Sergio Sanz-Rodríguez ; Fernando Díaz-de-María

RBF-based QP estimation model for VBR control in H.264/SVC

Suggested Citation
Sanz-Rodríguez, Sergio ; Díaz-de-María, Fernando: RBF-based QP estimation model for VBR control in H.264/SVC. - In: IEEE transactions on circuits and systems for video technology : a publication of the Circuits and Systems Society. - ISSN: 1558-2205 (online). - 21 (2011), 9. - pp. 1263-1277. - DOI:10.1109/TCSVT.2011.2143330. *(Postprint version is cited, page numbers differ.)*

RBF-based QP Estimation Model for VBR Control in H.264/SVC

Sergio Sanz-Rodríguez, *Student Member, IEEE,* Fernando Díaz-de-María, *Member, IEEE*

***Abstract*—In this paper we propose a novel variable bit rate (VBR) controller for real-time H.264/scalable video coding (SVC) applications. The proposed VBR controller relies on the fact that consecutive pictures within the same scene often exhibit similar degrees of complexity, and consequently should be encoded using similar quantization parameter (QP) values for the sake of quality consistency. In oder to prevent unnecessary QP fluctuations, the proposed VBR controller allows for just an incremental variation of QP with respect to that of the previous picture, focusing on the design of an effective method for estimating this QP variation. The implementation in H.264/SVC requires to locate a rate controller at each dependency layer (spatial or coarse grain scalability). In particular, the QP increment estimation at each layer is computed by means of a radial basis function (RBF) network that is specially designed for this purpose. Furthermore, the RBF network design process was conceived to provide an effective solution for a wide range of practical real-time VBR applications for scalable video content delivery.**

In order to assess the proposed VBR controller, two real-time application scenarios were simulated: mobile live streaming and IPTV broadcast. It was compared to constant QP encoding and a recently proposed constant bit rate (CBR) controller for H.264/SVC. The experimental results show that the proposed method achieves remarkably consistent quality, outperforming the reference CBR controller in the two scenarios for all the spatio-temporal resolutions considered.

***Index Terms*—Rate Control, Variable Bit Rate (VBR), Scalable Video Coding (SVC), H.264/SVC, H.264/advanced video coding (AVC), IPTV, streaming.**

I. Introduction

VIDEO coding has become one of the paramount research areas in recent years, given the growing popularity of multimedia communications caused by the development and improvement of the network infrastructures, the storage capacity, and the processing power of decoding terminals. According to the target application, two different coding methods can be distinguished: constant bit rate (CBR) and variable bit rate (VBR) coding. In CBR coding, commonly used for real-time video conference, a short-term average bit rate adaptation is required to ensure low buffer delay. However, in VBR coding, typically used for video streaming or digital storage, a long-term bit rate adaptation and, consequently, a longer buffer delay, is allowed for improving the visual quality consistency [1], [2].

In order that encoded video sequences can be properly transmitted and decoded, the rate control (RC) algorithm located at the encoding side operates in two steps. First, a bit budget is allocated to each coding unit according to the video complexity, the target bit rate and the buffer constraints given by the hypothetical reference decoder (HRD) requirement [3]. Second, a quantization parameter (QP) value is assigned to the coding unit so that the buffer fullness is maintained at secure levels, while minimizing the distortion.

Several RC algorithms for CBR coding have been recommended in the video coding standards, such as the Test Model Version 5 for MPEG-2 [4], the Verification Model Version 8 for MPEG-4 [5], the Test Model Version 8 for H.263 [6], Joint Model for H.264/advanced video coding (AVC) [7]. Beyond these baseline algorithms, the RC problem has been extensively studied. Most of the approaches have focused on modeling the discrete cosine transform (DCT) coefficients, providing analytical rate-distortion (R-D) functions for QP estimation. For instance, assuming a Gaussian probability density function (PDF) for DCT coefficients, a logarithmic R-D function can be inferred [8]. Alternatively, assuming a Cauchy PDF, a simple exponential R-D model is derived [9], [10]. On the other hand, using a Laplacian PDF, different linear [11], quadratic [5] or ρ-domain-based [12] R-D models have been proposed. Furthermore, Chen *et al* [13] proposed separate R-D models for the luminance and chrominance components of color video sequences; and Xie *et al* [14] proposed a sequence-based RC method for MPEG-4 that uses a rate-complexity model to track the non-stationary characteristics in the video source.

With respect to VBR coding, several RC algorithms have been proposed to provide a more consistent visual quality in a variety of applications, such as live streaming and broadcast [15], [16], one-pass digital storage [17], [18], or two-pass digital storage [19], [20]. It should be noted that, for digital storage, the RC algorithm is subject to a budget constraint instead of to a delay constraint. Other schemes, such as [21] and [22], have also been proposed taking advantage that VBR video can be easily incorporated in a networking infrastructure that supports VBR transport [2], to improve the visual quality while reducing the buffer delay. From the R-D modeling point of view, instead of using the analytical models described above for real-time CBR applications, several methods have been proposed that relies on the estimation of a QP increment with respect to a reference QP in order to reduce its variation [16], [18], [22].

Finally, it is also worth mentioning that an optimal solution to the RC problem has also been studied. These methods, which are based on the operational R-D theory, can be only used in off-line applications. The reader is referred to [23] for more information on this approach.

Manuscript received XX X, 2010; revised XX XX, 20XX.

The authors are with the Department of Signal Theory and Communications, Universidad Carlos III de Madrid, Leganés, Madrid 28911 Spain (e-mail: {sescalona, fdiaz}@tsc.uc3m.es).

Nowadays, many video transmission services over RTP/IP-based channels, such as Internet or wireless networks, have benefited from scalable video coding (SVC) features [24], [25]. For these kinds of channels, SVC is able to provide bit rate adaptation for varying channel conditions as well as for heterogeneous devices with different display resolutions and computational capabilities. SVC enables the extraction of either one or a subset of sub-streams from a high-quality bit stream so that these simpler sub-streams, bearing lower spatio-temporal resolutions or reduced quality versions of the original sequence, can be decoded by a given target decoder. Furthermore, specific forward error correction techniques can be used to ensure an error free transmission of more important layers, such as the lowest spatio-temporal resolution.

The scalable extension of the H.264/AVC standard named H.264/SVC has recently been standardized [26] and evaluated [27]. It provides both coding efficiency and decoding complexity similar to those achieved using single-layer coding. H.264/SVC supports spatial, temporal, and quality scalable coding. For spatial scalability, a layered coding approach is used to encode different picture sizes of an input video sequence. The base layer provides an H.264/AVC compatible bit stream for the lowest spatial resolution, while larger picture sizes are encoded by the enhancement layers. In addition, the redundancies between consecutive spatial layers can be exploited via inter-layer prediction tools in order to improve the coding efficiency.

Each spatial layer is capable of supporting temporal scalability by using hierarchical prediction structures, which go from these very efficient ones using hierarchical B pictures to those with zero structural delay. The pictures of the base temporal layer can only use previous pictures of the same layer as references. The pictures of an enhancement temporal layer can be bidirectionally predicted by pictures of a lower layer. The number of temporal layers in a spatial layer is determined by the group of pictures (GOP) size, defined in H.264/SVC as the distance between two consecutive I or P pictures, also named *key* pictures.

For quality or signal-to-noise ratio (SNR) scalability, different reconstruction quality levels with the same spatio-temporal resolution are provided. The H.264/SVC standard defines two types of SNR scalable coding: coarse grain scalability (CGS) and medium grain scalability (MGS). The first is a special case of spatial scalability with identical picture sizes. The second employs a multilayer approach within a spatial layer in order to provide a finer bit rate granularity in the R-D space.

Given a video transmission service that offers several qualities of service (QoS) and serves heterogeneous decoding devices, a layered coding approach implies that the RC scheme must be able to provide a set of HRD-compliant scalable sub-streams considering a variety of target bit rates, one per target decoding terminal bearing a particular spatio-temporal resolution or computational capability. This is the aim of the different RC algorithms that have been proposed for SVC during the last years. Most of them employ well-know analytical R-D functions for QP estimation: linear [28], quadratic [29], the so-called *square root* [30], ρ-domain-based [31], [32], TMN8-based [33], and exponential [34], [35] models. The bit allocation formulation for hierarchical GOP structures has also been studied. In particular, the dependency among spatial, quality, and temporal layers has been exploited in [34] and [35], though these solutions are not suitable for real-time scenarios given the required number of encoding iterations. In [36], an optimal distribution of the total target bit rate among the spatial/CGS layers was determined. It is worth mentioning that the quality scalability was specially investigated for MPEG-4 fine grain scalability (FGS) [30], [37] and MGS [34], [38], [39]. Nevertheless, with a few exceptions [30], [37], [39], the existing RC approaches for SVC are not still developed for those VBR applications that can benefit from the SVC features for video content delivery.

In this paper we propose a novel VBR controller for real-time H.264/SVC applications. As suggested in [16] for H.264/AVC, the proposed VBR controller assumes that consecutive pictures within the same scene often show similar degrees of complexity, and aims to prevent unnecessary QP fluctuations by allowing just an incremental variation of QP with respect to that of the previous picture. In order to adapt this idea to H.264/SVC, a rate controller is located at each dependency layer (spatial or CGS), so that each rate controller is responsible for determining the proper QP increment. In particular, this paper focuses on providing an effective QP increment estimation computed by means of a radial basis function (RBF) network, which has been specially designed for this purpose.

The paper is organized as follows. In Section II a detailed description of the proposed RC algorithm is provided. First, a brief overview is given. Then, the two main stages of the rate controller for each dependency layer, parameter updating and RBF-based QP increment estimation, are described. Section III describes the design of the RBF network for QP increment estimation. Section IV shows and discusses the experimental results. Finally, some conclusions are drawn in Section V.

II. A Novel VBR Controller for H.264/SVC

A. System Overview

Before starting to describe the proposed VBR controller, the notation used along the paper has been summarized in Table I for reference. The RC scheme is illustrated in Fig. 1 for a SVC encoder consisting of two dependency layers. Let us denote as D the number of dependency layers, identified as $d = \{0, 1, \ldots, D-1\}$, and let us denote as $T^{(d)}$ the number of temporal layers for a particular dependency layer, identified as $t = \{0, 1, \ldots, T^{(d)} - 1\}$.

Each dependency layer d involves a rate controller $RC^{(d)}$ and a virtual buffer. The virtual buffer at layer d receives the contributions of layers from 0 to d and simulates the encoder buffering process of the corresponding sub-stream. The generation of each sub-stream depends on two fundamental parameters: the target bit rate $R^{(d)}$ and output frame rate $f_{out}^{(d)}$. It should be noted that $R^{(d)}$ must be higher than those associated with lower dependency layers, i.e.,

$$R^{(d-i)} \leq R^{(d)} \qquad i = 0, 1, \ldots, d,$$

since those lower dependency layers form part of the d^{th} sub-stream.

TABLE I
SUMMARY OF NOTATION.

d	Dependency layer identifier
D	Number of dependency layers
t	Temporal layer identifier
j	Current picture number
BD	Buffer size in seconds
nTF	Normalized target buffer fullness with respect to BD
L	Number of Gaussian-type functions
H	Gaussian-type function
$\mathbf{C}$, $\mathbf{\Sigma}$, $\mathbf{w}$, w_0	Centers, widths, weights, and bias of the RBF network
Ψ	Cost function for training data labeling
$(\lambda_1,\lambda_2,\lambda_3)^T$	Weight vector for Ψ
θ	Scale factor for Ψ
For each layer d	
$T^{(d)}$	Number of temporal layers
$RC^{(d)}$	Rate control module
$R^{(d)}$	Target bit rate
$f_{out}^{(d)}$	Output frame rate
$QP^{(d)}$	Quantization parameter value
$Q^{(d)}$	Quantization step value
$\Delta QP^{(d)}$	Quantization parameter increment
$BS^{(d)}$	Buffer size in bits
$V^{(d)}$	Buffer fullness
$nV^{(d)}$	Normalized $V^{(d)}$ with respect to $BS^{(d)}$
$G^{(d)}$	Access unit target bits
$AU^{(d)}$	Access unit output bits
$nAU^{(d)}$	Normalized $AU^{(d)}$ with respect to $G^{(d)}$
$b^{(d,t)}$	Texture bits of the picture with identifier (d,t)
$h^{(d,t)}$	Header plus motion data bits of the picture with identifier (d,t)
$\overline{C}_{TEX}^{(d,t)}$	Average texture complexity of the layer (d,t)
$\overline{C}_{MOT}^{(d,t)}$	Average motion complexity of the layer (d,t)
α, β	Forgetting factors for complexity computation
$G_{NOM}^{(d)}$	Nominal bit budget
$\Delta G_{TEX}^{(d)}$	Target bit increment for texture information
$\Delta G_{MOT}^{(d)}$	Target bit increment for motion information
$N^{(d,t)}$	Number of pictures per GOP with identifier (d,t)
$\mathbf{X}^{(d)}$	Input vector to the RBF network
$D_j^{(d)}$	Distortion of the j^{th} picture
$\overline{D}^{(d)}$	Average distortion

In order to encode the j^{th} picture with spatio-temporal identifier (d,t), the $RC^{(d)}$ module should provide an appropriate $QP_j^{(d)}$ value, on a frame basis, so that the QP fluctuation is minimized (to improve visual quality consistency), while the buffer fullness $V^{(d)}$ is maintained at secure levels. To this end, the $RC^{(d)}$ module operation leans on three input parameters:

1) The fullness $V^{(d)}$ of the corresponding virtual buffer.
2) The amount of bits yield by the encoding of the spatial layers 0 to d for a given time instant. Henceforth, following the H.264/SVC nomenclature, we will refer to this amount of bits as an *access unit* (AU) output bits $AU^{(d)}$.
3) The QP value used for encoding the previous picture of the same dependency layer $QP_{j-1}^{(d)}$.

A proper QP increment $\Delta QP^{(d)}$ is estimated from the two firsts, and $QP_{j-1}^{(d)}$ is employed as a reference value to obtain the final quantization parameter as follows:

$$QP_j^{(d)} = QP_{j-1}^{(d)} + \Delta QP^{(d)}. \quad (1)$$

This approach takes advantage of the fact that the VBR environments allow for a slow QP evolution in order to maintain a consistent visual quality. Thus, it assumes similarity between consecutive frames and aims to model only those QP changes required to compensate for large bit rate deviations owing to time-varying video complexity. Consequently, the method for estimating the QP increment becomes the main focus of the proposed VBR controller.

It is also worth noting that, in the case of CGS scalability, the QP obtained is lower bounded by the QP of the reference layer, so that a higher quality for the enhancement layer is ensured:

$$QP_j^{(d)} = \min\left[QP_j^{(d-1)}, QP_j^{(d)}\right]. \quad (2)$$

The VBR control algorithm for a specific spatial or CGS layer, i.e., the algorithm that obtains an appropriate QP increment for the j^{th} picture with identifier (d,t) is illustrated in Fig. 2. As shown in the figure, the $RC^{(d)}$ module is organized in two stages named *parameter updating* and *RBF-based QP increment estimation*:

- *Parameter updating stage*: after encoding the $(j-1)^{th}$ picture with layer identifier (d,t') (t' is used instead of t because the previous picture can belong to a different temporal layer), some parameters required to estimate the QP increment are updated. In particular, the following two parameters are updated: 1) a normalized version of the buffer fullness, denoted as $nV^{(d)}$; and 2) a normalized version of the amount of bits generated by the AU, denoted as $nAU^{(d)}$. The normalized versions of the buffer fullness and the AU output bits are defined as follows:

$$nV^{(d)} = \frac{V^{(d)}}{BS^{(d)}}, \quad (3)$$

$$nAU^{(d)} = \frac{AU^{(d)}}{G^{(d)}}, \quad (4)$$

  where $V^{(d)}$ has already been defined as the buffer fullness; $BS^{(d)}$ denotes the buffer size, in bits, associated with the d^{th} dependency layer; $AU^{(d)}$ has already been defined as the AU output bits; and $G^{(d)}$ denotes the AU target bits.
- *RBF-based QP increment estimation stage*: before encoding the j^{th} picture, the proper QP increment $\Delta QP^{(d)}$ is estimated from four parameters (whose selection is discussed in Subsection II-C1): $nV^{(d)}$, $nAU^{(d)}$, and two additional constant parameters that are included so that the achieved solution is able to work in a variety of scenarios. The first constant parameter, denoted as nTF, is the normalized target buffer fullness with respect to the buffer size, and the second, denoted as BD, is the maximum buffering delay (or buffer size in seconds), which is related to that measured in bits as $BS^{(d)} = BD \times R^{(d)}$. Then the $\Delta QP^{(d)}$ value is added to $QP_{j-1}^{(d)}$ as indicated in Eq. (1). In particular, a nonlinear relation between the aforementioned input parameters and the desired QP increment has been obtained by training an RBF network that is able to deal with a wide range of practical situations, as described in Section III.

Both stages are described in detail in the following subsections.

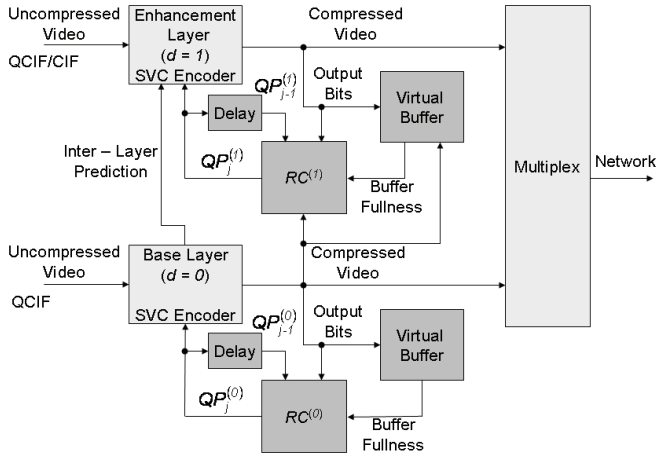

Fig. 1. Block diagram of the proposed H.264/SVC rate control scheme for two dependency layers ($D=2$).

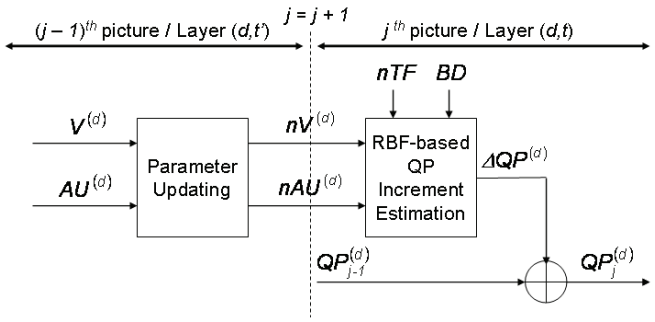

Fig. 2. Block diagram of the rate controller $RC^{(d)}$ for a specific dependency layer d. The $RC^{(d)}$ module is organized in two stages named *parameter updating* and *RBF-based QP increment estimation*. The first provides updated parameters for the second, which estimates the proper QP increment to end up with the QP value for the current picture, $QP_j^{(d)}$.

B. Parameter Updating

The aim of this subsection is to describe the updating procedure for parameters $nV^{(d)}$ and $nAU^{(d)}$. The updating equations for $nV^{(d)}$ and $nAU^{(d)}$ require the previous computation of both the buffer fullness and the AU target bits. In turn, the computation of the buffer fullness requires to obtain the AU output bits, and the estimation of AU target bits requires to estimate the average texture and motion complexities for each temporal layer. Therefore, the calculation of all of these quantities are described first, to end up with the updating equations for $nV^{(d)}$ and $nAU^{(d)}$.

1) Computation of AU Output Bits: Assuming that the picture coding order in SVC is established so that the AUs are sequentially encoded (the encoding of an AU starts when the previous has been completed) [26], the total number of bits generated by $AU_{j-1}^{(d,t')}$ obeys:

$$AU_{j-1}^{(d,t')}=\sum_{m=0}^{d}\left(b_{j-1}^{(m,t')}+h_{j-1}^{(m,t')}\right), \tag{5}$$

where $b_{j-1}^{(m,t')}$ and $h_{j-1}^{(m,t')}$ are, respectively, the amount of texture bits and header plus motion data bits generated by the $(j-1)^{th}$ picture, with spatio-temporal layer identifier (m,t').

2) Buffer Fullness Updating: Once the AU output bits have been obtained, the virtual buffer fullness is updated as follows:

$$V_j^{(d)}=V_{j-1}^{(d)}+AU_{j-1}^{(d,t')}-\frac{R^{(d)}}{f_{out}^{(d)}}. \tag{6}$$

3) Estimation of the Average Texture and Motion Complexities of a Layer (d,t'): Let us define $\overline{C}_{TEX}^{(d,t')}$ as the average texture complexity of the encoded pictures at spatial/CGS layers 0 to d belonging to the temporal layer t'. The following updating equation is proposed:

$$\overline{C}_{TEX}^{(d,t')}=\alpha\sum_{m=0}^{d}\left(Q_{j-1}^{(m)}b_{j-1}^{(m,t')}\right)+(1-\alpha)\overline{C}_{TEX}^{(d,t')}, \tag{7}$$

where α is a forgetting factor that is set to 0.5 in our experiments, and $Q_{j-1}^{(m)}$ is the quantization step value associated with $QP_{j-1}^{(m)}$. Likewise, the average motion complexity $\overline{C}_{MOT}^{(d,t')}$ is defined as:

$$\overline{C}_{MOT}^{(d,t')}=\beta\sum_{m=0}^{d}h_{j-1}^{(m,t')}+(1-\beta)\overline{C}_{MOT}^{(d,t')}, \tag{8}$$

where β is a forgetting factor that is also set to 0.5 in our experiments. It is also worth mentioning that for the lowest temporal layer, which can include I or P pictures, these average complexities are reset (that is, α and β are temporary set to 1) when the current type of picture is different from the previous one at the same temporal layer, so that potential complexity mismatches due intrinsic encoding differences between I and P pictures are prevented.

4) Estimation of AU Target Bits: In order for the substream associated with the d^{th} dependency layer to satisfy the target bit rate constraint $R^{(d)}$, the amount of AU output bits should be controlled according to a bit budget $G^{(d,t')}$, which is determined by the following model:

$$G^{(d,t')}=G_{NOM}^{(d)}+\Delta G_{TEX}^{(d,t')}+\Delta G_{MOT}^{(d,t')}, \tag{9}$$

where $G_{NOM}^{(d)}$ is the nominal bit budget:

$$G_{NOM}^{(d)}=\frac{R^{(d)}}{f_{out}^{(d)}}, \tag{10}$$

and $\Delta G_{TEX}^{(d,t')}$ and $\Delta G_{MOT}^{(d,t')}$ represent the bit increments that depend on the relative texture and motion complexities among temporal layers, respectively, i.e.:

$$\Delta G_{TEX}^{(d,t')}=\frac{R^{(d)}}{f_{out}^{(d)}}\left(\frac{\overline{C}_{TEX}^{(d,t')}\sum_{u=0}^{T^{(d)}-1}N^{(d,u)}}{\sum_{u=0}^{T^{(d)}-1}\left(\overline{C}_{TEX}^{(d,u)}N^{(d,u)}\right)}-1\right), \tag{11}$$

$$\Delta G_{MOT}^{(d,t')}=\overline{C}_{MOT}^{(d,t')}-\frac{\overline{C}_{TEX}^{(d,t')}\sum_{u=0}^{T^{(d)}-1}\left(\overline{C}_{MOT}^{(d,u)}N^{(d,u)}\right)}{\sum_{u=0}^{T^{(d)}-1}\left(\overline{C}_{TEX}^{(d,u)}N^{(d,u)}\right)}, \tag{12}$$

with $N^{(d,u)}$ being the total number of pictures per GOP with layer identifier (d,u).

5) $nV^{(d)}$ and $nAU^{(d)}$ Updating Equations: After encoding the $(j-1)^{th}$ picture with layer identifier (d,t'), the parameters required to estimate the incremental variation of QP for the next picture are finally updated by means of the following expressions:

$$nV^{(d)} = \max\left[0, \min\left[\frac{V^{(d)}}{BS^{(d)}}, 1\right]\right], \tag{13}$$

$$nAU^{(d)} = \max\left[\frac{1}{2}, \min\left[\frac{AU^{(d,t')}}{G^{(d,t')}}, 2\right]\right]. \tag{14}$$

Since these parameters bear the current state of the encoding process in terms of buffer occupancy and target bit rate mismatch, the most appropriate QP variation should be derived from them. For instance, if $nV^{(d)}$ were close to 1 (overflow risk) and $nAU^{(d)}$ were close to 2 (large bit rate mismatch), then the QP increment would be high in order to quickly correct such mismatches. On the other hand, if $nV^{(d)}$ were close to 1 but $nAU^{(d)}$ were also close to 1, then the QP increment would not be high, so that the visual quality is maintained. Nevertheless, it is not easy to infer practical decision-making rules from particular examples such as the previous ones. Instead, this task has been addressed through a carefully designed QP increment estimation process that is described in the following subsection.

C. RBF-based QP Increment Estimation

This subsection discusses the reasons behind the features selected as components of the input vector to the RBF network and describes the proposed method to estimate the QP increment for the j^{th} picture.

1) Selection of the Input Vector to the RBF Network: There are many parameters that can potentially influence the selection of a proper QP increment value, such as measures of actual buffer fullness and AU output bits, target buffer fullness, buffer size, reference QP value, video content properties, GOP size, dependency and temporal layer identifiers, etc. In order to reach a good compromise between performance and computational cost, in this work we have selected four parameters: $nV^{(d)}$, $nAU^{(d)}$, nTF, and BD. The reasons for selecting these ones and rejecting others are given next.

The normalized versions of both buffer fullness $nV^{(d)}$ and AU output bits $nAU^{(d)}$ have to be considered in order to guarantee long-term average bit rate adaptation while maintaining the buffer occupancy at secure levels. In fact, similar parameters to these ones have been already successfully used in previous works on the same subject, as those described in [16].

The normalized target buffer fullness nTF is used by the rate controller to lead the buffer occupancy toward that reference point. Although in VBR scenarios it is common to operate with target buffer fullness values between 40% and 60% of buffer size, we decided to consider this parameter because its influence on the selection of $\Delta QP^{(d)}$ becomes crucial when it takes either lower or higher values since the risk of underflow or overflow, respectively, increases dramatically and must be controlled.

The buffer size BD is related to the region of the R-D space where the rate controller can operate; in other words, it determines the operating point between the constant-rate region (small buffer size) and the constant-quality region (large buffer size). Thus, the larger the buffer size, the smoother the QP variation should be so that the visual quality consistency is high.

On the other hand, the temporal layer identifier has been taken into account in an alternative manner that will be described in detail below. In particular, two different RBF networks were trained, one for the lowest temporal layer, and the other for the enhancement temporal layers.

Other parameters were considered and discarded for the sake of the *performance-complexity* tradeoff, in particular: reference QP value, video complexity measures, GOP size, and dependency layer identifier. Although all of these parameters have an undeniable influence on the selection of the QP increment, their contribution does not turn out to be essential in a VBR scenario where a long-term average bit rate adaptation is sufficient. On the other hand, if they were considered, both the complexity of the RBF network training process and the operation complexity would considerably increase due to the increment of the input vector dimension.

2) QP Increment Estimation: As previously stated, the proposed $\Delta QP^{(d)}$ estimation method operates on the following input vector:

$$\mathbf{X}^{(d)} = \left(nV^{(d)}, nAU^{(d)}, nTF, BD\right)^T, \tag{15}$$

implicitly assuming that all the virtual buffers share the same nTF and BD values.

A carefully designed RBF network is used to estimate $\Delta QP^{(d)}$ from the input vector $\mathbf{X}^{(d)}$. The RBF-based estimation obeys:

$$\Delta QP^{(d)} = \text{round}\left[w_0 + \sum_{i=1}^{L} w_i H_i\left(\mathbf{X}^{(d)}\right)\right], \tag{16}$$

where L is the number of basis functions $\left\{H_i\left(\mathbf{X}^{(d)}\right)\right\}_{i=1,\ldots,L}$ of the hidden layer, w_i the output weights, and w_0 the bias. It should be noted that the output of the RBF network is converted into an integer, given the discrete nature of the quantization parameter in H.264/SVC. The basis functions are Gaussian-type functions with centers $\mathbf{C}_i$ and widths $\mathbf{\Sigma}$, that is:

$$H_i\left(\mathbf{X}^{(d)}\right) = \exp\left(-\sum_{j=1}^{4} \frac{\left(X_j^{(d)} - C_{ij}\right)^2}{\Sigma_j^2}\right). \tag{17}$$

The Gaussian-type functions are the most common ones and, as shown later on, have provided good results in our experiments.

As it will be explained in detail in Section III, the training of the RBF network relies on a training data set containing pairs *input vector-desired output*, which have to be previously generated. Once these training data were generated, it was observed that the data distributions for the lowest temporal layer and the higher temporal layers were different enough to

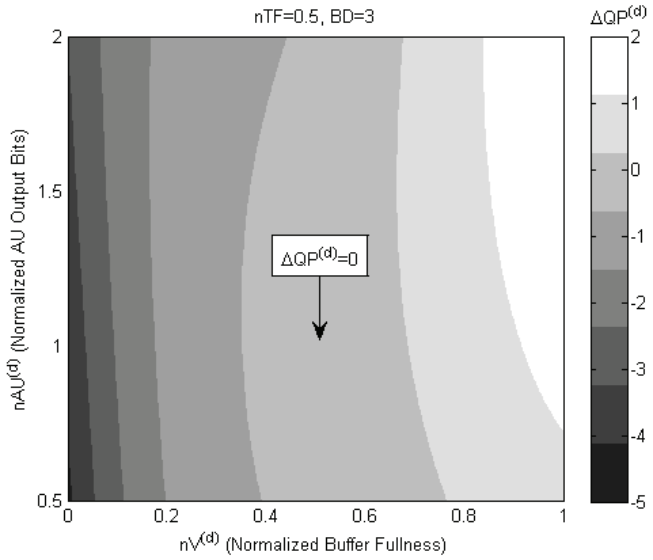

Fig. 3. Output of the key-picture RBF network for $nTF=0.5$ and $BD=3$.

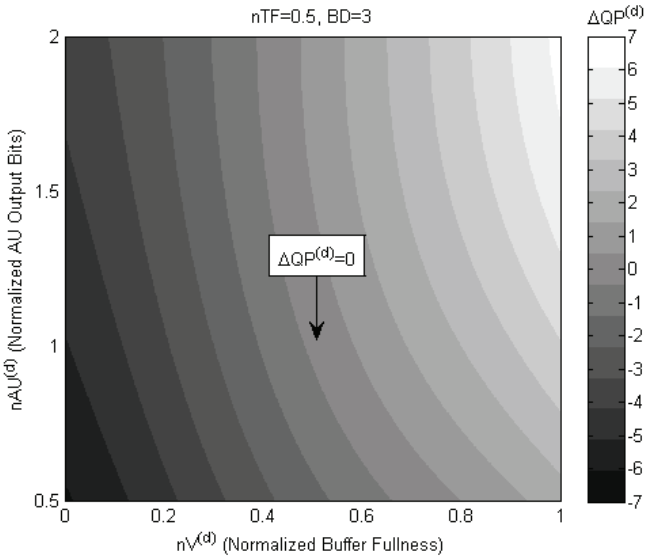

Fig. 5. Output of the non-key-picture RBF network for nTF=0.5 and BD=3.

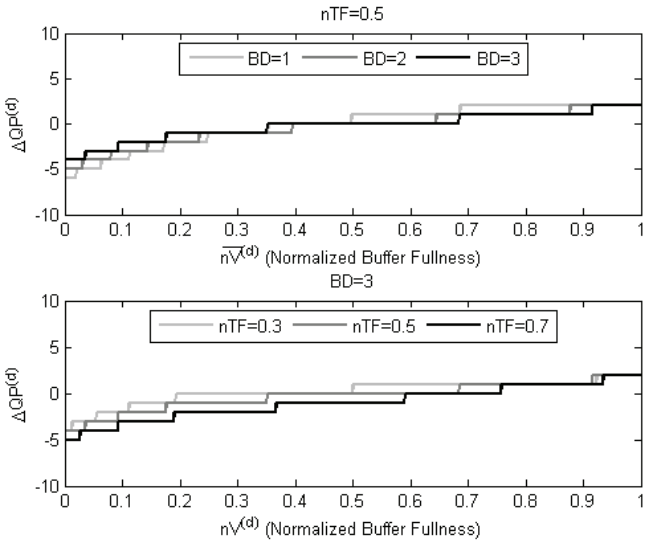

Fig. 4. Sample outputs of the key-picture RBF network for $nTF=0.5$ and several values of BD (Top), and for $BD=3$ and several values of nTF (Bottom). For the sake of clarity, only a cut of the three-dimensional surface for $nAU^{(d)}=1$ is drawn.

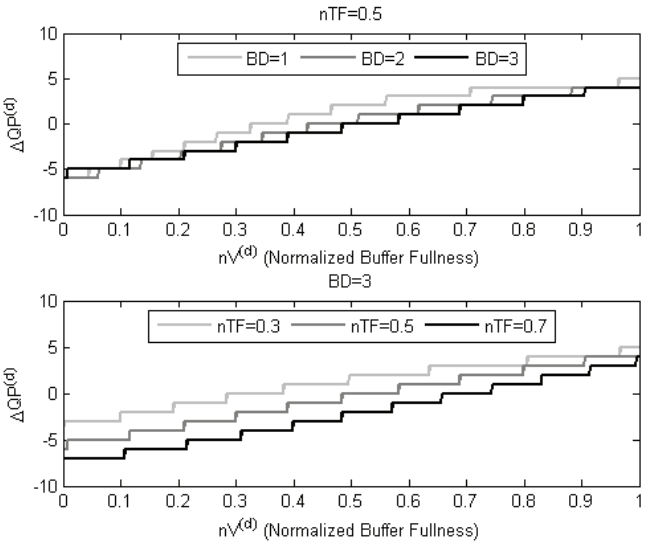

Fig. 6. Sample outputs of the non-key-picture RBF network for $nTF=0.5$ and several values of BD (Top), and for $BD=3$ and several values of nTF (Bottom). For the sake of clarity, only a cut of the three-dimensional surface for $nAU^{(d)}=1$ is drawn.

justify the design of two specific RBF networks. There were two alternatives for classifying the temporal layers into two subsets depending on in which subset the layer immediately higher than the lowest layer is considered. We decided to design one RBF network for key pictures (the lowest temporal layer) and the other for non-key pictures given the notable influence of the key picture quality on the global quality. Both QP increment models are named *key-picture* and *non-key-picture* RBF networks to emphasize that dependence on the frame type.

Furthermore, some experiments were performed to properly dimension the RBF networks. The results led us to select 7 Gaussian-type functions in both cases. It should be said that similar results were obtained for any higher number of RBFs.

The output of both the key-picture and non-key-picture RBF networks are illustrated in Figs. 3 and 5, respectively, for $nTF=0.5$ and $BD=3$. Since the input parameters nTF and BD are set before starting the encoding process, the proposed estimation function can be seen as a surface whose shape depends on these constants. Several outputs are also depicted in Fig. 4, for the key-picture RBF network, and Fig. 6, for the non-key-picture RBF network, for different target buffer levels and buffer sizes. In these cases only a cut of the three-dimensional surface for $nAU^{(d)}=1$ is depicted for clarity reasons.

Once the system was implemented, some unnecessary fluctuations of the QP value at non-key pictures were observed, especially in cases of stationary video complexity when the

buffer level approached the target buffer fullness. The problem was related to the estimation of $nAU^{(d)}$, which is normalized by a bit budget that is computed from estimated video complexities. The estimation errors in the complexities cause random short-term variations in $nAU^{(d)}$ that, in turn, produce short-term QP fluctuations in non-key pictures since the output of the corresponding RBF network exhibits small step sizes $\Delta QP^{(d)}$ (see Figs. 5 and 6). The proposed modeling for QP increment estimation can not correct such fluctuations since the QP time evolution is not considered; in other words, the non-key-picture RBF network is not aware of the QP time evolution because the QP increment at the j^{th} time instant is estimated just from the input vector at the previous time instant. In order to solve this drawback, three solutions were studied. The first consisted of enlarging the input vector to span a couple of time instants; however, the associated computational cost turned out to be unacceptable. The second consisted of filtering $nAU^{(d)}$ to smooth its noisy instantaneous fluctuations [16], but the coding results were not satisfactory, especially at scene changes. The final solution consisted of expanding the input region $\left(nV^{(d)}, nAU^{(d)}\right)$ for which the output is $\Delta QP^{(d)} = 0$. To this end, a simple post-processing stage of the output of the non-key picture RBF network is proposed, that obeys:

$$\Delta QP^{(d)} = \begin{cases} -1 & \text{if } \Delta QP^{(d)} = -2 \\ 0 & \text{if } \Delta QP^{(d)} = -1 \\ 0 & \text{if } \Delta QP^{(d)} = 1 \\ 1 & \text{if } \Delta QP^{(d)} = 2. \end{cases} \quad (18)$$

This solution is used in every non-key picture and provides a good tradeoff between the performance in stationary video complexity and that achieved in time-varying situations.

D. Implementation Considerations

Although the complexity of the RC algorithm is negligible when compared to that of the encoding process as a whole, it deserves a brief comment. The RBF-based estimation of the QP increment can be seen as a parametric two-dimensional function, where the parameters are nTF and BD, and the inputs are $nV^{(d)}$ and $nAU^{(d)}$. Furthermore, since the QP increment is quantized, the output of this two-dimensional function is discrete. Therefore, if the two input variables are also quantized the function can be readily implemented as a look-up table. In summary, a look-up table can be used to implement the RBF-based estimation of the QP increment. A different look-up table should be used for each pair of parameter values (nTF, BD).

III. RBF Network Design

In order to find the most suitable RBF network parameters for both key and non-key pictures, training and validation processes were performed. Such processes are described in the following subsections.

A. Generation of the Training Data Set

A training data set consisting of pairs:

$$\left\{\mathbf{X}^{(d)}, \Delta QP^{*(d)}\right\}, \quad (19)$$

where $\mathbf{X}^{(d)}$ is the input feature vector defined in Eq. (15) and $\Delta QP^{*(d)}$ is the desired output QP increment, should be generated in order to properly train an RBF network for our purposes. The generation of these training pairs is actually a key step in the success of the proposed approach. This subsection is devoted to describe this process.

The training data set was extracted from a representative set of video sequences exhibiting a large variety of spatio-temporal contents, so that the trained RBF networks could work properly for any type of input sequence. This set of video sequences used for training consisted on two parts:

- Some of the well-known sequences commonly used in the field; specifically: "Akiyo", "City", "Container", "Crew", "Hall", "Highway", "Ice", "News", "Paris", "Silent", "Soccer", and "Tempete". We used 300 pictures per sequence and some of them were upsampled and/or downsampled in order to get common intermediate format (CIF), quarter CIF (QCIF) and 4×CIF (4CIF) resolutions.
- Some sequences extracted from high-quality digital video discs (DVD). In this case, we used 900 pictures per sequence that were downsampled to get QCIF and CIF resolutions from standard definition (SD).

Furthermore, none of these training sequences was used in the performance assessment of the proposed VBR controller conducted in Section IV.

For each training sequence, a reduced number of consecutive GOP pairs were selected along the sequence. The first GOP of each pair was used to initialize the average texture and motion complexities (a complete GOP is needed because initial average texture and motion complexities are required for each spatio-temporal layer). The second GOP was used to actually extract training data pairs $\left\{\mathbf{X}^{(d)}, \Delta QP^{*(d)}\right\}$. In order to obtain training samples for a variety of scenarios, each GOP pair was encoded using K different configurations. These K different configurations involved several encoder- and RC-related parameters: number of dependency layers, spatial resolutions, GOP size, target bit rate, target buffer level, and buffer size.

1) Getting Initial Average Complexities: Given an encoding configuration k, a baseline QP, denoted as $QP_{R_k}^{(d)}$, was chosen for each dependency layer d so that the corresponding target bit rate for the whole sequence $R_k^{(d)}$ would be generated. Then, the first GOP of each GOP pair was encoded P times, each one using a different QP increment with respect to $QP_{R_k}^{(d)}$, i.e., $\left\{QP_{R_k}^{(d)} + \Delta QP_p^{(d)}\right\}_{p=1,..,P}$, and the computed average texture and motion complexities for each QP increment were stored as initial complexities for the subsequent process. Specifically, in our experiments the number of encodings for a given baseline QP was $P = 10$, using QP increments from -5 to 5.

2) Generating Training Pairs: As previously mentioned, once the initial average texture and motion complexities had been obtained for every layer, the second GOP was used to extract the training pairs. For each picture j of the second GOP, the aim was to determine the optimum QP increment for a wide range of potential conditions concerning the buffer occupancy and the adjustment to the AU target bits. In order

to achieve this variety of encoding conditions, the multiple encoding process initiated for the first GOP continued along the second GOP for the same set of P quantization values. As a result, before encoding the j^{th} picture, all the previous pictures had been encoded P times, so that a set of P input vectors would be available:

$$\mathbf{X}_{j,k,p}^{(d)} = \left(nV_{j,k,p}^{(d)}, nAU_{j,k,p}^{(d)}, nTF_k, BD_k \right)^T,$$

where variables $nV_{j,k,p}^{(d)}$ and $nAU_{j,k,p}^{(d)}$ summarize the encoding state after the $(j-1)^{th}$ picture. Then the challenge was to find the optimum $\Delta QP^{*(d)}$ for each one of the P possible input vectors, which represent a variety of encoding conditions. To this end, a second set of Q quantization increments $\left\{ \Delta QP_q^{(d)} \right\}_{q=1,\ldots,Q}$ with respect to $\left\{ QP_{R_k}^{(d)} + \Delta QP_p^{(d)} \right\}$ was used to encode the j^{th} picture. Particularly, in our experiments a total of $Q=23$ quantization increments from -11 to 11 were used to find the optimum $\Delta QP^{*(d)}$.

Finally, for each input vector $\mathbf{X}_{j,k,p}^{(d)}$, the QP increment $\Delta QP_q^{(d)}$ that minimized certain cost function Ψ was chosen as the optimum one:

$$\Delta QP^{*(d)} = \underset{\Delta QP_q^{(d)}}{\operatorname{argmin}} \Psi\left(\Delta QP_q^{(d)} \right). \tag{20}$$

The cost function has been designed ad hoc for this problem aiming at properly balance several conflicting factors: quality consistency, buffer control, and QP consistency. Specifically Ψ adopts the following form:

$$\Psi\left(\Delta QP_q^{(d)} \right) = \lambda_1 \theta \left(\frac{D_j^{(d)} - \overline{D}^{(d)}}{255} \right)^2 + \lambda_2 \left(\frac{V_{j+1}^{(d)}}{BD_k \times R_k^{(d)}} - nTF_k \right)^2 + \lambda_3 \left(\frac{\Delta QP_q^{(d)}}{\Delta QP_{MAX}^{(d)}} \right)^2. \tag{21}$$

The first term monitors the quality consistency by means of the squared normalized difference between the distortion $D_j^{(d)}$ of the current picture and the average distortion $\overline{D}^{(d)}$ of all the previously encoded pictures. The *mean of the absolute error* between the original and reconstructed luminance pictures was used as distortion metric.

The second term considers the buffer control through the squared difference between the normalized current buffer level $V_{j+1}^{(d)} / BD_k \times R_k^{(d)}$ and normalized target buffer fullness nTF_k.

The third term watches over the QP consistency by means of the squared ratio of the considered ΔQP and the maximum allowed QP deviation $\Delta QP_{MAX}^{(d)}$, which was set to 11 QP units in our experiments. The motivation for this third term comes from the fact that, in same cases, due to the high coding efficiency of SVC at high spatio-temporal layers, several QP increments yield quite similar distortion and number of output bits because of the low energy of the AC transformed coefficients.

The weight vector $(\lambda_1, \lambda_2, \lambda_3)^T$ was selected by means of a validation process (described in the next subsection) to achieve the best tradeoff among the three terms of the cost function. In order to obtain more meaningful values for the weights, the first term of the cost function was scaled by introducing an additional factor θ such that its dynamic range would be similar to those of the second and third terms. In particular, θ was set to 100 in our experiments. Finally, as we are only interested in the relative weights, the three weights are made to sum up to one.

Before starting out the network training, a set of possible weight vectors for the cost function was pre-established by considering different tradeoffs among quality consistency, buffer control, and QP consistency. Subsequently, several sets of training data were generated per dependency layer following the method previously described. Additionally, a reduced set of values for both the normalized target buffer fullness nTF and buffer size BD were selected, so that a wide range of VBR applications would be covered; specifically, nTF and BD were sampled in the following ranges: $0.1 \leq nTF \leq 0.9$ and $1 \leq BD \leq 3$.

For any of the pre-established cost function weight vectors, the following conclusions were drawn from the training data distributions:

1) Figs. 7 and 8 show superimposed training data distributions for both key and non-key pictures. Each figure was obtained for a different weight vector: Fig. 7 comes from the weight vector selected for key pictures (see next subsection), while Fig. 8 uses the weight vector selected for non-key pictures. As can be observed, in any case the data distributions were different enough to justify the design of two specific RBF networks.
2) As shown in Figs. 9 and 10, the training data distributions for each dependency layer were similar enough to each other to justify the use of the same RBF network for all the layers considered. Fig. 9 shows the data for key pictures and the corresponding weight vector, while Fig. 10 focuses on non-key pictures.

B. RBF Network Training and Parameter Selection

For each pre-established weight vector, two training data sets, one for key pictures and the other for non-key pictures, were generated. Each RBF network was trained several times considering each one of the pre-established weight vectors, different random initializations, and different numbers L of radial basis functions. For this purpose, a training algorithm based on Gaussian processes (GP) [40] was used because it provides a robust solution for the network parameters that relies on maximizing a marginal likelihood. In particular, a Matlab toolbox due to Snelson and Gharahmani [41] available in [42] was used. This toolbox implements a sparse approximation to GP regression to reduce the training process complexity.

In order to select the best weight vector and the best L value, the resulting RBF networks were experimentally assessed for different RC configurations by encoding several video sequences belonging to the training set. First, the weight vector that provided the best quality consistency without incurring in buffer overflows and underflows was selected. The results for both key-picture and non-key-picture RBF networks are given in Table II. Second, once the best weight vector had been fixed, the lowest L value that properly fitted the data was selected

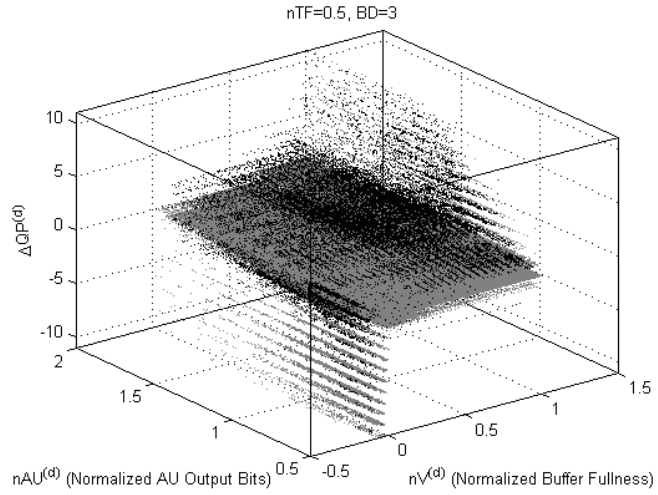

Fig. 7. Training data distributions for key pictures (black) and non-key pictures (gray), with $nTF = 0.5$ and $BD = 3$. The weight vector in Eq. (21) used for generating these distributions was: $\lambda_1 = 0.90$, $\lambda_2 = 0.09$, $\lambda_3 = 0.01$. A high-quality plot is available on-line in [43].

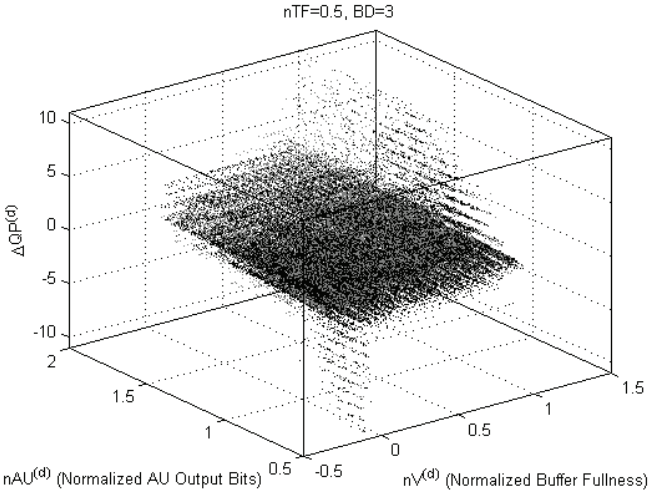

Fig. 9. Key-picture training data distributions for the base layer (black) and the enhancement layers (gray), with $nTF = 0.5$ and $BD = 3$. The weight vector in Eq. (21) used for generating these distributions was: $\lambda_1 = 0.90$, $\lambda_2 = 0.09$, $\lambda_3 = 0.01$. A high-quality plot is available on-line in [43].

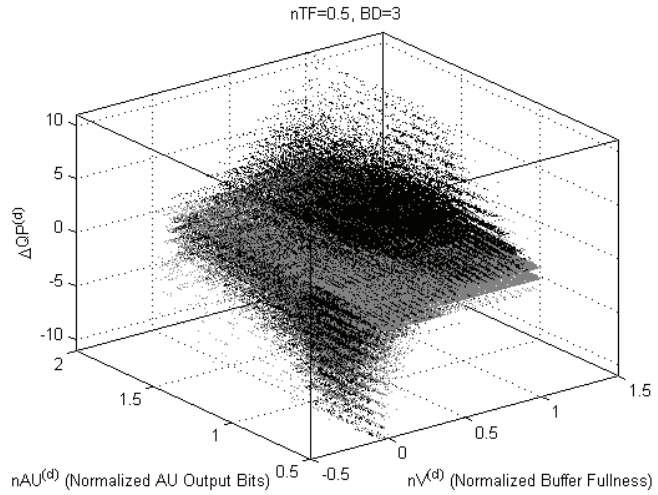

Fig. 8. Training data distributions for key pictures (black) and non-key pictures (gray), with $nTF = 0.5$ and $BD = 3$. The weight vector in Eq. (21) used for generating these distributions was: $\lambda_1 = 0.75$, $\lambda_2 = 0.24$, $\lambda_3 = 0.01$. A high-quality plot is available on-line in [43].

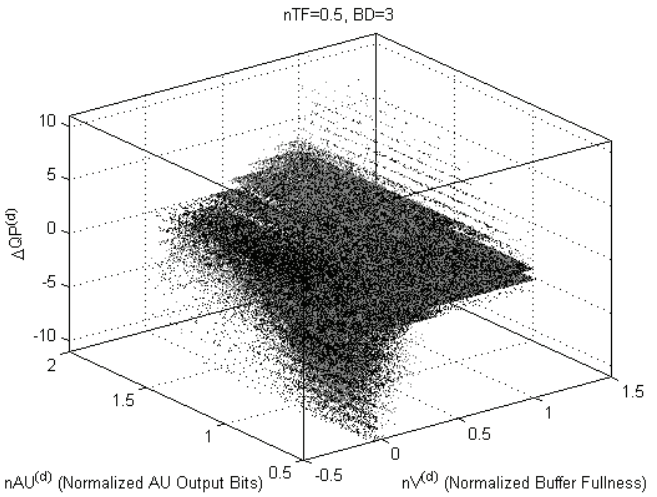

Fig. 10. Non-key-picture training data distributions for the base layer (black) and the enhancement layers (gray), with $nTF = 0.5$ and $BD = 3$. The weight vector in Eq. (21) used for generating these distributions was: $\lambda_1 = 0.75$, $\lambda_2 = 0.24$, $\lambda_3 = 0.01$. A high-quality plot is available on-line in [43].

to be $L = 7$. The resulting RBF network parameters for both key and non-key pictures are given in Appendix A.

IV. Experiments and Results

The Joint Scalable Video Model (JSVM) H.264/SVC reference software version JSVM 9.16 [44] was used to implement the proposed VBR controller. In order to assess its performance, our proposal was compared to two methods: 1) constant QP (CQP) encoding[1], which can be seen as an unconstrained VBR controller [1], was used as a reference for nearly constant quality video; and 2) the frame level CBR control algorithm described in [28].

[1]Constant QP encoding means that every temporal layer within a spatial/CGS layer shares the same QP value, while the QP value of each spatial/CGS layer can be different in order to reach the pre-established target bit rate $R^{(d)}$.

Following the recommendations for SVC testing conditions described in [45], both the H.264/SVC encoder and the proposed RC algorithm were configured to simulate on a personal computer two real-time application scenarios: mobile live streaming and IPTV broadcast. In the following subsections, both the SVC and RC configurations for each of the proposed testing scenarios are described, and then the experimental results are shown and discussed.

A. Description of the Application Scenarios

1) Mobile Live Streaming: A brief description of the SVC encoder configuration for mobile live streaming is given in the following paragraphs. For a more detailed explanation of this application the reader is referred to [24].

A high-quality scalable bit stream that consists of a base layer and a set of enhancement layers is made available

TABLE II
SELECTED WEIGHT VECTORS FOR THE COST FUNCTION IN EQ. (21).

	λ_1	λ_2	λ_3
Key Picture	0.90	0.09	0.01
Non-Key Picture	0.75	0.24	0.01

by a service provider. A mobile terminal, which can be a multimedia phone, PDA or laptop, accesses that scalable bit stream through a wireless network and decodes the sub-stream that complies with the arranged QoS. Particularly, starting out with the design suggested in [45] as reference, the following spatial/CGS encoding and RC configuration was used:

- Number of pictures: 900.
- GOP size/Intra period: 8/32 pictures.
- GOP structure: hierarchical B pictures.
- Search range for motion estimation: 16×16 pixels.
- Number of dependency layers: $D=5$
  - $d=0$: QCIF, $f_{out}^{(0)}=6.25$ Hz $\left(T^{(0)}=2\right)$
  - $d=1$: QCIF, $f_{out}^{(1)}=12.5$ Hz $\left(T^{(1)}=3\right)$
  - $d=2$: CIF, $f_{out}^{(2)}=12.5$ Hz $\left(T^{(2)}=3\right)$
  - $d=3$: CIF, $f_{out}^{(3)}=12.5$ Hz $\left(T^{(3)}=3\right)$
  - $d=4$: CIF, $f_{out}^{(4)}=25$ Hz $\left(T^{(4)}=4\right)$
- Symbol mode: CAVLC at every dependency layer (as suggested in [25]).
- Rate control parameters
  - Target buffer fullness: $nTF=50\%$.
  - Buffer size: $BD=3$ s.

Two sets of video sequences at 25 Hz exhibiting a variety of complexities were used in our experiments. The first set consisted of four well-known test sequences recommended in [45] for streaming applications: "Bus", "Football", "Foreman" and "Mobile". These sequences were concatenated to themselves several times to reach the aforementioned number of pictures. The second set consisted of three sequences displaying scene changes: "Soccer-Mobile-Foreman", "Spiderman" (movie), and "The Lord of the Rings" (movie). "Soccer-Mobile-Foreman" was formed by concatenating 300 frames of each sequence. The other two were extracted from high-quality DVDs and downsampled to either QCIF or CIF format, and have been made available on-line in [43]. They show many scene cuts, so they are challenging from the RC point of view.

All the sequences were encoded using the set of QP values that best approached some pre-established target bit rates. For the first group of sequences the target bit rates were those suggested in [45] for the spatial/CGS testing scenario. For the second group, the following medium-quality target bit rates were selected: 64 ($d=0$), 96 ($d=1$), 192 ($d=2$), 384 ($d=3$) and 512 kbps ($d=4$). In all cases, the exact output bit rates obtained by CQP encoding were used as target bit rates $R^{(d)}$ for both the RC algorithm in [28] and the proposed VBR controller.

2) IPTV Broadcast: TV broadcast through IP networks involving heterogeneous terminals (resolutions) is one of the natural fields of application for scalable video coding [25]. According to both the IP network characteristics and the target IPTV set-top box definition, a wide variety of scenarios can be specified. Nevertheless, in order to define the IPTV broadcast scenario used in this paper, we only took into consideration the display resolution and computational capabilities of the receiving devices, regardless the actual underlying type of IP network (fixed or mobile access, managed or unmanaged core). In particular, SD and high definition (HD) TV were selected as target resolutions (emphasizing the difference with respect to those employed for the mobile live streaming scenario) for the following spatial/CGS encoding and RC configuration:

- Number of pictures: 500/600.
- GOP size/Intra period: 16/16 pictures.
- GOP structure: hierarchical B pictures.
- Search range for motion estimation: 32×32 pixels.
- Number of dependency layers: $D=4$
  - $d=0$: SDTV, $f_{out}^{(0)}=25/30$ Hz $\left(T^{(0)}=4\right)$
  - $d=1$: SDTV, $f_{out}^{(1)}=25/30$ Hz $\left(T^{(1)}=4\right)$
  - $d=2$: HDTV (720p), $f_{out}^{(2)}=50/60$ Hz $\left(T^{(2)}=5\right)$
  - $d=3$: HDTV (720p), $f_{out}^{(3)}=50/60$ Hz $\left(T^{(3)}=5\right)$
- Symbol mode: CABAC at every dependency layer.
- Rate control parameters:
  - Target buffer fullness: $nTF=40\%$.
  - Buffer size: $BD=1.5$ s.

The following set of HDTV test video sequences of duration 10 s, which are available on-line in [46], were used in our experiments: "Mobcal_720p50", "Parkrun_720p50", "Shields_720p50" and "Stockholm_720p60". They were downsampled to obtain the corresponding SDTV versions.

The criterion used to select the target bit rate for each dependency layer was that recommended in [45] for the testing scenario. The criterion suggests doubling the rate starting from the lowest until reaching the highest for each spatial resolution, and increasing the minimum rate by a factor of four between consecutive spatial resolutions. Thus, the following target bit rates were proposed to cover the medium-quality range: 1024 ($d=0$), 2048 ($d=1$), 4096 ($d=2$) and 8192 kbps ($d=3$).

Similarly to the mobile live streaming application, the set of QP values that best approached the target bit rates was found, and the actual output bit rates were used as target bit rates for the two RC algorithms.

B. Experimental Results and Discussion

In order to assess the performance of the proposed VBR controller from a quality point of view, the average luminance PSNR μ_{PSNR} was used. The Bjøntegaard recommendation [47] was followed to compute PSNR differences with respect to CQP encoding. The average results over all the test video sequences in terms of PSNR increments $\Delta\mu_{PSNR}$ are summarized in Tables III and IV for mobile live streaming and IPTV broadcast scenarios, respectively. Two rows per spatial/CGS layer are shown, one for [28] and another for the proposed method. As can be observed, the performance achieved by the proposed method in terms of average PSNR was similar to that of CQP encoding, and notably superior to that of [28]. Furthermore, the good results achieved by the proposed method at layers 2 and 3 in the IPTV broadcast scenario (Table IV) deserve a special comment. These layers correspond to HD sequences and no samples of HD sequences

TABLE III
AVERAGE RESULTS ACHIEVED BY BOTH THE RC ALGORITHM IN [28] AND THE PROPOSED VBR CONTROLLER FOR THE MOBILE LIVE STREAMING SCENARIO. INCREMENTAL RESULTS ARE GIVEN WITH RESPECT TO CONSTANT QP ENCODING.

d	Algorithm	$\Delta\mu_{\mathrm{PSNR}}$ (dB)	$\Delta\overline{\sigma}_{\mathrm{PSNR,j}}$ (dB)	Bit Rate Error (%)	#O/#U	μ_{V} (%)
0	[28]	-0.19	0.41	1.87	8/0	57.42
	Proposed	-0,13	0.12	0.93	0/0	52.45
1	[28]	-0.43	0.75	1.35	9/0	57.29
	Proposed	-0,14	0.14	1.25	0/0	59.30
2	[28]	-0.33	0.35	0.68	6/0	54.91
	Proposed	-0,10	0.05	0.87	0/0	53.41
3	[28]	-0.20	0.36	0.44	0/0	52.81
	Proposed	-0,07	0.05	0.69	0/0	52.81
4	[28]	-0.46	0.51	0.30	0/0	53.45
	Proposed	-0,07	0.06	0.90	0/0	57.29

TABLE IV
AVERAGE RESULTS ACHIEVED BY THE RC ALGORITHM IN [28] AND THE PROPOSED VBR CONTROLLER FOR THE IPTV BROADCAST SCENARIO. INCREMENTAL RESULTS ARE GIVEN WITH RESPECT TO CONSTANT QP ENCODING.

d	Algorithm	$\Delta\mu_{\mathrm{PSNR}}$ (dB)	$\Delta\overline{\sigma}_{\mathrm{PSNR,j}}$ (dB)	Bit Rate Error (%)	#O/#U	μ_{V} (%)
0	[28]	-0.07	0.70	0.57	0/0	49.77
	Proposed	-0.11	0.31	1.86	0/0	38.16
1	[28]	-0.52	0.45	0.41	0/0	46.55
	Proposed	-0.15	0.26	1.99	0/0	35.80
2	[28]	-0.74	0.25	0.31	0/0	45.42
	Proposed	0.06	0.16	1.77	0/0	37.43
3	[28]	-0.40	0.20	0.14	0/0	44.16
	Proposed	0.06	0.20	1.43	0/0	35.14

were used for training. Therefore, these results prove that the RBF networks generalize properly and are able to work well for any resolution.

Tables V and VI show a detailed comparison of the three assessed algorithms for two representative video sequences. "The Lord of the Rings", taken from the mobile live streaming scenario, is a good example of non-stationary video complexity. On the other hand, "Stockholm", from the IPTV broadcast scenario, is an example of stationary video complexity. The analysis of these results allowed us to draw two main conclusions: 1) for non-stationary complexity sequences, the performance of the proposed method was remarkably good, exceeding even that of the nearly constant quality system at some dependency layers; and 2) for stationary complexity sequences, the performance of the proposed method was quite close to that of the nearly constant quality system.

Representative behaviors of the encoder buffer occupancy and the PSNR and QP time evolutions corresponding to the third enhancement layer ($d=3$) are depicted in Figs. 11 ("The Lord of the Rings") and 12 ("Stockholm"). When compared to [28], the proposed VBR controller made better use of the buffer to provide PSNR and QP time evolutions closer to those of the nearly constant quality system. Furthermore, in the non-stationary scenario, the strong correlation among buffer occupancy, PSNR time evolution, and QP time evolution reveals that the proposed method made a proper use of the buffer to successfully allocate larger amounts of bits for more complex scenes, and vice versa. Consequently, the potential quality fluctuation of the compressed video was kept low, in particular at scene changes (see, for example, the PSNR time evolution around pictures $\#260$ and $\#703$). It is also worth noting that the proposed method did an excellent work on minimizing unnecessary changes in QP time evolution, which is our main design goal; particularly, in the stationary scenario, it was able to provide a performance close to that of the nearly constant quality system. In terms of PSNR time evolution, the results were not so good for some sequences, such as that shown in Fig. 12. In these cases, the GOP-periodic PSNR leaps are due to large R-D differences between key and non-key pictures. As can be observed, this behavior also happens in CQP encoding whose performance we intend to meet.

In order to assess the proposed VBR control algorithm from the quality consistency point of view, a time-local version of the PSNR standard deviation was computed. This local PSNR standard deviation aims to measure the quality consistency within a scene, so reducing the impact of the scene changes on the PSNR standard deviation. Thus, small local PSNR standard deviations indicate smooth short-term PSNR fluctuations and therefore high quality consistency. In particular, the local PSNR standard deviation was computed over a time-window as follows:

$$\sigma_{PSNR,j}=\sqrt{\frac{1}{W}\sum_{i=j-W/2}^{j+W/2-1}\Big(PSNR_i-\mu_{PSNR,W}\Big)^2}, \qquad (22)$$

where W denotes the time-window size (in number of pictures) and $\mu_{PSNR,W}$ the average PSNR for a given window size. In particular, W was set to $2^{T^{(d)}}$ pictures in our experiments, which is a time interval short enough to minimize the influence of PSNR leaps at the scene changes. Finally, in order to summarize the results in an unique measurement, the mean value of the local PSNR standard deviation, denoted as $\overline{\sigma}_{PSNR,j}$, was computed.

Additionally, it should be noticed that, since the local PSNR standard deviation does not take into account any buffer constraint, CQP encoding provided a smaller local PSNR standard deviation (see Fig. 11). Obviously, this smaller local PSNR standard deviation was in exchange for high instantaneous bit rate variations at the scene changes that are not allowed in a constrained buffer scenario. The results in terms of $\overline{\sigma}_{PSNR,j}$ increment with respect to CQP encoding, $\Delta\overline{\sigma}_{PSNR,j}$, are provided in Tables III and IV. As can be observed, the proposed VBR controller achieved better quality consistency than that of the RC algorithm in [28]. Furthermore, the results, especially at higher spatial/CGS layers, were remarkably close to those of CQP encoding, in spite of the buffer constraint.

The proposed VBR controller was also assessed in terms of target bit rate adjustment and mean buffer level. In particular, its performance was comparatively evaluated by computing the output bit rate error, the number of pictures in which either an overflow ($\#$O) or an underflow ($\#$U) occurred, and the mean buffer level, μ_V. As can be observed in Tables III – VI, both the RC scheme in [28] and the proposed algorithm provided in most cases output bit rate differences below 2%,

TABLE V
PERFORMANCE COMPARISON BETWEEN THE RC ALGORITHM IN [28] AND THE PROPOSED VBR CONTROLLER FOR A SPECIFIC NON-STATIONARY COMPLEXITY VIDEO SEQUENCE, "THE LORD OF THE RINGS". THE RESULTS ACHIEVED BY CONSTANT QP ENCODING HAVE ALSO BEEN INCLUDED FOR REFERENCE. THE EXPERIMENTS WERE CONDUCTED USING THE CONFIGURATION OF THE MOBILE LIVE STREAMING SCENARIO FOR THE FOLLOWING TARGET BIT RATES: 66.47 (d = 0), 97.32 (d = 1), 189.47 (d = 2), 388.07 (d = 3) AND 500.56 kbps (d = 4).

d	Algorithm	μ_{PSNR} (dB)	$\overline{\sigma}_{\mathrm{PSNR,j}}$ (dB)	Bit Rate Error (%)	#O/#U	μ_{V} (%)
0	CQP	34.45	0.66	-	42/48	49.76
	[28]	33.14	1.10	3.82	55/0	78.04
	Proposed	34.35	0.90	1.57	0/0	53.46
1	CQP	34.39	0.67	-	100/107	46.90
	[28]	33.19	2.05	1.72	66/0	69.65
	Proposed	34.30	0.97	1.93	0/0	58.08
2	CQP	32.88	0.91	-	96/111	47.15
	[28]	32.26	1.51	0.30	40/0	63.69
	Proposed	32.80	1.09	1.93	0/0	52.19
3	CQP	35.24	0.82	-	92/114	45.22
	[28]	35.43	1.31	1.26	0/0	52.99
	Proposed	35.33	0.97	1.57	0/0	52.97
4	CQP	35.14	0.82	-	205/237	45.58
	[28]	34.86	1.57	1.00	0/0	53.82
	Proposed	35.23	0.98	2.43	0/0	63.35

TABLE VI
PERFORMANCE COMPARISON BETWEEN THE RC ALGORITHM IN [28] AND THE PROPOSED VBR CONTROLLER FOR A SPECIFIC STATIONARY COMPLEXITY VIDEO SEQUENCE, "STOCKHOLM". THE RESULTS ACHIEVED BY CONSTANT QP ENCODING HAVE ALSO BEEN INCLUDED FOR REFERENCE. THE EXPERIMENTS WERE CONDUCTED USING THE CONFIGURATION OF THE IPTV BROADCAST SCENARIO FOR THE FOLLOWING TARGET BIT RATES: 975.92 (d = 0), 1885.90 (d = 1), 4209.83 (d = 2) AND 7331.63 kbps (d = 3).

d	Algorithm	μ_{PSNR} (dB)	$\overline{\sigma}_{\mathrm{PSNR,j}}$ (dB)	Bit Rate Error (%)	#O/#U	μ_{V} (%)
0	CQP	35.54	0.20	-	0/0	50.31
	[28]	35.47	0.91	0.80	0/0	49.48
	Proposed	35,53	0.34	-1.64	0/0	37.20
1	CQP	38.60	0.14	-	0/0	50.55
	[28]	37.94	0.54	0.21	0/0	46.14
	Proposed	38,58	0.26	-1.89	0/0	35.36
2	CQP	34.18	0.18	-	0/0	43.30
	[28]	33.60	0.34	0.29	0/0	45.10
	Proposed	34,27	0.23	-1.88	0/0	35.59
3	CQP	34.93	0.25	-	0/0	40.71
	[28]	34.53	0.32	0.15	0/0	43.91
	Proposed	34.98	0.32	-1.17	0/0	33.95

which is the maximum bit rate error recommended in [45] for the spatial/CGS testing scenario. The average results in terms of μ_V achieved by the proposed method were close to the target buffer fullness, thus proving a good long-term adaptation to the target bit rate at each dependency layer. Furthermore, the results in terms of #O and #U revealed that the proposed VBR controller was able to significantly reduce both the overflow and underflow risks in sequences with scene changes, such as "The Lord of the Rings". The poor performance of the RC algorithm in [28] at scene changes was due to the lack of a specif mechanism to deal with such events. The use of a scene change detector would be helpful to improve its performance in such cases.

Finally, from the complexity point of view, the central processing unit (CPU) time consumed by the proposed VBR controller and the RC scheme in [28] were measured by means of a high-resolution performance counter. In order to minimize the measurement error caused by occasional multi-task operations, each sequence was encoded five times and the minimum CPU time was selected for the complexity analysis (nevertheless, it is worth mentioning that the variance of the measured CPU times was very small). The complexity results using an Intel Core2 Duo CPU E8400@3.0 GHz are given in Table VII for the mobile live streaming scenario and in Table VIII for the IPTV broadcast scenario. As can be observed, the RC algorithm in [28] consumed an average CPU time per AU of 239 μs for the mobile live streaming scenario and 2071 μs for the IPTV broadcast scenario, while the proposed VBR controller only consumed 26 μs and 33 μs, respectively. These differences in terms of complexity between both algorithms are mainly due to the R-D model employed by the CBR controller in [28]. This RC algorithm, which follows the usual approach in H.264/AVC [7], first estimates the frame complexity and subsequently the QP value. The QP value estimation relies on a linear regression that is computationally heavier than the proposed RBF networks. Furthermore, the complexity estimation requires performing simple operations on the whole picture, what explains the significant CPU time increment that happens in the IPTV broadcast scenario (which operates on larger pictures).

Furthermore, as previously described in Section II-D, the complexity of the RBF-based QP estimation can be reduced even more by means of a look-up table-based implementation. In particular, preliminary experiments using 10×8 $\left(nV^{(d)} \times nAU^{(d)}\right)$ look-up tables for QP increment estimation were conducted, achieving nearly equivalent results. Therefore, the proposed RBF networks can be successfully implemented using look-up tables.

V. CONCLUSIONS AND FURTHER WORK

In this paper a novel VBR controller for real-time H.264/SVC video coding applications has been proposed. The VBR controller aims to improve the quality consistency by preventing unnecessary QP fluctuations. The proper QP increment estimation at each dependency layer is computed by means of two RBF networks, one for key pictures and the other for non-key pictures, that are specially designed for this purpose. This approach offers the additional advantage of not using any analytic R-D model for QP estimation, so the chicken-and-egg dilemma for frame complexity estimation is no longer a concern. Furthermore, the input vector to the RBF-based QP increment model is enlarged with two additional constant parameters to provide an effective solution for a wide range of both target buffer fullness and buffer size.

Two real-time application scenarios were simulated to assess the performance of the VBR controller, which was compared to both constant QP encoding, as a reference for nearly constant quality, and a recently proposed CBR controller for SVC [28]. For stationary complexity sequences, the average quality achieved by the VBR controller was quite close to that of the nearly constant quality system, since the time

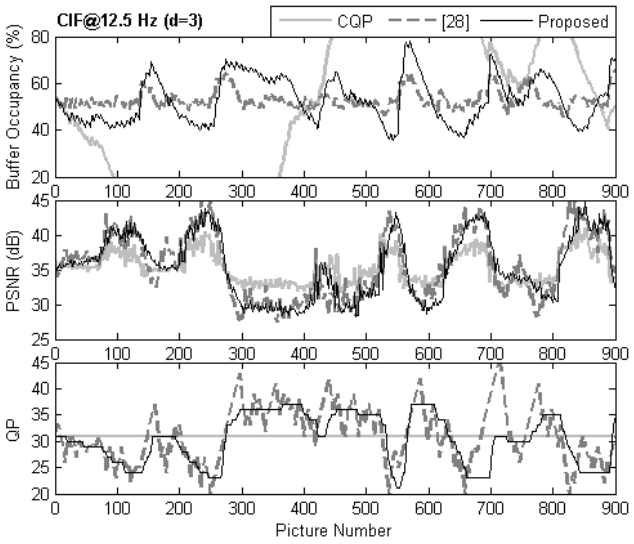

Fig. 11. Encoder buffer occupancy, PSNR and QP time evolutions corresponding to the third enhancement layer from "The Lord of the Rings". High-quality plots corresponding to every spatial/CGS layer are available on-line in [43].

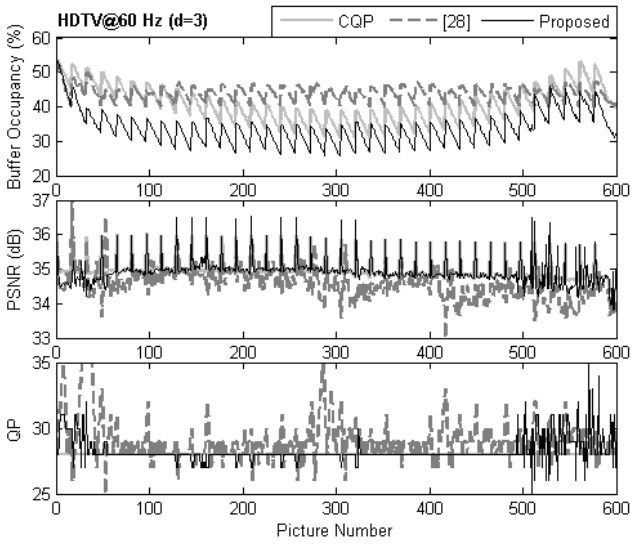

Fig. 12. Encoder buffer occupancy, PSNR and QP time evolutions corresponding to the third enhancement layer from "Stockholm". High-quality plots corresponding to every spatial/CGS layer are available on-line in [43].

TABLE VII
CPU TIME COMPARISON BETWEEN THE RC ALGORITHM IN [28] AND THE PROPOSED VBR CONTROLLER FOR THE MOBILE LIVE STREAMING SCENARIO USING AN INTEL CORE2 DUO CPU E8400@3.0 GHZ.

Sequence	**CPU Time (μs)**	
	[28]	**Proposed**
"Bus"	211355	23658
"Football"	221029	22555
"Foreman"	220253	23793
"Mobile"	209543	23149
Average	**215545**	**23289**
Average per *access unit*	**239**	**26**

TABLE VIII
CPU TIME COMPARISON BETWEEN THE RC ALGORITHM IN [28] AND THE PROPOSED VBR CONTROLLER FOR THE IPTV BROADCAST SCENARIO USING AN INTEL CORE2 DUO CPU E8400@3.0 GHZ.

Sequence	**CPU Time (μs)**	
	[28]	**Proposed**
"Mobcal"	1065523	16349
"Parkrun"	1038447	17061
"Shields"	1049664	16758
"Stockholm" (first 500 pictures)	988212	16550
Average	**1035462**	**16679**
Average per *access unit*	**2071**	**33**

evolution of QP was maintained almost constant in time. For non-stationary complexity sequences, the average quality of the proposed algorithm was remarkably good, exceeding even that of the nearly constant quality system at some dependency layers, since it was able to allocate larger amounts of bits for more complex scenes, and vice versa.

In terms of quality consistency, the performance of the proposed VBR controller was significantly better than that of the CBR algorithm in [28]. Furthermore, the experimental results, especially at higher spatial/CGS layers, were remarkably close to those of CQP encoding, in spite of the buffer constraint. With respect to the overflow and underflow risks, again the results revealed that the proposed VBR control algorithm was notably superior.

Finally, from the complexity point of view, the proposed method notably outperformed the RC scheme in [28].

To sum up, the proposed VBR controller achieved an excellent performance in terms of average quality, quality consistency, long-term adjustment to the target rate, and buffer overflow and underflow prevention at each spatial/CGS layer, with low complexity.

As future work, we plan to extend the VBR controller to MGS coding.

APPENDIX A
RBF PARAMETERS

The centers, widths and weights of the Gaussian-type functions used in our experiments for both key-picture and non-key-picture RBF networks are the following (also available on-line in electronic format in [43]):

1) Key-picture RBF parameters

$$w_0 = -1.94234, \qquad \mathbf{w} = \begin{pmatrix} 116.92009 \\ 45.00974 \\ 22.41989 \\ -14.39316 \\ -100.53808 \\ -57.14093 \\ -127.18792 \end{pmatrix},$$

$$\mathbf{C} = \begin{pmatrix} 0.34878 & 2.24208 & 0.32736 & 2.57098 \\ 0.64341 & 4.02300 & 0.56932 & -4.81181 \\ 0.75362 & 1.56418 & 0.47553 & 3.07934 \\ 0.72347 & -0.25308 & -0.10081 & -0.12420 \\ -0.99480 & -0.34192 & -1.39094 & 1.72556 \\ 0.06001 & 1.14999 & 3.47226 & -2.24075 \\ 0.40772 & 2.43468 & 0.39291 & 2.68413 \end{pmatrix},$$

$$\mathbf{\Sigma} = \begin{pmatrix} 0.68895 \\ 4.29915 \\ 2.31009 \\ 5.84732 \end{pmatrix}.$$

2) Non-key-picture RBF parameters

$$w_0 = -0.41095, \qquad \mathbf{w} = \begin{pmatrix} 1485.93883 \\ -206.80386 \\ -486.69837 \\ -1.91249 \\ -1366.10007 \\ 536.11049 \\ 33.63052 \end{pmatrix},$$

$$\mathbf{C} = \begin{pmatrix} 0.48170 & -0.18319 & 0.33508 & -0.20148 \\ 0.80986 & -0.12825 & 0.24415 & 0.45383 \\ 0.62855 & 0.77388 & 0.47196 & 2.75271 \\ 0.24348 & 1.16350 & 0.18820 & 2.71590 \\ 0.44971 & -0.22937 & 0.35083 & -0.19297 \\ 0.63746 & 0.66580 & 0.44850 & 2.63895 \\ 1.51031 & 1.34230 & 0.36623 & 1.02694 \end{pmatrix},$$

$$\mathbf{\Sigma} = \begin{pmatrix} 0.92423 \\ 3.38358 \\ 1.09690 \\ 3.75779 \end{pmatrix}.$$

REFERENCES

[1] T. Lakshman, A. Ortega, and A. Reibman, "VBR video: tradeoffs and potentials," *Proceedings of the IEEE*, vol. 86, no. 5, pp. 952–973, 1998.

[2] A. Ortega, "Variable bit-rate video coding," *in Compressed Video over Networks, M.-T. Sun and A. R. Reibman, Eds. New York: Marcel Dekker*, pp. 343–382, 2000.

[3] J. Ribas-Corbera, P. Chou, and S. 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.

[4] "Test Model 5 [Online], http://www.mpeg.org/MPEG/MSSG/tm5."

[5] T. Chiang and Y.-Q. Zhang, "A new rate control scheme using quadratic rate distortion model," in *Image Processing, 1996. Proceedings., International Conference on*, vol. 1, 1996, pp. 73–76 vol.2.

[6] J. Ribas-Corbera and S. Lei, "Rate control in DCT video coding for low-delay communications," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 9, no. 1, pp. 172–185, 1999.

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

[8] B. Tao, B. Dickinson, and H. Peterson, "Adaptive model-driven bit allocation for MPEG video coding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 10, no. 1, pp. 147–157, Feb 2000.

[9] N. Kamaci, Y. Altunbasak, and R. Mersereau, "Frame bit allocation for the H.264/AVC video coder via Cauchy-density-based rate and distortion models," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 15, no. 8, pp. 994–1006, 2005.

[10] S. Sanz-Rodriguez, O. del Ama-Esteban, M. de Frutos-Lopez, and F. Di andaz-de Maria, "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, 2010.

[11] S. Ma, W. Gao, and Y. Lu, "Rate-distortion analysis for H.264/AVC video coding and its application to rate control," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 15, no. 12, pp. 1533–1544, 2005.

[12] Z. He, Y. K. Kim, and S. Mitra, "Low-delay rate control for DCT video coding via ρ-domain source modeling," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 11, no. 8, pp. 928–940, 2001.

[13] Z. Chen and K. N. Ngan, "Towards rate-distortion tradeoff in real-time color video coding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 2, pp. 158 –167, feb. 2007.

[14] B. Xie and W. Zeng, "A sequence-based rate control framework for consistent quality real-time video," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 16, no. 1, pp. 56–71, 2006.

[15] N. Mohsenian, R. Rajagopalan, and C. A. Gonzales, "Single-pass constant- and variable-bit-rate MPEG-2 video compression," *IBM Journal of Research and Development*, vol. 43, no. 4, pp. 489 –509, jul. 1999.

[16] M. Rezaei, M. Hannuksela, and M. Gabbouj, "Semi-fuzzy rate controller for variable bit rate video," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 5, pp. 633–645, May 2008.

[17] A. Jagmohan and K. Ratakonda, "MPEG-4 one-pass VBR rate control for digital storage," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 5, pp. 447–452, 2003.

[18] M. de Frutos-Lopez, O. del Ama-Esteban, S. Sanz-Rodriguez, and F. Diaz-de Maria, "A two-level sliding-window VBR controller for real-time hierarchical video coding," in *Image Processing, 2010. ICIP 2010. IEEE International Conference on*, Sept. 2010.

[19] P. H. Westerink, R. Rajagopalan, and C. A. Gonzales, "Two-pass MPEG-2 variable-bit-rate encoding," *IBM Journal of Research and Development*, vol. 43, no. 4, pp. 471 –488, 1999.

[20] Y. Yu, J. Zhou, Y. Wang, and C. W. Chen, "A novel two-pass VBR coding algorithm for fixed-size storage application," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 11, no. 3, pp. 345 –356, Mar. 2001.

[21] W. Ding, "Joint encoder and channel rate control of VBR video over ATM networks," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 7, no. 2, pp. 266–278, 1997.

[22] J. Bai, Q. Liao, X. Lin, and X. Zhuang, "Rate-distortion model based rate control for real-time VBR video coding and low-delay communications," *Signal Processing: Image Communication*, vol. 17, no. 2, pp. 187 – 199, 2002.

[23] Z. Chen and K. N. Ngan, "Recent advances in rate control for video coding," vol. 22, no. 1. New York, NY, USA: Elsevier Science Inc., 2007, pp. 19–38.

[24] R. Schaefer, H. Schwarz, D. Marpe, T. Schierl, and T. Wiegand, "MCTF and scalability extension of H.264/AVC and its application to video transmission, storage, and surveillance," in *Proceedings of VCIP 2005, Peking, China*, July 2005.

[25] T. Wiegand, L. Noblet, and F. Rovati, "Scalable video coding for IPTV services," *Broadcasting, IEEE Transactions on*, vol. 55, pp. 527 –538, june 2009.

[26] 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.

[27] M. Wien, H. Schwarz, and T. Oelbaum, "Performance Analysis of SVC," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 17, no. 9, pp. 1194–1203, Sept. 2007.

[28] Y. Liu, Z. G. Li, and Y. C. Soh, "Rate control of H.264/AVC scalable extension," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 1, pp. 116–121, Jan. 2008.

[29] A. Leontaris and A. M. Tourapis, "Rate control for the Joint Scalable Video Model (JSVM)," *Video Team of ISO/IEC MPEG and ITU-T VCEG, JVT-W043, San Jose, California*, April 2007.

[30] M. Dai, D. Loguinov, and H. Radha, "Rate-distortion analysis and quality control in scalable internet streaming," *Multimedia, IEEE Transactions on*, vol. 8, no. 6, pp. 1135 –1146, 2006.

[31] Y. Pitrey, M. Babel, and O. Deforges, "One-pass bitrate control for MPEG-4 scalable video coding using ρ-domain," *Broadband Multimedia Systems and Broadcasting, 2009. BMSB '09. IEEE International Symposium on*, pp. 1–5, May 2009.

[32] M. Liu, Y. Guo, H. Li, and C.-W. Chen, "Low-complexity rate control based on ρ-domain model for scalable video coding," in *Image Processing, 2010. ICIP 2010. IEEE International Conference on*, Sept. 2010.

[33] L. Xu, W. Gao, X. Ji, D. Zhao, and S. Ma, "Rate control for spatial scalable coding in SVC," in *Picture Coding Symposium, 2007. PCS 2007*, Nov. 2007.

[34] Y. Cho, J. Liu, D.-K. Kwon, and C.-C. Kuo, "Joint quality-temporal (Q-T) bit allocation for H.264/SVC," in *Circuits and Systems, 2009. ISCAS 2009. IEEE International Symposium on*, May 2009, pp. 2361 –2364.

[35] J. Liu, Y. Cho, Z. Guo, and J. Kuo, "Bit allocation for spatial scalability coding of H.264/SVC with dependent rate-distortion analysis," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 20, no. 7, pp. 967 –981, 2010.

[36] A. Unterweger and H. Thoma, "The influence of bit rate allocation to scalability layers on video quality in H.264 SVC," in *Picture Coding Symposium, 2007. PCS 2007*, Nov. 2007.

[37] X. M. Zhang, A. Vetro, Y. Shi, and H. Sun, "Constant quality constrained rate allocation for FGS-coded video," *Circuits and Systems for Video*

Technology, IEEE Transactions on, vol. 13, no. 2, pp. 121–130, Feb 2003.
[38] H. Mansour, V. Krishnamurthy, and P. Nasiopoulos, "Rate and distortion modeling of medium grain scalable video coding," in *Image Processing, 2008. ICIP 2008. IEEE International Conference on*, Oct. 2008, pp. 2564–2567.
[39] H. Lee, Y. Lee, D. Lee, J. Lee, and H. Shin, "Implementing rate allocation and control for real-time H.264/SVC encoding," in *Consumer Electronics (ICCE), 2010 Digest of Technical Papers International Conference on*, 2010, pp. 269 –270.
[40] C. E. Rasmussen and C. Williams, *Gaussian Processes for Machine Learning*. MIT Press, 2006.
[41] E. Snelson and Z. Ghahramani, *Sparse Gaussian Processes using Pseudo-inputs*. NIPS 18, 2005.
[42] [Online], "http://www.gatsby.ucl.ac.uk/ snelson/".
[43] [Online], "http://www.tsc.uc3m.es/∼sescalona/RbfVbrSvc/."
[44] J. Vieron, M. Wien, and H. Schwarz, "JSVM 11 software," *24th Meeting: Geneva, Doc. JVT-X203*, July 2007.
[45] M. Wien and H. Schwarz, "Testing conditions for SVC coding efficiency and JSVM performance evaluation," *JVT-Q205, 16th JVT Meeting*, Poznan, Poland, July 2005.
[46] [Online], "http://media.xiph.org/ldv/pub/test_sequences/".
[47] G. Bjøntegaard, "Calculation of average PSNR differences between RD curves," *VCEG contribution, VCEG-M33, Austin*, April 2001.

Sergio Sanz-Rodríguez (S'07) received the Technical Telecommunication Engineering degree in 2001 and the Telecommunication Engineering degree in 2005, both from Universidad Politécnica de Madrid, Madrid, Spain. He is currently working towards his Ph.D. degree in Telecommunication Engineering at Universidad Carlos III de Madrid, Madrid, Spain.

His primary research interests include rate control for video coding, scalable video coding, perceptual video coding and video signal processing.

Fernando Díaz-de-María (M'97) received the Telecommunication Engineering degree in 1991 and Ph.D. degree in 1996 from Universidad Politécnica de Madrid, Madrid, Spain. From October 1996, he is an Associate Professor at the Department of Signal Theory and Communications, Universidad Carlos III de Madrid, Madrid, Spain. From Oct. 97, he has held several offices in both, his Department and his University.

His primary research interests include image and video analysis and coding. He has led numerous projects and contracts in these fields. He is co-author of several papers in prestigious international journals, two chapters in international books and quite a few papers in revised national and international conferences.