are fundamental to understand the model proposed in Section III.

JPEG2000 is the state-of-the-art in image compression and is based on the Discrete Wavelet Transform (DWT), together with Embedded Block Coding with Optimized Truncation (EBCOT) [8]. D stages of DWT analysis decompose the image into $3D$+1 subbands, labeled LH_d, HL_d, HH_d and LL_D, for $d = 1, ..., D$. An useful overview of the main features of JPEG2000 can be found in [9], while for a complete technical description the reader is referred to [10]. Each subband is partitioned into rectangular blocks called code-blocks, each of which is independently coded. Resolution scalability is obtained by discarding the code-blocks of detail subbands and omitting the final DWT synthesis stage. Quality scalability is obtained through a "quality layers" abstraction. Each layer represents an incremental contribution (possibly empty) from the embedded bit-stream associated with each code-block in the image. Discarding one or more layers (starting from the highest one) produces a representation of the code-block with lower quality. JPEG2000 also defines collections of spatially adjacent code-blocks as "precincts". Each precinct of resolution level LL_d consist of the code-blocks corresponding to the same spatial region within the subbands LH_{d+1}, HL_{d+1} and HH_{d+1} if $d < D$, or within the subband LL_D if $d = D$. The data-stream associated with each precinct is organized as a

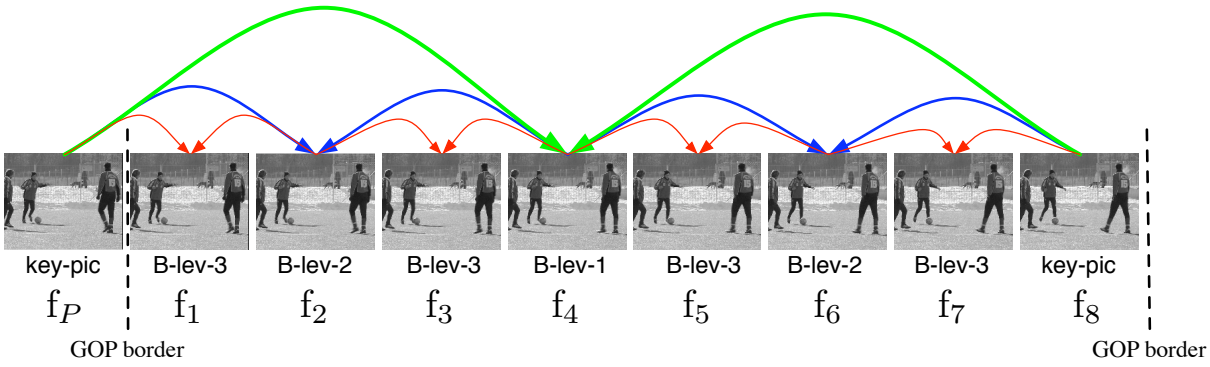


Fig. 2. Hierarchical B-pictures temporal decomposition with GOP size equal to 8

collection of "packets", one for each quality layer.

III. Rate extraction optimization

As previously described, the rate adaptation problem can be formulated as a problem of optimal allocation of resources. Since in the proposed scalable video codec the frames (intra-frames and moto-compensated residual) are encoded using JPEG2000, the "resources" that have to be allocated are the JPEG2000 coding elements. In the proposed integer model the minimum addressable coding element is the JPEG2000 packet, whereas we avoid to consider for the adaptation the motion information.

In the proposed approach the adaptation is performed independently on each GOP (with length equal to F frames and F is a power of 2), and consequently the optimal allocation of the JPEG2000 packets has to be performed in order to minimize the overall distortion over the GOP, given by:

$$D_{TOT} = \sum_{t=1}^{F} D^t \tag{2}$$

where D^t is the distortion for the frame t evaluated as MSE between the original and the decoded frame. The total distortion (2) represents the objective function $\mathbf{c}^T\mathbf{x}$ of the model (1). The "budget" available for the allocation is given by the bandwidth or equivalently by the data-rate. Given the GOP length and the frame-rate of the sequence, it is always possible to determine the amount of bits L that can be used to encode each GOP, where constant bit-rate transmission is considered.

In section III-A the expressions for the distortion of the key-picture and the B-pictures within each GOP will be provided, avoiding the mathematical details that can be found in [11], while in section III-B the ILP model will be presented.

A. Distortion computation

First, we introduce the distortion contribution given by a single JPEG2000 packet. Let I be the number of precincts in each frame (supposed to be constant over the frames), K the number of quality layers included in the data-stream associated with each precinct and F the GOP length (in frames). We define the following quantities: $\mathcal{P}_i^t$ a generic precinct i, $i = 1, ..., I$, belonging to frame t, $t = 1, ..., F$, $\mathcal{P}_i^{t,k}$ the decoded version of $\mathcal{P}_i^t$ using k, $k = 1, ..., K$, quality layers and $\mathcal{L}_i^{t,k}$ the size (in bits) of the first k packets related to the precinct $\mathcal{P}_i^t$. The distortion introduced approximating $\mathcal{P}_i^t$ with $\mathcal{P}_i^{t,k}$ is given by $D_i^{t,k} = \|\mathcal{P}_i^t - \mathcal{P}_i^{t,k}\|^2$. Introducing the rate-distortion slope $\mathcal{S}_i^{t,n}$, defined as the ratio $\Delta\mathcal{D}/\Delta\mathcal{L}$ related to each quality layer of each code-block, the distortion $D_i^{t,k}$ can be expressed as:

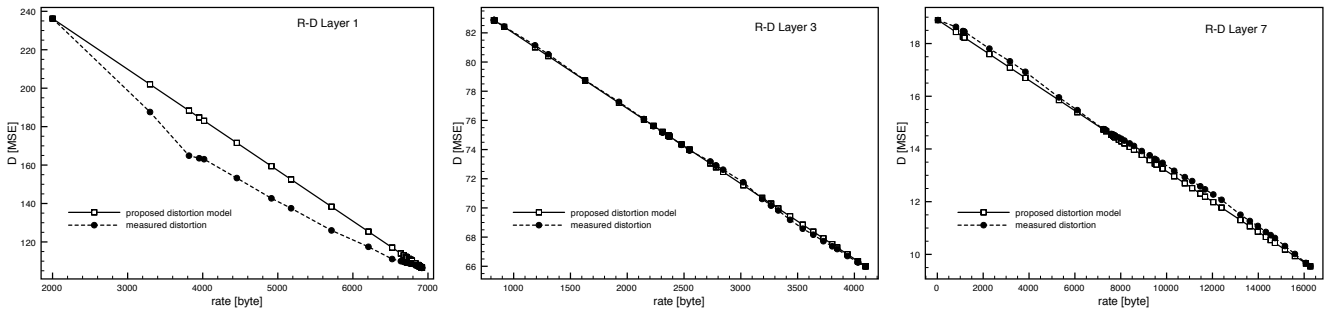
$$\mathcal{D}_i^{t,k} = \mathcal{D}_i^{t,K} + \sum_{n=k+1}^{K} \mathcal{S}_i^{t,n}\Delta\mathcal{L}_i^{t,n} \tag{3}$$

where $D_i^{t,K}$ is the distortion experienced (possibly equal to 0) if all the quality layers are considered, $\Delta\mathcal{L}_i^{t,n} = \mathcal{L}_i^{t,n} - \mathcal{L}_i^{t,n-1}$ is the size (in bits) of the quality layer n and $\mathcal{S}_i^{t,n}\Delta L_i^{t,n}$ is the distortion contribution given by the layer n. Unfortunately, the exact calculation of the distortion introduced for each precinct $\mathcal{P}_i^t$ require the knowledge of the rate-distortion slopes $\mathcal{S}_i^{t,n}$. Typically, in order to reduce the overhead required to maintain this slope information, only the rate-distortion slope threshold values used for the layer generation are included into the JPEG2000 codestream header (see [10] for more details). This means that the rate distortion slope for a particular layer n is considered to be constant over the precincts and equal to the threshold $\mathcal{T}_t^n$. Using the rate-distortion slope thresholds the distortion (3) can be approximated as:

$$\hat{\mathcal{D}}_i^{t,k} = \mathcal{D}_i^{t,K} + \sum_{n=k+1}^{K} \mathcal{T}_t^n\Delta\mathcal{L}_i^{t,n} \tag{4}$$

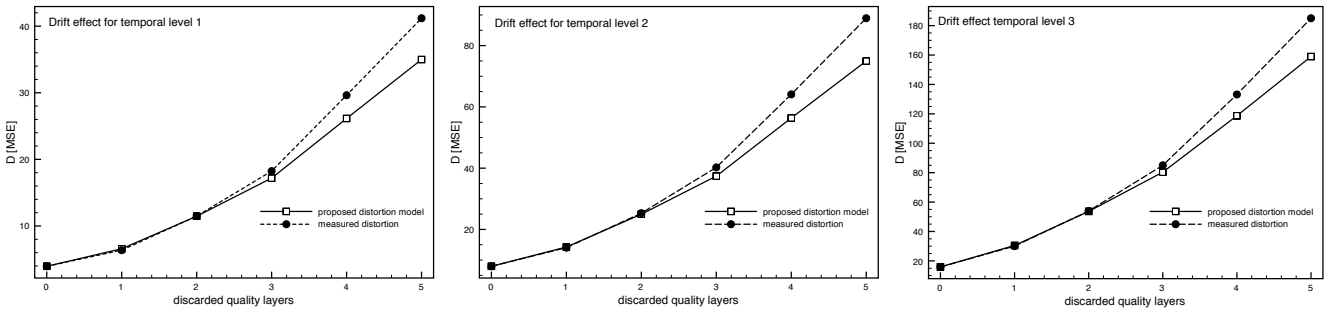
In order to estimate the overall distortion D_{TOT}, the main assumption that is considered hereafter is that the JPEG2000 packets are independent, that is approximately true for the 9/7 tap biorthogonal filters typically used in JPEG2000. This leads to an additive model for the distortion of a frame so that the sum of all the precincts distortion contribution $\sum_{i=1}^{I}\mathcal{D}_i^t$, where the single contribution $\mathcal{D}_i^t$ is estimated using the equation (4), can be considered as a good approximation of D^t.

In order to estimate the distortion of a generic frame t inside



(a) R-D performance of layer 1 (b) R-D performance of layer 3 (c) R-D performance of layer 7

Fig. 3. Typical Rate-Distortion performance of a key-picture encoded with 8 quality layers



(a) Drift effect on temporal level 1 (b) Drift effect on temporal level 2 (c) Drift effect on temporal level 3

Fig. 4. Analysis of the drift distortion contribution on B-frames at different temporal levels

each GOP, different expressions has to be considered for intra-frames (key-pictures) and inter-frames (B-pictures). To correctly define these quantities in view of their use inside the integer problem, we need to introduce the problem variables as follows. Let $\underline{\mathbf{x}} = (\mathbf{x^1}, ..., \mathbf{x^F})$ be a vector of binary variables where $\mathbf{x^t}$ is given by:

$$\mathbf{x^t} = \left(x_1^{t,1}, \ldots, x_1^{t,K}, x_2^{t,1}, \ldots, x_2^{t,K}, \ldots, x_I^{t,1}, \ldots, x_I^{t,K}\right)$$

Each binary variable $x_i^{t,k}$ assumes value 1 if the JPEG2000 packet made by the layer k of the precinct $\mathcal{P}_i^t$ is considered in the decoding process and 0 otherwise.
Intra-frames are managed as still pictures, and this leads to a straightforward expression for the distortion estimation:

$$\hat{D}^F = \mathcal{D}^{F,K} + \sum_{i=1}^{I}\sum_{k=1}^{K}(1 - x_i^{F,k})\mathcal{T}_F^k \Delta\mathcal{L}_i^{F,k} \qquad (5)$$

where we recall that with the hirarchical B-picture decomposition the key-picture is always the last picture of each GOP and consequently it is in position F, and that $\mathcal{D}^{F,K}$ is the key picture distortion if all the quality layers are decoded. The correctness of the expression (5) has been experimentally validated, and the results are reported in Figure 3 which is referred to a key-picture encoded with 8 quality layers. Figure 3 compares the distortion model given by the equation (5) to the real distortion experimentally measured. Figures 3(a), 3(b) and 3(c) show respectively the Rate-Distortion performance for the quality layer 1,3 and 7. For a particular layer, each point in the R-D curve is the contribution given by a precinct, i.e. the contribution of a packet, to the distortion reduction of the whole frame. As it can be noticed in Figure 3 the distortion model given by equation (5) is quite accurate, except for the very first quality layer. For the purpose of the rate adaptation, this distortion model inconsistency in the lowest quality layer is negligible, since in typical adaptation scenario the lowest quality layer is always fully considered in the decoding process, and, as it can be noticed from Figure 3(a), the overall distortion contribution given by layer 1 is correctly estimated with the model (5).

The expression of the distortion for an inter-frame is not easy to derive, since the hierarchical B-picture decomposition with closed-loop motion estimation introduces complex relationships between the distortion on the inter-frames at a particular level of temporal decomposition and the distortion on the reference frames of the higher temporal levels.
Briefly, the distortion on the inter-frames depends on two factors: the distortion on the motion-compensated residual related to the inter-frame and the "drift" effect that depends on the fact that in SNR scalability different "quality version" of the reference frames could be used for motion compensation. If the quality version that has been used at the encoder for the motion estimation and compensation process is not available at the decoder, for example because parts of the video bitstream are discarded for rate adaptation purpose, the drift is introduced. The drift problem can be reduced using a low quality version of the reference frames for motion

$$\hat{D}_\tau = \overbrace{\sum_{y=1}^{2^{\tau-1}} \mathcal{D}^{\lambda,K}}^{\text{initial distortion}} + \overbrace{\sum_{y=1}^{2^{\tau-1}} \sum_{i=1}^{I} \sum_{k=1}^{K} (1 - x_i^{\lambda,k}) \mathcal{T}_\lambda^k \Delta\mathcal{L}_i^{\lambda,k}}^{\text{motion-compensated residual component}} + \overbrace{\sum_{z=1}^{\tau-1} \sum_{w=1}^{2^{z-1}} 2^{\tau-2-z} \left[\sum_{i=0}^{I-1} \sum_{k=1}^{H} (1 - x_i^{\lambda_z,k}) \mathcal{T}_{\lambda_z}^k \Delta\mathcal{L}_i^{\lambda_z,k} \right]}^{\text{drift component from lower temporal layers}} +$$

$$+ \underbrace{2^{\tau-3} \sum_{i=1}^{I} \sum_{k=1}^{K} (1 - x_i^{F,k}) \mathcal{T}_F^k \Delta\mathcal{L}_i^{F,k}}_{\text{drift component from GOP key-picture}} + \underbrace{2^{\tau-3} \sum_{i=1}^{I} \sum_{k=1}^{K} (1 - x_i^{P,k}) \mathcal{T}_P^k \Delta\mathcal{L}_i^{P,k}}_{\text{drift component from previous GOP key-picture}} \quad (6)$$

estimation and compensation, but it is well know that this choice decreases the prediction efficiency and consequently the coding performance. Therefore, the common adopted solution is to use the highest available quality version, i.e. considering all the quality layers, for the reference frames in motion estimation and compensation process, and to accept the drift effect. Furthermore, with the hierarchical B-picture temporal decomposition, the drift effect also depends on the temporal level. Intuitively, referring to Figure 2, it is clear as the drift component of the distortion introduced on the "B-level-1" picture f_4 depends only on the key-pictures f_P and f_8, the drift effect introduced on the "B-level-2" picture f_2 depends on the the key-picture f_P and the "B-level-1" picture f_4 (that is also affected by f_8), and so on.

Although we intentionally avoid the mathematical details, it can be shown that a good approximation of the overall distortion on the "B-level-τ" frames, $\tau = 1, ..., \log_2 F$, is given by equation (6), where

$$\lambda = s + (y-1)\delta \qquad \delta = \frac{F}{2^{\tau-1}} \quad s = \frac{F}{2^\tau}$$

$$\lambda_z = s_z + (w-1)\delta_z \qquad \delta_z = \frac{F}{2^{z-1}} \quad s_z = \frac{F}{2^z}$$

It is important to note as in expression (6) it has been considered that the highest available quality version of the reference frames has been used for motion estimation and compensation, as previously described. This means that the three drift components are affected by all the missing quality layers up to K.

Although the proof of the expression (6) can be found in [11], the expression has been experimentally validated, and the results are shown in Figure 4. Figure 4 shows the distortion on the frames that belong to different temporal levels, assuming that the motion-compensated residual if fully considered in the decoding process. Consequently, the distortion is given by the effect of the three drift components and the initial distortion (not equal to 0 since lossy coding is considered). In order to show the effect of the drift and its approximation given by equation (6), we discard whole quality layers of the reference frames in the decoding process, from 0 discarded layers (no drift) to 5 quality layers, where the encoding process has been performed generating 8 quality layers.

As it can be noticed in Figure 4 the distortion model given by equation (6) is accurate assuming to discard up to 3 quality layers, a model error of approximately 10% is evidenced for the fourth discarded layer while a greater error is introduced from the fifth discarded layer. Nevertheless, some consideration has to be done in order to justify the error introduced. First, the aim of the proposed work is not to give a quantitative expression of the GOP distortion, but to provide a consistent approximation in order to use it for the rate adaptation model described in the following section. Furthermore, the approximation error is introduced only if we discard many quality layers, that means that most of the original video bitstream has to be discarded for rate adaptation purposes. Nevertheless, the typical rate adaptation scenario that has been considered in the simulations performed assumes to discard from 30% to 50% of the compressed video bitstream. It has been verified that in this range the proposed distortion models are accurate.

B. The ILP model

We now describe how modeling the optimal extraction as an ILP problem. The vector $\underline{\mathbf{x}}$ of binary variables has already been defined. We use binary variables since, in order to optimize the decoding performance, we assume to decode only full packets and the situation in which part of packets are forwarded or discarded is not taken into account. The objective function of the ILP problem is represented by the overall distortion (2) that, using the distortion models introduced in the previous section, can be approximated as

$$\hat{D}_{TOT} = \sum_{t=1}^{F} \hat{D}^t = \hat{D}_F + \sum_{\tau=1}^{\log_2 F} \hat{D}_\tau \quad (7)$$

where $\hat{D}_F$ and $\hat{D}_\tau$ are respectively given by equations (5) and (6).

In the basic ILP model two types of constraints can be identified: the quality layer constraints and the budget constraint. By the progressive layers generation process of the EBCOT algorithm, the extraction has to verify the following conditions:

$$x_i^{t,k} - x_i^{t,k+1} \geq 0 \quad (8)$$

with $i = 1, ..., I$, $k = 1, ..., K-1$ and $t = 1, ..., F$. As previously described, the amount of packets that have to be discarded to perform the adaptation depends on the available bandwidth, that could be converted in terms of bits (L) available for each GOP. Consequently, the following budget constraint has to be considered:

$$\sum_{t=1}^{F} \sum_{i=1}^{I} \sum_{k=1}^{K} x_i^{t,k} \Delta\mathcal{L}_i^{t,k} \leq L \quad (9)$$

The exact solution of the described ILP problem provides the Rate-Distortion optimal way to adapt the bitstream for each GOP. Additionally, further constraints could be introduced in order to satisfy particular decoding requirements. For example, in the performed tests the video sequences have been encoded in order to enable near constant decoded quality if the full video bitstream is decoded. Typically, after the rate adaptation process this feature is not maintained. In order to limit the decoded quality fluctuation potentially introduced by the adaptation, the following constraints could be introduced in the ILP model:

$$\beta\frac{\hat{D}_{TOT}}{F} \leq \hat{D}^t \leq \gamma\frac{\hat{D}_{TOT}}{F} \qquad t = 1, ..., F \qquad (10)$$

where $\beta \leq 1$ and $\gamma \geq 1$. The constraints (10) controls the fluctuation of the distortion on the single frame with respect to the mean distortion over the GOP. Adjusting the values β and γ it is possible to control the level of the distortion fluctuations. Nevertheless, it is important to note that the introduction of the constraint (10) does not guarantee that the ILP problem have a solution for all the values of β and γ. Furthermore, it is expected that the decoding performance will be reduced in order to satisfy the constraints. The effect of the constraints (10) on the decoding performance will be analyzed in section IV.

In contrast to Linear Programming (LP) problems, which can be efficiently solved, ILP problems are typically NP-hard thus requiring a computational time which increases exponentially with the problem size. The analyzed problem is NP-hard. Nevertheless, it has a particular structure. The matrix associated to the quality layer constraints (8) can be shown to be Totally Unimodular (TUM) (a proof is provided in [11]). ILP problems with TUM constraint matrix and integer right-hand-sides can be solved very efficiently, since the optimal solution of the related LP problem (obtained relaxing the constraint that variables $\mathbf{x}$ are integer) corresponds to the optimal integer solution. However, it has to be noted that the constraints matrix of our problem is not TUM, since we have the extra budget constraint (9) and eventually the constraints (10) for the distortion control. Nevertheless, the constraint (9) is a classical knapsack constraint, that can be efficiently managed by common solvers for mixed integer linear programming problems as CPLEX. Different considerations has to be done for the constraints (10), that, depends on the values of β and γ, could decrease the model resolution efficiency.

IV. Experimental validation

To validate our method we compare it to the approach proposed in [4]. It is worth noticing that the two approaches are applied on different scalable video codec, i.e. SVC and the proposed JPEG2000-based video codec. In order to compare the different coding performance of the two systems Figures 5(a) and 5(b) show the comparison of the two codecs in single layer mode, i.e. generating a non-scalable bitstream. As it can be noticed, SVC has better compression performance. A deeper investigation shows that this difference depends on the features of the video sequence.

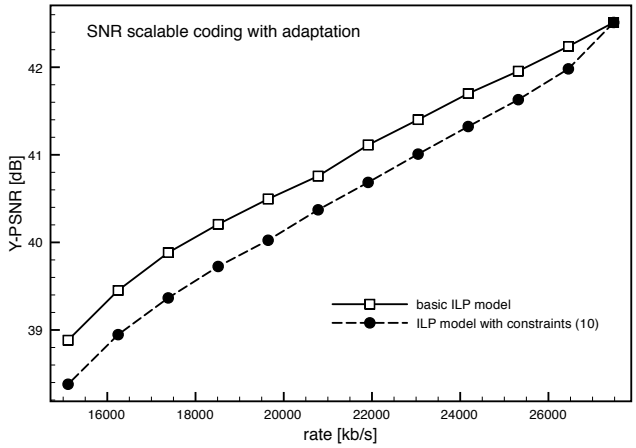
Two aspects are evaluated related to the SNR extraction performance: the mean PSNR over the frames obtained extracting sub-bitstream at different data-rate from the full SNR scalable bitstream, and the PSNR fluctuations (evaluated as the PSNR standard deviation) between the frames. The PSNR fluctuations are evaluated since the method proposed in [4] operationally starts by discarding parts of the full scalable bitstream from the lower temporal layers. This approach enables to maximize the mean distortion over the frames but could introduce a fluctuation of the video quality that generates annoying visual artifacts.

As previously described, due to the discrepancy between the real distortion and the proposed distortion models, the rate adaptation is performed in order to discard up to the 50% of the full video bitstream. As shown in Figures 5(c) and 5(d), the proposed extraction model enables comparable performance for Soccer sequence, and better performance for Harbour sequence. However, for both the sequences the coding performance in SNR extraction scenario increases compared to the single layer scenario. Furthermore, Figures 5(e) and 5(f) show as the proposed approach maintains limited the PSNR fluctuations compared to [4], especially at higher bitrates.

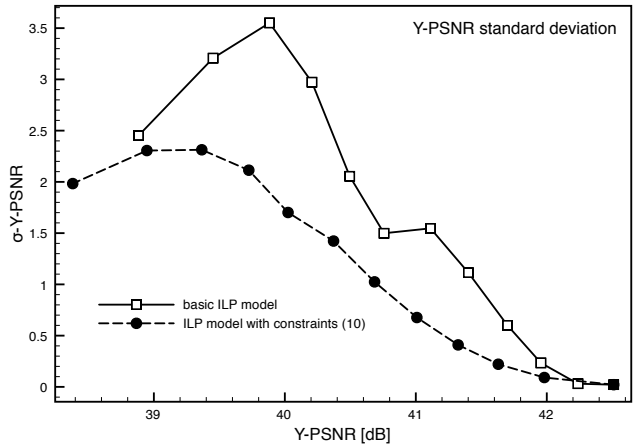
It has to be noted that at lower bitrates the extraction performance of the proposed method decreases compared to SVC. This not only depends on the lower coding efficiency of our codec compared to SVC, but also by the consideration that the proposed distortion models became inaccurate when many quality layers have to be discarded, as described in section III-A, generating a non-optimal extraction.

Relatively to the additional constraints (10), in Figure 6 is shown the effect of the constraints on the extraction performance and on the PSNR fluctuations, where the distortion fluctuations have been controlled setting $\beta = 0$ and $\gamma = 1.1$. Figure 6(a) shows how the additional constraints decrease the extraction performance of approximately $0,5dB$, but, as shown in Figure 6(b), with a considerable reduction of the PSNR fluctuations.

As a final remark, it is important to note as the computational time of the proposed adaptation approach is negligible. For example, in the test reported in Figure 5 we used a video sequence at 4CIF resolution (704x576 pixel), a GOP length equal to 8 frames, and we configured the JPEG2000 encoder in order to have 5 levels of resolution, precinct size equal to 64×64 pixel and 8 quality layers. This leads to approximately 3000 JPEG2000 packets (equal to the problem size) for each GOP. For each GOP the optimal allocation of the rate, obtained solving the ILP problem with the CPLEX version 8.1 [12], is performed in fractions of seconds, approximately 2 or 3 tenths of a second. This is mainly due the special structure of the described problem. Similar computational time can be obtained also increasing the problem size, for example for video at High Definition (HD) resolution. Furthermore, even in the test reported in Figure 6 when the additional constraints (10) have been added to the model, the optimal allocation is performed in similar computational time. This is mainly due to the fact that the distortion threshold $\gamma = 1.1$ is not very



(a) Harbour SNR extraction



(b) Harbour Y-PSNR standard deviation

Fig. 6. Analysis of the effect of the constraints (10)

restrictive. Decreasing its value increases the computational time required to solve the model.

V. Conclusions

In this work we propose an efficient method for SNR scalable video adaptation based on the formulation of the adaptation problem as an Integer Linear Programming problem and successfully applied to a JPEG2000-based scalable video codec. The proposed approach shows two very interesting features. First, it provides a comparable performance with respect to the adaptation method used for SVC. Secondly, the TUM property of part of the constraints matrix of the proposed ILP problem can be exploited to find efficient approximated solutions (for instance, by means of Lagrangian Relaxation) to more complex adaptation problems where additional constraints such as a further control on the distortion fluctuation are introduced.

References

[1] "Advanced video coding for generic audiovisual services," ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), ITU-T and ISO/IEC JTC 1, Version 1: May 2003, Version 2: May 2004, Version 3: Mar. 2005, Version 4: Sept. 2005, Version 5 and Version 6: June 2006, Version 7: Apr. 2007, Version 8 (including SVC extension): Consented in July 2007.

[2] H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the Scalable Video Coding extension of the H.264/AVC standard," *IEEE Transaction on Circuit and System for Video Technology*, vol. 17, no. 9, pp. 1103–1120, 2007.

[3] T. Thang, J.-G. Kim, J. W. Kang, and J.-J. Yoo, "SVC adaptation: Standard tools and supporting methods," *Signal Processing: Image Communication*, 2009.

[4] I. Amonou, N. Cammas, S. Kervadec, and S. Pateux, "Optimized rate-distortion extraction with quality layers in the Scalable extension of H.264/AVC," *IEEE Transaction on Circuit and System for Video Technology*, vol. 17, no. 9, pp. 1186–1193, 2007.

[5] J. Sun, G. Wen, D. Zhao, and W. Li, "On Rate-Distortion Modelling and Extraction of H.264/SVC Fine-Granular Scalable Video," *IEEE Transaction on Circuit and System for Video Technology*, vol. 19, no. 3, pp. 323–336, 2009.

[6] ITU-T, "JSVM 10 software," jVT-W203, April 2007.

[7] H. Schwarz, D. Marpe, and T. Wiegand, "Comparison of MCTF and closed-loop hierarchical B pics," jVT input document P059, July 2005.

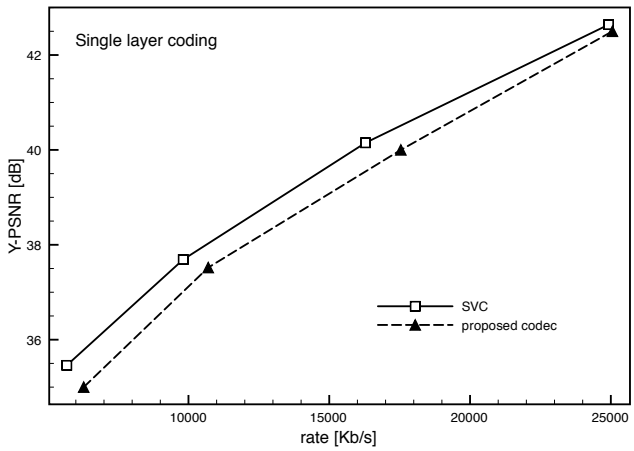
[8] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Transaction on Image Processing*, vol. 9, no. 7, pp. 1158–1170, July 2000.

[9] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The JPEG2000 still image coding system: An overview," *IEEE Transaction on Consumer Electronics*, vol. 46, no. 1, pp. 1103–1127, November 2000.

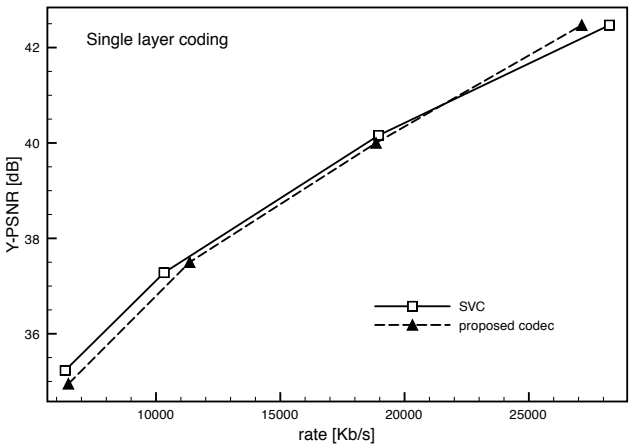
[10] D. Taubman and M. Marcellin, *JPEG2000: Image Compression Fundamentals, Standards and Practice*. Kluwer Academic, Boston, MA, USA, 2002.

[11] L. Lima, "Scalability of visual information for improved communication," PhD thesis, 2009. Shortly available at www.ing.unibs.it/ livio.lima/.

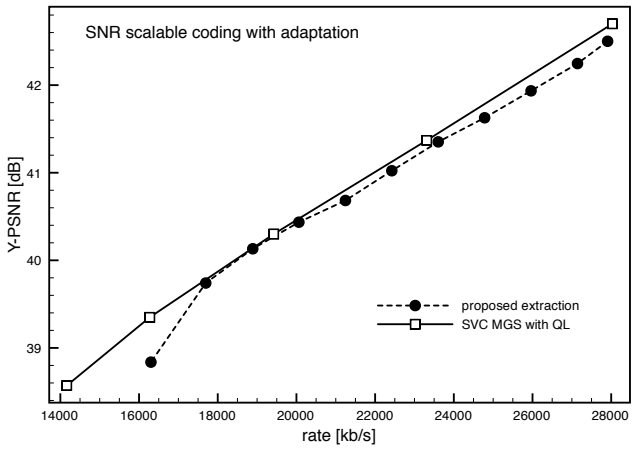
[12] I. INC., "ILOG CPLEX 8.1 reference manual," Incline Village: ILOG Inc., CPLEX Div., 2002.



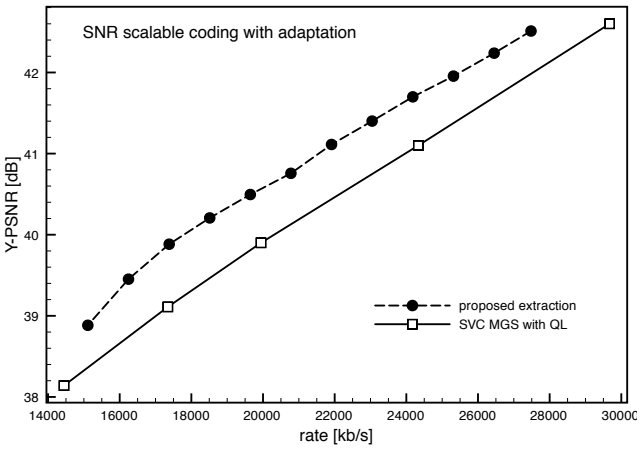
(a) Soccer single layer



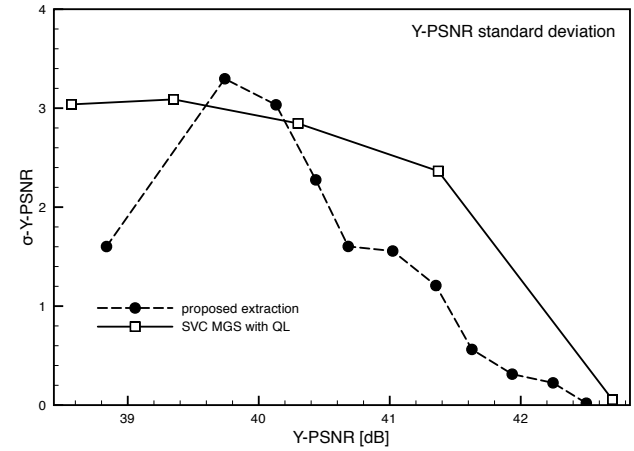
(b) Harbour single layer



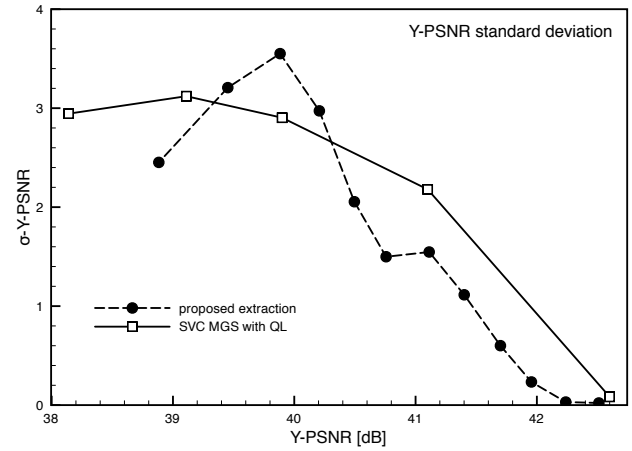
(c) Soccer SNR extraction



(d) Harbour SNR extraction



(e) Soccer Y-PSNR standard deviation



(f) Harbour Y-PSNR standard deviation

Fig. 5. Analysis of the extraction performance