Inter-Layer Bit-Allocation for Scalable Video Coding

Guan-Ju Peng, *Member, IEEE,* Wen-Liang Hwang, *Senior Member, IEEE,*
and Sao-Jie Chen, *Senior Member, IEEE*

***Abstract*—In this paper, we present a theoretical analysis of the distortion in multi-layer coding structures. Specifically, we analyze the prediction structure used to achieve temporal, spatial, and quality scalability of scalable video coding (SVC), and show that the average peak-signal-to-noise (*PSNR*) of SVC is a weighted combination of the bit rates assigned to all the streams. Our analysis utilizes the end user's preference for certain resolutions. We also propose a rate-distortion (R-D) optimization algorithm, and compare its performance with that of a state-of-the-art scalable bit allocation algorithm. The reported experiment results demonstrate that the R-D algorithm significantly outperforms the compared approach in terms of the average *PSNR*.**

I. Introduction

Scalable video coding (SVC) facilitates the encoding of a bitstream containing representations with lower spatial resolutions, frame rates, and quality, which are designed to meet the requirements of the heterogeneous display and computational capabilities of the target devices. A client with restricted resources (display resolution, processing power and bandwidth) can only decode a part of the delivered bitstream. Thus, SVC can be used in wide range of multicast applications, such as Internet and wireless applications, where scalability is necessary in order to deal with the variable transmission conditions to the end users. Another benefit of SVC is that it can adapt to a network-aware environment on-the-fly [1], [2] when feedback is provided by the network and the end users.

H.264/SVC is a state-of-the art SVC codec that significantly reduces the gap in rate-distortion (R-D) efficiency between state-of-the art signal layer coding and scalable coding [3], [4]. The performance of SVC depends to a large extent on the settings of several parameters [5]. The quantization parameters (QP), the ratio of the I, P, and B frames, and the target bit rate have the most influence on the performance. In this paper, we study the multiple-layer bit rate allocation problem in SVC, also known as the optimal quantization parameter (QP) assignment to each layer in SVC. With the objective of simplifying the analysis without affecting its generality, we fix the values of several SVC coding parameters. Specifically, we assume that the motion vectors have been acquired already. In addition, we use the hierarchical B-frame structure for temporal scalability and inter-layer residual prediction for spatial and coarse-grain quality scalability [4].

The optimal bit allocation of a rate-constrained encoder control system is usually derived by applying the Lagrangian technique [6]. In contrast to single-layer video coding, SVC requires that all users are served simultaneously in a single bitstream. Thus, the data items in an SVC bitstream are highly correlated to each other. This inter-dependency can cause a coding error in one layer to propagate to other layers and thereby complicate the bit allocation process. Another factor that affects bit allocation under SVC is the end user's preference. For example, the bit allocation scheme for users subscribing to the highest resolution should be different from that for users subscribing to the lowest resolution, since the latter only uses the base layer information. Hence, the preferences for some resolutions should also be considered by the bit allocation scheme. However, incorporating users' preferences into the bit allocation process implies that the preference information should be acquired by the encoder through a feedback mechanism. This is usually considered a disadvantage in a broadcasting environment.

Ramchandran, Ortega, and Vitterli [7] studied bit allocation in a multi-layer coding environment. They model the distortion in all layers as a weighted average of the distortions of the layers, and then use R-D optimization based on the Lagrangian technique to optimize the weighted distortion. In [8], Schwartz and Wiegand propose an encoder control mechanism that jointly optimizes the coding parameters of the base layer and enhancement layers under H.264/SVC. Their algorithm also utilizes a weighted combination of the distortions of all the layers to balance the coding efficiency of different layers. Although the above approaches demonstrate the correlation between the coding performance and the values of the weighting factors, analyses of the derivation of the weighting factors are not provided. Recently, Koziri and Eleftheriadis [9] presented an interesting approach that models the distortion dependency between layers as a stochastic process for joint optimization of scalable coding. However, their analysis is limited to Gaussian sources and spatial dependency.

In this paper, we propose a theoretical analysis of the weighting factor approach for joint optimization of scalable coding. We analyze the effect of a coding error in one layer on the other layers in terms of the residual prediction of temporal, spatial, and quality scalability under SVC. Then, we demonstrate that the weighting factor of a layer i is a function of all the layers affected by the coding error in layer i, and the end user's preference for subscribing to the affected layers. Based on the analysis, we derive the main result, namely, the average *PSNR* can be represented as the

Guan-Ju Peng and Sao-Jie Chen are with the Graduate Institute of Electronic Engineering, National Taiwan University, Taipei, Taiwan.

Wen-Liang Hwang is with the Institute of Information Science, Academia Sinica, Taipei, Taiwan.

weighted combination of the bit rate assigned to each layer, where the coefficient is the weighting factor. We also propose an R-D optimization algorithm. Experiments on H.264/SVC JSVM 9.18 [10] demonstrate that the algorithm achieves a significant improvement over the state-of-the-art method [8], [7] in terms of the average *PSNR*.

The remainder of this paper is organized as follows. In the next section, we consider several issues that are relevant to bit allocation under SVC. In Section III, we analyze the coding error of the predicting frames and the predicted frame in two adjacent layers; and in Section IV, we extend the derived result to all the frames in adjacent layers. In Section V, we derive the R-D function of SVC; and in Section VI, we present our algorithm for solving the optimal bit allocation problem in SVC. We discuss a number of implementation issues and the experimental results in Section VII; and then summarize our conclusions in Section VIII.

II. Aspects of the SVC Bit-allocation Problem

In this section, we discuss three important aspects of the SVC bit-allocation method for a scalable video codec, namely, the R-D model, the measurement of the source coder's performance, and the structure of data dependency under SVC.

A. The rate-distortion model

In our analysis, we use He and Mitra's ρ-domain source model [11], which relates the number of zeros to the rate-distortion function of quantized DCT coefficients. Let ρ denote the percentage of zeros in the quantized DCT coefficients. In the model, the rate R is linearly dependent on ρ and the distortion D is exponentially dependent on ρ. The relations are shown in the following equations in which α and θ are parameters and σ is the picture variance:

$$\begin{aligned} R(\rho) &= \theta(1-\rho), && (1)\\ D(\rho) &= \sigma^2 e^{-\alpha(1-\rho)}, && (2) \end{aligned}$$

where D represents the mean-square-error (MSE). Substituting Equation (1) into Equation (2), we obtain the result such that

$$D(R) = \sigma^2 e^{-R\gamma}, \qquad (3)$$

and $\gamma = \alpha/\theta$. The parameter γ is propositional to $\frac{dPSNR}{dR}$:

$$\begin{aligned} PSNR &= 10\log_{10} 255^2 - 10\log_{10} e \ln D && (4)\\ &= 10\log_{10}\frac{255^2}{\sigma^2} + (10\log_{10} e)\gamma R. && (5) \end{aligned}$$

Equation (5) is obtained by substituting Equation (2) into Equation (4). He and Mitra's model assumes that γ is a constant; however, if γ is a constant, then, according to Equation (5), the *PSNR* and R are linearly related. This model is usually correct at high bit rates, but not so exact at low bit rates. Thus, we assume that the value of γ changes slowly with respect to R and can be approximated as a constant at high bit rates.

B. Quality measurement of SVC

To assess the performance of the multi-layer structure, we use the model proposed in [12]. Suppose there are N subscribers, from 1 to N, and the video quality they receive is measured by the peak-signal-to-noise ratio, i.e., $PSNR_1, PSNR_2, PSNR_3, ..., PSNR_N$, respectively. We also introduce the parameter ψ_i to denote the preference of subscriber i in the system. Then, the overall quality of the N subscriber system is $\sum_{i=1}^{N}\psi_i PSNR_i$.

A scalable codec supports several spatial, temporal, and quality resolutions. Let S, T and R represent the sets of spatial, temporal, and quality resolutions, respectively; and let $[s,t,r]$ denote a particular resolution with $s \in S$, $t \in T$, and $r \in R$. In addition, let $q(i)$ denote the resolution that subscriber i requests. Based on the subscribers to the resolution $[s,t,r]$, we have

$$\begin{aligned} Q_N &= \sum_{i=1}^{N}\psi_i PSNR_{q(i)} \\ &= \sum_{s\in S, t\in T, r\in R} PSNR_{[s,t,r]} \sum_{q(i)=[s,t,r]}\psi_i. && (6) \end{aligned}$$

If we normalize Q_N by dividing it by the preferences of all subscribers; i.e.,

$$\sum_{i=1}^{N}\psi_i = \sum_{s\in S, t\in T, r\in R}\ \sum_{q(i)=[s,t,r]}\psi_i, \qquad (7)$$

we obtain the average *PSNR*:

$$P\bar{S}NR = \sum_{s\in S, t\in T, r\in R}\mu_{[s,t,r]} PSNR_{[s,t,r]}, \qquad (8)$$

where the preference factor of the $[s,t,r]$ resolution is

$$\mu_{[s,t,r]} = \frac{\sum_{q(i)=[s,t,r]}\psi_i}{\sum_{s\in S, t\in T, r\in R}\sum_{q(i)=[s,t,r]}\psi_i}, \qquad (9)$$

which represents the proportion of preferences for the resolution $[s,t,r]$. If we replace the *PSNR* in Equation(8) by $10\log_{10}\frac{255^2}{D}$ and use the facts that $0 \le \mu_{[s,t,r]} \le 1$ and $\sum_{s\in S, t\in T, r\in R}\mu_{[s,t,r]} = 1$, we obtain

$$P\bar{S}NR = 10\log_{10} 255^2 - 10\log_{10}\prod_{s\in S, t\in T, r\in R} D_{[s,t,r]}^{\mu_{[s,t,r]}}. \qquad (10)$$

Equation (10) indicates that the maximization of the average *PSNR* can be obtained by minimizing

$$\prod_{s\in \mathbf{S}, t\in T, r\in R} D_{[s,t,r]}^{\mu_{[s,t,r]}}. \qquad (11)$$

C. Layer dependency and the sequence of approximations

To achieve high quality scalability, SVC usually encodes data into different layers of granularity. Recall that S, T and R represent the spatial, temporal, and quality layer identifiers respectively; and let (s,l,r) denote a particular *stream* in which the spatial layer identifer $s \in S$, the temporal *level* identifer $l \in T$, and the quality layer identifier $r \in R$. In this paper, we do not distinguish between layers and resolutions.

However, temporal layer l and temporal level l have different meanings [4]. *Specifically, temporal layer l contains all the frames in temporal levels $0, 1, \cdots, l$; and temporal level l only contains the frames in that level.*

The data dependency structure can be represented by a directed graph $G = (V, E)$ with a vertex set V and an edge set $E = V \times V$. A directed edge $\vec{uv}$ indicates that the edge is from vertex u to vertex v, but not from v to u. In SVC, a stream is represented by a vertex and the dependency between two streams is represented by an edge between their corresponding vertices. A directed edge from the stream (s, l, r) to the stream (s', l', r') indicates that the data in (s', l', r') is predicted based on the data in (s, l, r). We assume that the elements in S, T, and R can be enumerated as $S = \{0, 1, \cdots, |S| - 1\}$, $T = \{0, 1, \cdots, |T| - 1\}$, and $R = \{1, \cdots, |R|\}$.

In addition, we assume that SVC has the following prediction structure. The data in the stream (s, l, r) can be used to predict the data in streams $(s + 1, l, r)$ (spatial prediction), $(s, l, r + 1)$ (quality prediction), and $(s, l + 1, r)$ (temporal prediction), provided that $s + 1 < S$, $r \leq R$, and $l + 1 < T$. If the edge is defined accordingly, the directed graph is a directed acyclic graph that does not have cycles and any vertex (s, l, r) can be reached from $(0, 0, 1)$. In H.264/SVC, the *coarse-grain quality prediction* has a particular structure, as shown in Figure 1. The data in the stream (s, l, r) with $r < |R|$ predicts that in the stream $(s, l, r + 1)$; meanwhile, the data in the stream $(s, l, |R|)$ predicts that of stream $(s + 1, l, 1)$.

We use $I_{i,(s,l,\infty)}$ to indicate that the input frame i is of spatial resolution s, is in temporal level l. The coarsest approximation of $I_{i,(s,t,\infty)}$ is $I_{i,(s,t,1)}$, which is the reconstructed frame with one quality layer, and the next coarsest approximation is $I_{i,(s,t,2)}$, reconstructed with the first two quality layers, and so on. Thus, from coarse to fine, the sequence of approximation of $I_{i,(s,l,\infty)}$ is $I_{i,(s,l,1)}, I_{i,(s,l,2)}, \cdots$. To derive the coding error of the input frame $I_{i,(s,l,\infty)}$, we need to examine the error that occurs in each prediction stage in Figure 1 and its propagation to the other prediction stages.

III. Prediction Residuals and Distortion Propagation

In this section, we derive the prediction residuals of different types of data predictions, as well as the relations between the distortion of the predicted frame in one stream and that of the predicting frames in another stream.

A. Prediction residuals

In the derivations, we use a column vector to represent a frame, and assume that the motion vectors have been obtained. If $\Delta_{i,(s,l,r)}$ represents the coding error between $I_{i,(s,l,\infty)}$ and $I_{i,(s,l,r)}$, we have

$$I_{i,(s,l,r)} = I_{i,(s,l,\infty)} - \Delta_{i,(s,l,r)}, \tag{12}$$

where $I_{i,(s,l,r)}$ is the reconstructed frame at the quality layer r. Note that $\Delta_{i,(s,l,r)}$ decreases as the quality layer r increases.

Notations

In SVC, an input frame is subjected to temporal prediction, spatial prediction, and quality prediction. Thus, the notation used to represent an object must specify the prediction sequence applied to obtain the object. We use the following notations to represent objects:

1. $X^{r,\mathbf{t}}_{i,(s,l)}$ denotes an object X associated with frame i of spatial resolution s and temporal level l. The object is derived by applying temporal prediction to the input frame $I_{i,(s,l,\infty)}$ with the predicting frames of quality resolution r. For example, if $X = I$, then $I^{r,\mathbf{t}}_{i,(s,l)}$ is the predicted frame of $I_{i,(s,l,\infty)}$ when $I_{i,(s,l,\infty)}$ is temporally predicted with the predicting frames of quality resolution r. If $X = \Delta$, where Δ is the prediction residual, then $\Delta^{r,\mathbf{t}}_{i,(s,l)}$ denotes the residual obtained after applying temporal prediction to $I_{i,(s,l,\infty)}$ with the predicting frames of quality resolution r. Similarly, if $X = C$ and C is a constant, then $C^{r,\mathbf{t}}_{i,(s,l)}$ denotes the derived constant obtained in a similar way.
2. $X^{r,\mathbf{s}}_{i,(s,l)}$ denotes an object X associated with frame i of spatial resolution s and temporal level l. The object is derived by applying temporal and spatial prediction to the input frame $I_{i,(s,l,\infty)}$ with the predicting frames of quality resolution r. If $X = I$, then $I^{r,\mathbf{s}}_{i,(s,l)}$ is the predicted frame of $I_{i,(s,l,\infty)}$ when $I_{i,(s,l,\infty)}$ is temporally and spatially predicted with the predicting frames of quality resolution r. If $X = \Delta$, where Δ is the prediction residual, then $\Delta^{r,\mathbf{s}}_{i,(s,l)}$ is the residual obtained after temporal and spatial prediction of $I_{i,(s,l,\infty)}$ with the predicting frames of quality resolution r. Similarly, if $X = C$ and C is a constant, then $C^{r,\mathbf{s}}_{i,(s,l)}$ denotes the derived constant.
3. $X^{r,\mathbf{q}}_{i,(s,l)}$ denotes an object X obtained by applying temporal prediction and quality prediction to the input frame $I_{i,(s,l,\infty)}$ with the predicting frames of quality resolution r. If $X = I$, then $I^{r,\mathbf{q}}_{i,(s,l)}$ is the resulting predicted frame of $I_{i,(s,l,\infty)}$; and if $X = \Delta$, where Δ is the residual, then $\Delta^{r,\mathbf{q}}_{i,(s,l)}$ is the resulting residual. Similarly, if $X = C$ and C is a constant, then $C^{r,\mathbf{q}}_{i,(s,l)}$ is the derived constant.

Following Equation (12), the residuals $\Delta^{r,\mathbf{t}}_{i,(s,l)}$, $\Delta^{r,\mathbf{s}}_{i,(s,l)}$, and $\Delta^{r,\mathbf{q}}_{i,(s,l)}$ can be represented as $I_{i,(s,l,\infty)} - I^{r,\mathbf{t}}_{i,(s,l)}$, $I_{i,(s,l,\infty)} - I^{r,\mathbf{s}}_{i,(s,l)}$, and $I_{i,(s,l,\infty)} - I^{r,\mathbf{q}}_{i,(s,l)}$, respectively.

A. Temporal prediction

In SVC, temporal scalability with dyadic temporal levels can be implemented effectively by a hierarchical prediction structure. In temporal prediction, the macroblocks of the input frame $I_{i,(s,l,\infty)}$ can be either *INTER*- or *INTRA*- predicted. The *INTER* macroblocks are predicted by the corresponding reconstructed frames, $I_{i-m,(s,l-1,r)}$ and $I_{i+m,(s,l-1,r)}$, with $l \geq 1$. Let $A^{r,\mathbf{t}}_{i,(s,l)}$ denote the pixels predicted in *INTRA* macroblocks; then, $I^{r,\mathbf{t}}_{i,(s,l)}$, as defined in **Notation** 1 is obtained by

$$\begin{aligned}
& \Delta^{r,\mathbf{t}}_{i,(s,l)} \\
= \; & I_{i,(s,l,\infty)} - \\
& \{P_{i,i-m} I_{i-m,(s,l-1,r)} + P_{i,i+m} I_{i+m,(s,l-1,r)}\} - \\
& A^{r,\mathbf{t}}_{i,(s,l)} && (13) \\
= \; & (I_{i,(s,l,\infty)} - P_{i,i-m} I_{i-m,(s,l-1,\infty)} \\
& -P_{i,i+m} I_{i+m,(s,l-1,\infty)} - A^{r,\mathbf{t}}_{i,(s,l)}) + \\
& (P_{i,i-m}\Delta_{i-m,(s,l-1,r)} + P_{i,i+m}\Delta_{i+m,(s,l-1,r)}) \\
& && (14) \\
= \; & C^{r,\mathbf{t}}_{i,(s,l)} + \\
& (P_{i,i-m}\Delta_{i-m,(s,l-1,r)} + P_{i,i+m}\Delta_{i+m,(s,l-1,r)}), \\
& && (15)
\end{aligned}$$

where P is the matrix representation of the motion compensation prediction method for $INTER$ macroblocks, and $C^{r,\mathbf{t}}_{i,(s,l)}$ is the constant error, as defined in **Notation** 1. The second term in Equation (15) represents the propagation of the residuals $\Delta_{i-m,(s,l-1,r)}$ and $\Delta_{i+m,(s,l-1,r)}$ of the reconstructed predicting frames in the previous temporal level. If we assume that the first and second terms in Equation (15) are uncorrelated, then we have

$$\begin{aligned}
& \sigma^2_{i,(s,l,r)} = var(C^{r,\mathbf{t}}_{i,(s,r)}) + \\
& var(P_{i,i-m}\Delta_{i-m,(s,l-1,r)} + P_{i,i+m}\Delta_{i+m,(s,l-1,r)}). \\
& && (16)
\end{aligned}$$

Note that $\sigma^2_{i,(s,l,r)}$ is the variance, which is used as part of the distortion calculation by the ρ-domain source model shown in Equation (3).

B. Spatial prediction

Although spatial prediction is sometimes referred to as *INTRA* prediction, in this paper, it means prediction based on the information about the video at a lower spatial resolution. Spatial prediction of the *INTER* macroblocks is achieved by inter-layer spatial residual prediction, which predicts a temporal residual by up-sampling the corresponding reconstructed temporal residual in the previous spatial resolution. We use the matrix U to denote an up-sampling method applied to macroblocks in a frame. If a macroblock is not $INTER$-predicted, it can be predicted by either $INTRA$ prediction or inter-layer intra prediction. Since $INTRA$-predicted macroblocks have been considered, we only need to take the inter-layer intra predicted macroblocks into account. Let $A^{r,\mathbf{s}}_{i,(s,l)}$ denote the predicted pixels in the macroblocks to which inter-layer intra-prediction is applied. The residual $\Delta^{1,\mathbf{s}}_{i,(s,l)}$, as defined in **Notation** 2, can be derived with $s \geq 1$ as follows:

$$\Delta^{1,\mathbf{s}}_{i,(s,l)} = \Delta^{1,\mathbf{t}}_{i,(s,l)} - U(I_{i,(s-1,l,|R|)} - I^{|R|,\mathbf{q}}_{i,(s-1,l)}) - A^{r,\mathbf{s}}_{i,(s,l)}, \quad (17)$$

where the first term is the temporal residual in Equation (15); and the second term is the up-sampled reconstructed residual of the macroblocks, where inter-layer residual prediction is applied in frame i in $(s-1, l, |R|)$. This equation indicates that the residual $\Delta^{1,\mathbf{t}}_{i,(s,l)}$ is being predicted. Substituting Equation (12) for $I_{i,(s-1,l,|R|)}$ and Equation (15) for $\Delta^{1,\mathbf{t}}_{i,(s,l)}$, we obtain

$$\begin{aligned}
& \Delta^{1,\mathbf{s}}_{i,(s,l)} \\
= \; & (C^{1,\mathbf{t}}_{i,(s,l)} + \\
& (P_{i,i-m}\Delta_{i-m,(s,l-1,1)} + P_{i,i+m}\Delta_{i+m,(s,l-1,1)}) \\
& -A^{r,\mathbf{s}}_{i,(s,l)}) - \\
& U(I_{i,(s-1,l,\infty)} - \Delta_{i,(s-1,l,|R|)} - I^{|R|,\mathbf{q}}_{i,(s-1,l)}) && (18) \\
= \; & (C^{1,\mathbf{t}}_{i,(s,l)} - U(\Delta^{|R|,\mathbf{q}}_{i,(s-1,l)}) - A^{r,\mathbf{s}}_{i,(s,l)}) + \\
& (P_{i,i-m}\Delta_{i-m,(s,l-1,1)} + P_{i,i+m}\Delta_{i+m,(s,l-1,1)}) \\
& +U(\Delta_{i,(s-1,l,|R|)}) && (19) \\
= \; & C^{1,\mathbf{s}}_{i,(s,l)} + \\
& (P_{i,i-m}\Delta_{i-m,(s,l-1,1)} + P_{i,i+m}\Delta_{i+m,(s,l-1,1)}) + \\
& U(\Delta_{i,(s-1,l,|R|)}), && (20)
\end{aligned}$$

where the first term $C^{r,\mathbf{s}}_{i,(s,l)}$ is the constant, the second term is the error propagated from $(s, l-1, r)$, and the third term is the error propagated from $(s-1, l, r)$. Assuming these terms are uncorrelated, for the streams with $s > 0$ and $r = 1$ (only temporal prediction and spatial prediction are applied), the variance of the residual $\Delta^{1,\mathbf{s}}_{i,(s,l)}$ is

$$\begin{aligned}
& \sigma^2_{i,(s,l,1)} \\
= \; & var(C^{1,\mathbf{s}}_{i,(s,l)}) + \\
& var(P_{i,i-m}\Delta_{i-m,(s,l-1,1)} + P_{i,i+m}\Delta_{i+m,(s,l-1,1)}) \\
& +var(U(\Delta_{i,(s-1,l,|R|)})). && (21)
\end{aligned}$$

C. Quality prediction

Coarse-grain quality prediction of H.264/SVC is similar to spatial prediction after removing the up-sampling operator in Equation (17). We use the matrix Y to select the pixels used by inter-layer residual prediction. Let $A^{r,\mathbf{q}}_{i,(s,l)}$ denote the pixels in the macroblocks predicted by inter-layer intra prediction; then, the residual $\Delta^{r,\mathbf{q}}_{i,(s,l)}$ (as defined in **Notation** 3) with $r \geq 2$ can be represented as follows:

$$\begin{aligned}
& \Delta^{r,\mathbf{q}}_{i,(s,l)} \\
= \; & (I_{i,(s,l,\infty)} - I^{r,\mathbf{t}}_{i,(s,l)}) - \\
& Y(I_{i,(s,l,r-1)} - I^{r-1,\mathbf{q}}_{i,(s,l)}) - A^{r,\mathbf{q}}_{i,(s,l)} && (22) \\
= \; & (\Delta^{r,\mathbf{t}}_{i,(s,l)}) - Y(I_{i,(s,l,r-1)} - I^{r-1,\mathbf{q}}_{i,(s,l)}) - A^{r,\mathbf{q}}_{i,(s,l)} \\
= \; & (C^{r,\mathbf{t}}_{i,(s,l)} + \\
& (P_{i,i-m}\Delta_{i-m,(s,l-1,r)} + P_{i,i+m}\Delta_{i+m,(s,l-1,r)})) \\
& +Y(\Delta_{i,(s,l,r-1)}) - A^{r,\mathbf{q}}_{i,(s,l)} \\
= \; & (C^{r,\mathbf{t}}_{i,(s,l)} - A^{r,\mathbf{q}}_{i,(s,l)}) + \\
& (P_{i,i-m}\Delta_{i-m,(s,l-1,r)} + P_{i,i+m}\Delta_{i+m,(s,l-1,r)}) \\
& +Y(\Delta_{i,(s,l,r-1)}) \\
= \; & (C^{r,\mathbf{q}}_{i,(s,l)}) + \\
& (P_{i,i-m}\Delta_{i-m,(s,l-1,r)} + P_{i,i+m}\Delta_{i+m,(s,l-1,r)}) \\
& +Y(\Delta_{i,(s,l,r-1)}).
\end{aligned}$$

Since $\Delta^{r,\mathbf{t}}_{i,(s,l)} = I_{i,(s,l,\infty)} - I^{r,\mathbf{t}}_{i,(s,l)}$, the quality prediction is a residual prediction with the residual $\Delta^{r,\mathbf{t}}_{i,(s,l)}$ being predicted.

Similar to Equation (21), the distortion of the residual $\Delta^{r,\mathbf{q}}_{i,(s,l)}$ can be written as

$$\begin{aligned}&\sigma^2_{i,(s,l,r)} = var(C^{r,\mathbf{q}}_{i,(s,l)}) + \\ &var(P_{i,i-m}\Delta_{i-m,(s,l-1,r)} + P_{i,i+m}\Delta_{i+m,(s,l-1,r)}) \\ &+var(Y(\Delta_{i,(s,l,r-1)})).\end{aligned} \tag{23}$$

Although the effects of *INTRA* prediction and inter-layer intra-prediction are considered in (13), (17), and (22), in the following analysis, we do not take account of the errors propagated to them. Inter-layer intra prediction is applied to macroblocks whose co-located macroblocks in the base layer are $INTRA$-predicted. The statistical results indicate that the probability of a macroblock having an *INTRA* mode in B frames is at most 7% ($QP = 50$) and 4% on average [13]. As a consequence, we can still derive a good approximation of the optimal rate allocation even when the error propagations of *INTRA* prediction and inter-layer intra-prediction are excluded.

B. Distortion propagation in spatial, temporal, and quality prediction

In this sub-section, we explore the relationship between the distortion of the predicted frame and that of the predicting frames. Under H.264/SVC, quality prediction is not applied to quality layer 1, and spatial prediction is not applied to spatial resolution 0. Thus, the derivation of the distortion relationship between the predicting and predicted frames can be divided into three cases: (1) stream $(0, l, 1)$ with $l \geq 1$, where only temporal prediction is used; (2) stream $(s, l, 1)$ with $s, l \geq 1$, where temporal and spatial prediction are used; and (3) stream (s, l, r) with $s, l \geq 1$, $r \geq 2$, where temporal and quality prediction are used.

Case 1: stream $(0, l, 1)$ with $l \geq 1$. Note that the following derivations can also be applied to $l = 0$, since the stream $(0, 0, 1)$ in the current GOP is temporally predicted by the stream $(0, 0, 1)$ in the previous GOP.

In Equation (16), if we substitute 0 and 1 for s and r respectively, we obtain

$$\begin{aligned}&\sigma^2_{i,(0,l,1)} = var(C^{1,\mathbf{t}}_{i,(0,l)}) + \\ &var(P_{i,i-m}\Delta_{i-m,(0,l-1,1)} + P_{i,i+m}\Delta_{i+m,(0,l-1,1)}).\end{aligned} \tag{24}$$

The equation can be re-written as

$$\sigma^2_{i,(0,l,1)} = h^T_{i,(0,l,1)} \begin{pmatrix} var(C^{1,\mathbf{t}}_{i,(0,l)}) \\ var(P_{i,i-m}\Delta_{i-m,(0,l-1,1)} + P_{i,i+m}\Delta_{i+m,(0,l-1,1)}) \end{pmatrix}, \tag{25}$$

where

$$\begin{aligned}&h_{i,(0,l,1)}[0] = (\kappa^\alpha)_{i,(0,l,1)}\mathbf{1}\{var(C^{1,\mathbf{t}}_{i,(0,l)}) \geq \\ &var(P_{i,i-m}\Delta_{i-m,(0,l-1,1)} + P_{i,i+m}\Delta_{i+m,(0,l-1,1)})\}; \\ &h_{i,(0,l,1)}[1] = (\kappa^\beta)_{i,(0,l,1)}\mathbf{1}\{var(C^{1,\mathbf{t}}_{i,(0,l)}) < \\ &var(P_{i,i-m}\Delta_{i-m,(0,l-1,1)} + P_{i,i+m}\Delta_{i+m,(0,l-1,1)})\}.\end{aligned} \tag{26}$$

with $(\kappa^\alpha)_{i,(0,l,1)} = \frac{\sigma^2_{i,(0,l,1)}}{var(C^{1,\mathbf{t}}_{i,(0,l)})}$ and $(\kappa^\beta)_{i,(0,l,1)} = \frac{\sigma^2_{i,(0,l,1)}}{var(P_{i,i-m}\Delta_{i-m,(0,l-1,1)}+P_{i,i+m}\Delta_{i+m,(0,l-1,1)})}$; the value of the indicator function $\mathbf{1}\{$*Statement*$\}$ is 1 if *Statement* is true, and 0 otherwise. Note that $h_{i,(0,l,1)}$ is a 2×1 vector with one of its components set to zero. In a value larger than 0, the component in h is called the dominating term of h because the distortion of the component is larger than that of the other components.

Equations (24) and (25) have different interpretations of distortion propagation. Equation (24) indicates that the variance $\sigma^2_{i,(0,l,1)}$ is contributed by two terms: one from the distortion propagation in the predicting frames, and the other from the coding of the predicted frame. In contrast, Equation (25) indicates that the variance is caused by the distortion propagation in the predicting frames or by encoding of the predicted frame, but not both. Thus, Equation (25) can be regarded as an approximation of Equation (24) by assuming that the variance is propagated from the distortion of the predicting frames or caused by encoding the predicted frame. The approximation greatly simplifies our analysis of distortion propagation in the complex prediction structure of H.264/SVC. In Appendix 1, we provide a simple example to illustrate the approximation's effect on the analysis of distortion propagation.

If the number of bits assigned to encode $\Delta_{i-m,(0,l-1,1)}$ and $\Delta_{i+m,(0,l-1,1)}$ is small (i.e., a low bit rate), the variance of $(P_{i,i-m}\Delta_{i-m,(0,l-1,1)} + P_{i,i+m}\Delta_{i+m,(0,l-1,1)})$ will be large; hence, at a low bit rate, the error of the predicting frames in the previous stream is propagated to and dominates the distortion of the predicted frame in the current stream. On the other hand, if a sufficiently large number of bits are assigned to encode $\Delta_{i-m,(0,l-1,1)}$ and $\Delta_{i+m,(0,l-1,1)}$ (i.e., a high bit rate), the variance of $(P_{i,i-m}\Delta_{i-m,(0,l-1,1)}+P_{i,i+m}\Delta_{i+m,(0,l-1,1)})$ will be small. Thus, at a high bit rate, the error of the predicting frames in the previous stream is irrelevant to the predicted frame in the current stream because $var(C^{1,\mathbf{t}}_{i,(0,l)})$ is a constant.

In Appendix 2, we show that, at low bit rates,

$$\begin{aligned}&\sigma^2_{i,(0,l,1)} \approx \\ &((C^\alpha)^{1,\mathbf{t}}_{i-m,(0,l-1)}(C^\beta)^{1,\mathbf{t}}_{i+m,(0,l-1)} \\ &D_{i-m,(0,l-1,r)}D_{i+m,(0,l-1,r)})^{\frac{1}{2}},\end{aligned} \tag{27}$$

where $(C^\alpha)^{1,\mathbf{t}}_{i-m,(0,l-1)}$ and $(C^\beta)^{1,\mathbf{t}}_{i+m,(0,l-1)}$ are constants, as defined in **Notation** 2. The above equation indicates that the variance of the temporal residual is related to the distortion of the associated predicting frames by the geometric mean of $(C^\alpha)^{1,\mathbf{t}}_{i-m,(0,l-1)}D_{i-m,(0,l-1,r)}$ and $(C^\beta)^{1,\mathbf{t}}_{i+m,(0,l-1)}D_{i+m,(0,l-1,r)}$.

Case 2: stream $(s, l, 1)$ with $s, l \geq 1$.

In this case, the frames are predicted by both temporal prediction and spatial prediction. By Equation (21), we obtain

$$\begin{aligned}\sigma^2_{i,(s,l,1)} &= var(C^{1,\mathbf{s}}_{i,(s,l)}) \\ &+ var(P_{i,i-m}\Delta_{i-m,(s,l-1,1)} + P_{i,i+m}\Delta_{i+m,(s,l-1,1)}) \\ &+ var(U(\Delta_{i,(s-1,l,|R|)})).\end{aligned} \tag{28}$$

Similar to Case 1, we rewrite Equation (28) as follows:

$$\sigma^2_{i,(s,l,1)} = h^T_{i,(s,l,1)} \begin{pmatrix} var(C^{1,\mathbf{s}}_{i,(s,l)}) \\ var(P_{i,i-m}\Delta_{i-m,(s,l-1,1)} + P_{i,i+m}\Delta_{i+m,(s,l-1,1)}) \\ var(U(\Delta_{i,(s-1,l,|R|)})) \end{pmatrix}, \tag{29}$$

where $h_{i,(s,l,1)}$ is a 3×1 vector in which only one of the components has a non-zero value (i.e., the dominating term). For example, the first component of $h_{i,(s,l,1)}$ will be non-zero, denoted as $(\kappa^{\alpha})_{i,(s,l,1)}$, if all the variances of $\Delta_{i-m,(s,l-1,1)}$, $\Delta_{i+m,(s,l-1,1)}$, and $\Delta_{i,(s-1,l,|R|)}$ have smaller values, corresponding to the high bit rates assigned to the predicting frames in streams $(s,l-1,1)$ and $(s-1,l,|R|)$; that is,

$$var(C^{1,\mathbf{s}}_{i,(s,l)}) \geq var(P_{i,i-m}\Delta_{i-m,(s,l-1,1)} + P_{i,i+m}\Delta_{i+m,(s,l-1,1)}), \tag{30}$$

and

$$var(C^{1,\mathbf{s}}_{i,(s,l)}) \geq var(U(\Delta_{i,(s-1,l,|R|)})). \tag{31}$$

Similarly, $(\kappa^{\beta})_{i,(s,l,1)}$ indicates that the second component of $h_{i,(s,l,1)}$ a non-zero component; and $(\kappa^{\gamma})_{i,(s,l,1)}$ indicates that the third component of $h_{i,(s,l,1)}$ a non-zero component. The second component will be non-zero if the predicting frames in stream $(s,l-1,1)$ are assigned low bit rates; that is,

$$var(P_{i,i-m}\Delta_{i-m,(s,l-1,1)} + P_{i,i+m}\Delta_{i+m,(s,l-1,1)}) > var(C^{1,\mathbf{s}}_{i,(s,l)}), \tag{32}$$

and

$$var(P_{i,i-m}\Delta_{i-m,(s,l-1,1)} + P_{i,i+m}\Delta_{i+m,(s,l-1,1)}) \geq var(U(\Delta_{i,(s-1,l,|R|)})). \tag{33}$$

The third component will be non-zero if the predicting frame in stream $(s-1,l,|R|)$ is assigned a low bit rate; that is,

$$var(U(\Delta_{i,(s-1,l,|R|)})) > var(C^{1,\mathbf{s}}_{i,(s,l)}), \tag{34}$$

and

$$var(U(\Delta_{i,(s-1,l,|R|)})) > var(P_{i,i-m}\Delta_{i-m,(s,l-1,1)} + P_{i,i+m}\Delta_{i+m,(s,l-1,1)}). \tag{35}$$

Note that $(\kappa^{\alpha})_{i,(s,l,1)} = \frac{\sigma^2_{i,(s,l,1)}}{var(C^{1,\mathbf{s}}_{i,(s,l)})}$, $(\kappa^{\beta})_{i,(s,l,1)} = \frac{\sigma^2_{i,(s,l,1)}}{var(P_{i,i-m}\Delta_{i-m,(s,l-1,1)}+P_{i,i+m}\Delta_{i+m,(s,l-1,1)})}$, and $(\kappa^{\gamma})_{i,(s,l,1)} = \frac{\sigma^2_{i,(s,l,1)}}{var(U(\Delta_{i,(s-1,l,|R|)}))}$.

In Appendix 3, we show that, at low bit rates, $var(U(\Delta_{i,(s-1,l,|R|)}))$ can be approximated as

$$\sigma^2_{i,(s,l,1)} = (C^{\omega})^{|R|,\mathbf{s}}_{i,(s-1,l)} D_{i,(s-1,l,|R|)}, \tag{36}$$

where $(C^{\omega})^{|R|,\mathbf{s}}_{i,(s-1,l)}$ depends on the up-sampling method employed. If bilinear interpolation is used to up-sample most of the macroblocks in $\Delta_{i,(s-1,l,|R|)}$, then $(C^{\omega})^{|R|,\mathbf{s}}_{i,(s-1,l)} \approx 1$.
Case 3: stream (s,l,r) with $s,l \geq 1$ and $r \geq 2$.

In this case, both temporal prediction and quality prediction are applied to predict a frame, and the variance of the predicted frame is calculated according to Equation (23). Similar to the previous cases, we rewrite Equation (23) as follows:

$$\sigma^2_{i,(s,l,r)} = h^T_{i,(s,l,r)} \begin{pmatrix} var(C^{r,\mathbf{q}}_{i,(s,l)}) \\ var(P_{i,i-m}\Delta_{i-m,(s,l-1,r)} + P_{i,i+m}\Delta_{i+m,(s,l-1,r)}) \\ var(Y(\Delta_{i,(s,l,r-1)})) \end{pmatrix}, \tag{37}$$

where $h_{i,(s,l,r)}$ is a 3×1 vector in which only one component non-zero (i.e., the dominating term).

We set the first component $h_{i,(s,l,r)}[1] = (\kappa^{\alpha})_{i,(s,l,r)}$, where $(\kappa^{\alpha})_{i,(s,l,r)} = \frac{\sigma^2_{i,(s,l,r)}}{var(C^{r,\mathbf{q}}_{i,(s,l)})}$, if $var(C^{r,\mathbf{q}}_{i,(s,l)})$ is the dominating term. This occurs when the values of $\Delta_{i-m,(s,l-1,r)}$, $\Delta_{i+m,(s,l-1,r)}$, and $\Delta_{i,(s,l,r-1)}$ are smaller, corresponding to the high bit rates assigned to the predicting frames in streams $(s,l-1,r)$ and $(s,l,r-1)$. Next, we set the second component $h_{i,(s,l,r)}[2] = (\kappa^{\beta})_{i,(s,l,r)}$, where $(\kappa^{\beta})_{i,(s,l,r)} = \frac{\sigma^2_{i,(s,l,r)}}{var(P_{i,i-m}\Delta_{i-m,(s,l-1,r)}+P_{i,i+m}\Delta_{i+m,(s,l-1,r)})}$, if the dominating term is $var(P_{i,i-m}\Delta_{i-m,(s,l-1,r)} + P_{i,i+m}\Delta_{i+m,(s,l-1,r)})$. This occurs when the predicting frames in stream $(s,l-1,r)$ are assigned low bit rates. Finally, we set the third component $h_{i,(s,l,r)}[3] = (\kappa^{\gamma})_{i,(s,l,r)}$, where $(\kappa^{\gamma})_{i,(s,l,r)} = \frac{\sigma^2_{i,(s,l,r)}}{var(Y(\Delta_{i,(s,l,r-1)}))}$, to indicate that $var(Y(\Delta_{i,(s,l,r-1)}))$ is the dominating term, which occurs when the predicting frame in stream $(s,l,r-1)$ is assigned a low bit rate.

In Appendix 4, we show that, at low bit rates, $var(Y(\Delta_{i,(s,l,r-1)}))$ can be approximated as

$$\sigma^2_{i,(s,l,r)} = (C^{\omega})^{r-1,\mathbf{q}}_{i,(s,l)} D_{i,(s,l,r-1)}, \tag{38}$$

where $(C^{\omega})^{r-1,\mathbf{q}}_{i,(s,l)}$ depends on the amount of inter-layer residual prediction applied to the macroblocks in $\Delta_{i,(s,l,r)}$. If it is applied to most of the macroblocks in $\Delta_{i,(s,l,r)}$, then $(C^{\omega})^{r-1,\mathbf{q}}_{i,(s,l)} \approx 1$.
We can summarize the three approximations of $\sigma^2_{i,(s,l,r)}$ as follows:
For the frames in stream $(0,l,1)$ with $l \geq 1$, we have

$$\sigma^2_{i,(0,l,1)} = h^T_{i,(0,l,1)} \begin{pmatrix} var(C^{1,\mathbf{t}}_{i,(0,l)}) \\ ((C^{\alpha})^{1,\mathbf{t}}_{i-m,(0,l-1)} (C^{\beta})^{1,\mathbf{t}}_{i+m,(0,l-1)} D_{i-m,(0,l-1,r)} D_{i+m,(0,l-1,r)})^{\frac{1}{2}}. \end{pmatrix}. \tag{39}$$

For the frames in stream $(s,l,1)$ with $s,l \geq 1$, we have

$$\sigma^2_{i,(s,l,1)} = h^T_{i,(s,l,1)} \begin{pmatrix} var(C^{1,\mathbf{s}}_{i,(s,l)}) \\ ((C^{\alpha})^{1,\mathbf{t}}_{i-m,(s,l-1)} (C^{\beta})^{1,\mathbf{t}}_{i+m,(s,l-1)} D_{i-m,(s,l-1,1)} D_{i+m,(s,l-1,1)})^{\frac{1}{2}} \\ (C^{\omega})^{|R|,\mathbf{s}}_{i,(s-1,l)} D_{i,(s-1,l,|R|)})) \end{pmatrix}. \tag{40}$$

For the frames in stream (s,l,r) with $s,l \geq 1$ and $r \geq 2$, we have

$$\sigma^2_{i,(s,l,r)} = h^T_{i,(s,l,r)} \begin{pmatrix} var(C^{r,\mathbf{q}}_{i,(s,l)}) \\ ((C^{\alpha})^{r,\mathbf{t}}_{i-m,(s,l-1)} (C^{\beta})^{r,\mathbf{t}}_{i+m,(s,l-1)} D_{i-m,(s,l-1,r)} D_{i+m,(s,l-1,r)})^{\frac{1}{2}} \\ (C^{\omega})^{r-1,\mathbf{q}}_{i,(s,l)} D_{i,(s,l,r-1)} \end{pmatrix}. \tag{41}$$

IV. Distortion of a Layer

We represent the distortion of a stream as a function of the bits assigned to encode the layers along the residual prediction path. Recall that, in our analysis, we adopt the dyadic temporal enhancement layer structure in which the input frames of a group of pictures (GOP) are organized in different temporal levels. We use $nu(l)$ to denote the number of frames in temporal level identifier l. The dyadic temporal enhancement layer structure means that

$$nu(0) = n_0, \;\; and \;\; nu(l) = 2nu(l-1) \;\; for \;\; l \geq 1, \qquad (42)$$

where n_0 is the number of frames in the stream $(0, 0, 1)$. For convenience, we define that $nu(-1) = 0$. The number of frames in temporal level l is $2^{l-1}n_0$. A user who subscribes to temporal resolution t will receive all the frames in the temporal level identifiers 0, 1, $\cdots$, t. Thus, the user will receive $\sum_{i=0}^{t} nu(i) = 2^t n_0$ frames. Note that the number of frames in the temporal layer identifier t differs from the number of frames in the temporal resolution t. If we enumerate the frames from 1 to $2^t n_0$ and divide them into $t+1$ temporal layers with identifiers from 0 to t, the frames numbered from 1 to n_0 will be assigned to temporal level identifier 0; and the frames numbered from $2^{l-1}n_0 + 1$ to $2^l n_0$ will be assigned to temporal layer identifier l, with $l \geq 1$. The average distortion of the frames in the resolution $[s, t, r]$, denoted by $D_{[s,t,r]}$, can be derived by calculating the geometric mean of the frame distortion $D_{i,[s,t,r]}$ as follows:

$$D_{[s,t,r]} = (\prod_{l=0}^{t} \prod_{i=nu(l-1)+1}^{nu(l)} D_{i,(s,l,r)})^{\frac{1}{2^t n_0}}. \qquad (43)$$

Thus, the distortion of the resolution $[s, t, r]$ is the geometric mean of the distortions in stream, $(s, 0, r), \cdots, (s, t, r)$. We let

$$b_{(s,l,r)} = \sum_{i=nu(l-1)+1}^{nu(l)} b_{i,(s,l,r)} \qquad (44)$$

denote the total number of bits assigned to the frames in the stream (s, l, r), and

$$\sigma^2_{(s,l,r)} = \prod_{i=nu(l-1)+1}^{nu(l)} \sigma^2_{i,(s,l,r)} \qquad (45)$$

denote the product variance of the frames in the stream (s, l, r).

We have analyzed the relationship between the distortion of the predicted frame and that of the predicting frames. In Section IV-A, we extend the results to the distortion between all the frames in the predicting layer and predicted layer. Then, in Section IV-B, we provide an example of error propagation along the prediction path when the referred layers are encoded at low bit rates.

A. Prediction error propagation in a temporal level

According to the modeling in Equation (3), the distortion of a frame $I_{i,(s,l,r)}$ is

$$D_{i,(s,l,r)} \;\; = \;\; \sigma^2_{i,(s,l,r)} \exp\{-\gamma_{i,(s,l,r)} b_{i,(s,l,r)}\}. \qquad (46)$$

To derive the distortion of a temporal *level* that is only *temporally* predicted, according to Equations (39), (46) and (45), the distortion of $(0, l, 1)$ is

$$\begin{aligned} & \prod_{i=nu(l-1)+1}^{nu(l)} D_{i,(0,l,1)} \\ = & \prod_{i=nu(l-1)+1}^{nu(l)} \sigma^2_{i,(0,l,1)} \exp\{-\gamma_{i,(0,l,1)} b_{i,(0,l,1)}\} \\ = & \prod_{i=nu(l-1)+1}^{nu(l)} h^T_{i,(0,l,1)} \\ & \left(\begin{array}{c} {\scriptstyle var(C^{1,\mathbf{t}}_{i,(0,l)})} \\ {\scriptstyle (C^{1,\mathbf{t},\alpha}_{i-m,(0,l-1)} C^{1,\mathbf{t},\beta}_{i+m,(0,l-1)} D_{i-m,(0,l-1,r)} D_{i+m,(0,l-1,r)})^{\frac{1}{2}}} \end{array} \right) \\ & \exp\{-\gamma_{i,(0,l,1)} b_{i,(0,l,1)}\}. \end{aligned} \qquad (47)$$

Note that each frame in the predicted temporal level l is predicted by two frames in the predicting level $l-1$: one for forward prediction and the other for backward prediction. If $l \geq 2$, the predicted level will have twice as many frames as the predicting level.

Similarly, to derive the distortion of the temporally and spatially predicted stream $(s, l, 1)$, with $s \geq 1$, Equations (40), (46) and (45) are used. The distortion of $(s, l, 1)$ can be computed as

$$\begin{aligned} & \prod_{i=nu(l-1)+1}^{nu(l)} D_{i,(s,l,1)} \\ = & \prod_{i=nu(l-1)+1}^{nu(l)} \sigma^2_{i,(s,l,1)} \exp\{-\gamma_{i,(s,l,1)} b_{i,(s,l,1)}\} \\ = & \prod_{i=nu(l-1)+1}^{nu(l)} h^T_{i,(s,l,1)} \\ & \left(\begin{array}{c} {\scriptstyle var(C^{1,\mathbf{s}}_{i,(s,l)})} \\ {\scriptstyle ((C^{\alpha})^{1,\mathbf{t}}_{i-m,(s,l-1)} (C^{\beta})^{1,\mathbf{t}}_{i+m,(s,l-1)} D_{i-m,(s,l-1,1)} D_{i+m,(s,l-1,1)})^{\frac{1}{2}}} \\ {\scriptstyle (C^{\omega})^{|R|,\mathbf{s}}_{i,(s-1,l)} D_{i,(s-1,l,|R|)}} \end{array} \right) \\ & \exp\{-\gamma_{i,(s,l,1)} b_{i,(s,l,1)}\}. \end{aligned} \qquad (48)$$

The distortion of the stream (s, l, r) after *quality* prediction ($r \geq 2$), can be related to the distortion of the frames in streams $(s,l-1,r)$ and $(s,l,r-1)$ as follows:

$$\begin{aligned}
&\prod_{i=nu(l-1)+1}^{nu(l)} D_{i,(s,l,r)} \\
= &\prod_{i=nu(l-1)+1}^{nu(l)} \sigma^2_{i,(s,l,r)} \exp\{-\gamma_{i,(s,l,r)} b_{i,(s,l,r)}\} \\
= &\prod_{i=nu(l-1)+1}^{nu(l)} h^T_{i,(s,l,r)} \\
&\left(\begin{array}{c} var(C^{r,\mathbf{q}}_{i,(s,l)}) \\ ((C^{\alpha})^{r,\mathbf{t}}_{i-m,(s,l-1)} (C^{\beta})^{r,\mathbf{t}}_{i+m,(s,l-1)} D_{i-m,(s,l-1,r)} D_{i+m,(s,l-1,r)})^{\frac{1}{2}} \\ (C^{\omega})^{r-1,\mathbf{q}}_{i,(s,l)} D_{i,(s,l,r-1)} \end{array} \right) \\
&\exp\{-\gamma_{i,(k,l,r)} b_{i,(k,l,r)}\}.
\end{aligned} \tag{49}$$

In the above derivation, we use the results of Equations (41), (44), (45), and (46).

B. Exploring error propagation

The derivation in Section IV-A can be used to determine the propagation of the coding error in one stream to other streams. As mentioned earlier, if a predicting stream is encoded at a high bit rate, its coding error will not be propagated to the predicted stream; on the other hand, if it is encoded at a low bit rate, the coding error can propagate to the predicted stream. The error propagation can be explored by substituting the distortion in Equation (46) for $D_{i-m,(0,l-1,r)}D_{i+m,(0,l-1,r)}$ in Equations (47) and (48), $D_{i-m,(0,l-1,r)}D_{i+m,(0,l-1,r)}$, $D_{i,(s-1,l,1)}$ in Equation (48), and $D_{i,(s,l,r-1)}$ in Equation (49). Let us assume that the h value of each frame (which we discuss in Section (VI-B)) is known; thus, we know the propagation of the distortion. For a stream (s,l,r), after replacing the model in Equation (46) for the distortion D several times (according to the values of h), the distortion in Equations (47), (48), and (49) can be written as

$$\begin{aligned}
&\prod_{i=nu(l-1)+1}^{nu(l)} D_{i,(s,l,r)} \\
= &C_{(s,l,r)} \prod_{k=0}^{s} \prod_{m=0}^{l} \prod_{n=0}^{r} \prod_{i=nu(m-1)+1}^{nu(m)} \\
&\exp\{-\omega^{(s,l,r)}_{i,(k,m,n)} \gamma_{i,(k,m,n)} b_{i,(k,m,n)}\} \\
= &C_{(s,l,r)} \\
&\exp\{\sum_{k=0}^{s} \sum_{m=0}^{l} \sum_{n=0}^{r} \sum_{i=nu(l-1)+1}^{nu(l)} -\omega^{(s,l,r)}_{i,(k,m,n)} \gamma_{i,(k,m,n)} b_{i,(k,m,n)}\},
\end{aligned} \tag{50}$$

where $C_{(s,l,r)}$ is a constant, and $\omega^{(s,l,r)}_{i,(k,m,n)}$ is an integer number indicating how many times the distortion $D_{i,(k,m,n)}$ is used to derive the distortion of the stream (s,l,r) in the error propagation process. Based on the results, we can derive the average distortion of SVC.

V. The Average Distortion of SVC

Our objective is to determine the optimal bit assignment that will minimize the average distortion function in Equation (11). To achieve the objective, we need to represent the average distortion as a function of the bits assigned to an individual stream. Substituting Equation (43) into Equation (11), we have

$$\begin{aligned}
&\prod_{r\in R} \prod_{s\in S} \prod_{t=0}^{|T|-1} (\prod_{l=0}^{t} \prod_{i=nu(l-1)+1}^{nu(l)} D_{i,(s,l,r)})^{\frac{\mu_{[s,t,r]}}{2^t n_0}} \\
= &\prod_{r\in R} \prod_{s\in S} \prod_{l=0}^{|T|-1} (\prod_{i=nu(l-1)+1}^{nu(l)} D_{i,(s,l,r)})^{\sum_{t=l}^{|T|-1} \frac{\mu_{[s,t,r]}}{2^t n_0}}.
\end{aligned} \tag{51}$$

Taking the minus logarithm of Equation (51), we have

$$-\sum_{r\in R} \sum_{s\in S} \sum_{l=0}^{|T|-1} \sum_{t=l}^{|T|-1} \log((\prod_{i=nu(l-1)+1}^{nu(l)} D_{i,(s,l,r)})^{\frac{\mu_{[s,t,r]}}{2^t n_0}}). \tag{52}$$

Substituting Equation (50) into Equation (52), the minus logarithm of the average distortion is calculated as

$$\begin{aligned}
&\sum_{r\in R} \sum_{s\in S} \sum_{l=0}^{|T|-1} \sum_{t=l}^{|T|-1} \frac{\mu_{[s,t,r]}}{2^t n_0} \log(\prod_{i=nu(l-1)+1}^{nu(l)} D_{i,(s,l,r)}) \\
= &\sum_{r\in R} \sum_{s\in S} \sum_{l=0}^{|T|-1} \sum_{t=l}^{|T|-1} \frac{\mu_{[s,t,r]}}{2^t n_0} \log(C_{(s,l,r)} \\
&\exp\{\sum_{k=0}^{s} \sum_{m=0}^{l} \sum_{n=0}^{r} \sum_{i=nu(l-1)+1}^{nu(l)} \\
&-\omega^{(s,l,r)}_{i,(k,m,n)} \gamma_{i,(k,m,n)} b_{i,(k,m,n)}\}) \\
= &\sum_{r\in R} \sum_{s\in S} \sum_{l=0}^{|T|-1} \sum_{t=l}^{|T|-1} \frac{\mu_{[s,t,r]}}{2^t n_0} \{\log(C_{(s,l,r)}) + \\
&\sum_{k=0}^{s} \sum_{m=0}^{l} \sum_{n=0}^{r} \sum_{i=nu(l-1)+1}^{nu(l)} \\
&-\omega^{(s,l,r)}_{i,(k,m,n)} \gamma_{i,(k,m,n)} b_{i,(k,m,n)})\} \\
= &\sum_{r\in R} \sum_{s\in S} \sum_{l=0}^{|T|-1} \sum_{t=l}^{|T|-1} \frac{\mu_{[s,t,r]}}{2^t n_0} \log C_{(s,l,r)} \\
+ &\sum_{r\in R} \sum_{s\in S} \sum_{l=0}^{|T|-1} \sum_{t=l}^{|T|-1} \sum_{k=0}^{s} \sum_{m=0}^{l} \sum_{n=0}^{r} \sum_{i=nu(l-1)+1}^{nu(l)} \\
&-\frac{\mu_{[s,t,r]}}{2^t n_0} \omega^{(s,l,r)}_{i,(k,m,n)} \gamma_{i,(k,m,n)} b_{i,(k,m,n)}.
\end{aligned} \tag{53}$$

The first and second terms in Equation (53), denoted by f_1 and f_2 respectively, can be deduced as follows.

1. The first term:

$$f_1 = \sum_{r\in R} \sum_{s\in S} \sum_{l=0}^{|T|-1} \sum_{t=l}^{|T|-1} \frac{\mu_{[s,t,r]}}{2^t n_0} \log C_{(s,l,r)} \tag{54}$$

The term represents the constant of the objective function, and is not considered in the bit allocation problem.

2. The second term:

$$\begin{aligned} f_2 &= \sum_{r\in R}\sum_{s\in S}\sum_{l=0}^{|T|-1}\sum_{t=l}^{|T|-1}\sum_{k=0}^{s}\sum_{m=0}^{l}\sum_{n=0}^{r}\sum_{i=nu(l-1)+1}^{nu(l)} \\ &\quad -\frac{\mu_{[s,t,r]}}{2^t n_0}\omega^{(s,l,r)}_{i,(k,m,n)}\gamma_{i,(k,m,n)}b_{i,(k,m,n)} \qquad (55) \\ &= \sum_{r\in R}\sum_{s\in S}\sum_{l=0}^{|T|-1}\sum_{i=nu(l-1)+1}^{nu(l)} \\ &\quad \omega_{i,(s,l,r)}\gamma_{i,(s,l,r)}b_{i,(s,l,r)} \qquad (56) \\ &= \sum_{r\in R}\sum_{s\in S}\sum_{l=0}^{|T|-1}\omega_{(s,l,r)}b_{(s,l,r)}, \qquad (57) \end{aligned}$$

where $\omega_{i,(s,l,r)}$ denotes the weight of $b_{i,(s,l,r)}$, which can be calculated by reordering the summations in Equation (55). The value of $\omega_{(s,l,r)}$ is the weight of the rate allocated to the stream (s,l,r) and can be computed by

$$\omega_{(s,l,r)} = \frac{\sum_{i=nu(l-1)+1}^{nu(l)}\omega_{i,(s,l,r)}\gamma_{i,(s,l,r)}b_{i,(s,l,r)}}{b_{(s,l,r)}}. \qquad (58)$$

The above derivations and Equation (11) show that maximizing the average *PSNR* of H.264/SVC with a coarse-grain quality prediction structure can be approximated by maximizing

$$\sum_{r\in R}\sum_{s\in S}\sum_{l=0}^{|T|-1}\omega_{(s,l,r)}b_{(s,l,r)}. \qquad (59)$$

VI. Solving the Inter-layer Bit Allocation Problem

Recall that $b_{(k,l,j)}$ represents the number of bits assigned to all the $2^{l-1}n_0$ frames in the stream (k,l,j), Thus, Equation (59) represents the inter-layer bit allocation problem in SVC. It is difficult to solve this equation by a direct approach because a weight contains h vectors whose values depend on the results of the bit assignment process. Hence, we solve the problem by finding the optimal bit assignment of a given weight profile instead (see Section VI-A), and then modify the profile based on the derived bit assignment. The proposed optimal bit allocation algorithm is described in Section VI-B.

A. Optimal Bit Allocation with Fixed Weights

When the weight in Equation (59) is given, finding the optimal bit allocation becomes a constrained linear programming problem. To simplify the formula, we let $i = r\times|S|\times|T| + t\times|S| + s$. The bit allocation problem involves finding the set of bits $\{b_i | 0 \le i < |R|\times|S|\times|T|\}$ that solve the problem (P):

$$(P) \qquad \max_{b_i} \sum_{i=0}^{|R|\times|S|\times|T|-1} b_i w_i, \qquad (60)$$

with the constraints

$$\left\{\begin{array}{ll} b_i \le B_i, & \forall i \\ \sum_{i=0}^{|R|\times|S|\times|T|-1} b_i \le C, & \end{array}\right. \qquad (61)$$

where B_i is the rate constraint for layer i, and $C = \sum_i B_i$ is the maximal rate allowed for encoding the GOP. Note that C and B_i are given values that depend on the user's bandwidth.

The *Lagrangian* corresponding to the minimization problem (P) is defined as

$$\begin{aligned} &\mathcal{L}(\xi,\lambda_i,b_i) \\ &= \min_{b_i} -\sum_i b_i w_i - \sum_i \lambda_i(B_i - b_i) - \xi(C - \sum_i b_i), \end{aligned} \qquad (62)$$

where ξ and λ_i are called Lagrange multipliers. From the *Lagrangian*, we have

$$\max_{\xi\ge 0,\lambda_i\ge 0}\mathcal{L}(\xi,\lambda_i,b_i) = \left\{\begin{array}{ll} -\sum_i w_i b_i & \text{if } B_i \ge b_i \text{ and } C \ge \sum_i b_i, \\ \infty & \text{otherwise.}\end{array}\right. \qquad (63)$$

Therefore, the solution of

$$\min_{b_i}\max_{\xi\ge 0,\lambda_i\ge 0}\mathcal{L}(\xi,\lambda_i,b_i) \qquad (64)$$

coincides with (P) in regions where $b_i \le B_i$ and $\sum_i b_i \le C$. The duality replaces "min" and "max" in the above equation, resulting in

$$d^* = \max_{\xi\ge 0,\lambda_i\ge 0}\min_{b_i}\mathcal{L}(\xi,\lambda_i,b_i) \le \min_{b_i}\max_{\xi\ge 0,\lambda_i\ge 0}\mathcal{L}(\xi,\lambda_i,b_i) = p^*. \qquad (65)$$

Since $\mathcal{L}(\lambda,\lambda_i,b_i)$ is a linear function, and therefore a convex function, and the problem (P) has a strictly feasible solution with $b_i < B_i$ and $\sum_i b_i < C$, according to Slater's theorem, we have $p^* = d^*$. Thus, the solution of $\max_{\lambda\ge 0,\lambda_i\ge 0}\min_{b_i}\mathcal{L}(\xi,\lambda_i,b_i)$ is the solution of (P), where $\min_{b_i}\mathcal{L}(\xi,\lambda_i,b_i)$ is called the dual function.

Let us define the vectors $\mathbf{a} = (w_0, w_1, \cdots, w_{|R|\times|S|\times|T|-1})^T$, $\mathbf{b} = (b_0, b_1, \cdots, b_{|R|\times|S|\times|T|-1})^T$, and $\lambda = (\lambda_0, \lambda_1, \cdots, \lambda_{|R|\times|S|\times|T|-1})^T$, where T is the transpose operation. The *Lagrangian* can be represented as

$$\mathcal{L}(\xi,\lambda_i,b_i) = -\mathbf{a}^T\mathbf{b} - \lambda^T(\mathbf{B} - \mathbf{b}) - \xi(C - \mathbf{1}^T\mathbf{b}), \qquad (66)$$

where $\mathbf{1}$ is a column vector whose element is 1. Taking the partial derivative of the *Lagrangian* with respect to $\mathbf{b}$ and ξ, we obtain

$$\frac{\partial}{\partial\mathbf{b}}\mathcal{L}(\xi,\lambda_i,b_i) = -\mathbf{a} + \lambda + \xi\mathbf{1} = \mathbf{0}, \qquad (67)$$

and

$$\frac{\partial}{\partial\xi}\mathcal{L}(\xi,\lambda_i,b_i) = C - \mathbf{1}^T\mathbf{b} = 0, \qquad (68)$$

respectively. Solving Equations (67) and (68), we have $\mathbf{1}^T b = C$ and $\lambda_i + \xi = w_i$ with $0 \le \lambda_i \le w_i, 0 \le \xi \le w_i$. If we let x^* be $\min_i w_i$ and $\lambda_i^* = w_i - x^*$, then we have two sets with $\lambda = 0$ and $\lambda > 0$ as follows:

$$S_{\lambda=0} = \{i|\lambda_i^* = 0\}; \;\; and \;\; S_{\lambda>0} = \{i|\lambda_i^* > 0\}.$$

Substituting the results of Equations (67) and (68) into the *Lagrangian*, the minimization of the dual function in the regions where $b_i \le B_i$ and $\sum_i b_i \le C$ can be written as

$$\min_{b_i\le B_i,\sum_i b_i = C}\mathcal{L}(\xi,\lambda_i^*,b_i), \qquad (69)$$

such that

$$\begin{aligned}&\mathcal{L}(\xi,\lambda_i^*,b_i)\\&=-\sum_{i\in S_{\lambda>0}}\{w_ib_i+\lambda_i^*(B_i-b_i)\}-\sum_{i\in S_{\lambda=0}}w_ib_i \qquad (70)\\&=-\sum_{i\in S_{\lambda>0}}(w_iB_i-x^*(B_i-b_i))-\sum_{i\in S_{\lambda=0}}w_ib_i. \qquad (71)\end{aligned}$$

Equation (71) is derived by substituting $\lambda_i^*=w_i-x^*$ into Equation (70). To minimize Equation (71), we choose $b_i^*=B_i$ for for $i\in S_{\lambda>0}$. We can then derive that

$$\min_{b_i\le B_i,\sum_i b_i=C}\mathcal{L}(\xi,\lambda_i^*,b_i)=-\sum_{i\in S_{\lambda>0}}w_iB_i-\sum_{i\in S_{\lambda=0}}w_ib_i, \qquad (72)$$

and

$$\sum_{i\in S_{\lambda>0}}B_i+\sum_{i\in S_{\lambda=0}}b_i=C. \qquad (73)$$

To minimize Equation (72), we find the optimal solution of

$$\max_{b_i}\sum_{i\in S_{\lambda=0}}w_ib_i \qquad (74)$$

According to the Cauchy Schwarz inequality, the maximum occurs when $b_i=\epsilon w_i$. Thus, we have

$$\sum_{i\in S_{\lambda=0}}b_i=\epsilon\sum_{i\in S_{\lambda=0}}w_i=C-\sum_{i\in S_{\lambda>0}}B_i. \qquad (75)$$

The value of ϵ can be computed as $\epsilon=\frac{C-\sum_{i\in S_{\lambda>0}}B_i}{\sum_{i\in S_{\lambda=0}}w_i}$. We conclude that the optimal bit allocation b_i^* for a given weight profile is

$$b_i^*=\begin{cases}(C-\sum_{k\in S_{\lambda>0}}B_k)\frac{w_i}{\sum_{k\in S_{\lambda=0}}w_k} & i\in S_{\lambda=0},\\ B_i & i\in S_{\lambda>0}.\end{cases} \qquad (76)$$

B. Optimal Bit Allocation Algorithm with Known Preferences

In SVC, the optimal allocation rate is usually controlled by the quantization parameter (QP), instead of the bit rate. The QP and bit rate can be related by $QP=a\ln(b)+c$, where a and c are the model's parameters and b is the bit rate [14]; however, we calculate the actual number of bits that correspond to a QP-value when encoding a frame. Thus, the proposed algorithm allocates the optimal bit rate to each layer of quality resolution r.

The steps of the algorithm are detailed in Table 1. First, to ensure that each stream is allocated the smallest number of bits, we assign the largest possible QP value, say 51, to each stream. From the QP values, we can determine the values of h, the scaling factor κ, and γ in the encoding process; and from those values, we can derive the weight of a stream. We then partition the streams into two sets: $S_{\lambda>0}$ and $S_{\lambda=0}$. For each stream that is not assigned an optimal bit rate (see Equation (76)), we reduce its QP value by an amount approximately equal to the increase in its bit rate. Then, based on the new QP values, we repeat the above process until every stream has been assigned an optimal bit rate. In Table 1, the set F contains the streams that have optimal bit assignments.

VII. Implementation Issues and Experimental Results

In this section, we consider some implementation issues and compare the performance of QP selection by our method and the method proposed in [7], [8].

A. Coding structure and implementation details

Our coding structure, which is a modification of that in [4], supports the selection of QP values during the encoding process, as shown in Figure 2. The implementation is based on H.264/SVC JSVM 9.18. The steps of the encoding process are as follows. The encoder processes one GOP at a time. Before selecting the QP values used in the quantization operation, the modes and motion vectors of the macroblocks in a GOP are computed by the H.264/SVC JSVM 9.18. The $MQPs$, defined as the QP values used to compute the modes and motion vectors, are obtained from the QP values of the previous GOP. For the first GOP, the MQP is set at 28 for all streams. After obtaining the modes and motion vectors, the inter-layer bit allocation method decides the QP values for all streams. Then, given the modes, motion vectors and QP values, the SVC encoder generates bit streams for each (s,l,r). Finally, the multiplexer combines the generated streams. The implementation details are as follows: the size of the GOP is 4; the inter-layer prediction option is enabled; a full search is performed for motion estimation; the motion vector accuracy is $1/4$ of a pixel; the search range is 32×32; the variable block size option is enabled; and a hierarchical prediction structure is used for temporal scalability.

B. Variance approximation

In this subsection, we examine the approximation of the variance in Equation (3) by Equations (39), (40), and (41). At each rate-distortion point, the calculations of the γ value, the h vector, and the scaling factor κ, which adjusts the variance of the dominating term in h to the actual variance, are based on the actual rate, distortion, and variance. Then the method to compute the distortion of each rate-distortion point is similar to that in our rate allocation algorithm. Given the approximated variance and the value of γ, we can predict the distortion of the adjacent rate-distortion point with a higher rate according to the modeling in Equation (3) and compare it with the distortion measured by the actual encoding process.

First, we examine the approximated variance in Equation (39). In the experiment, we let $S=\{352\times 288\}$, $T=\{7.5fps,15fps\}$, and $R=\{r_1\}$ (indicating only one quality layer). From the graphs in the bottom row of Figure 3, we observe that if the quantization step of the predicting stream is large, ($(352\times288,7.5fps,r_1)$ in this case), corresponding to a low bit rate, the propagated distortion dominates the variance calculation so that our modeling is very accurate. In contrast, when the quantization step of the predicting stream is small, corresponding to a high bit rate, the constant term in Equation (39) dominates the variance, as shown in the top row of Figure 3. However, if neither of the terms in Equation (39) dominates the process, our modeling is less accurate, as shown in the middle row of Figure 3.

In Figure 4, we examine the validation of Equation (40), which approximates the variance after spatial prediction and temporal prediction. In the experiment, we let $S = \{176 \times 144, 352 \times 288\}$, $T = \{30fps\}$, and $R = \{r_1\}$. For ease of presentation, we assume that if the streams have the same spatial resolution, they also have the same QP value. As shown in the bottom row of Figure 4, when the quantization step of the spatial predicting stream is large, $((176 \times 144, 30fps, r_1)$ in this case), the propagated distortion dominates the variance calculation, and our rate distortion modeling is very accurate. In contrast, when the quantization step of the spatial predicting stream is small, the constant term or the temporally propagated distortion may dominate the variance, as shown in the top row of Figure 4, and our approximation results are satisfactory. However, if none of the three terms in Equation (40) dominates the process, our modeling is less accurate, as shown in the middle row of Figure 4.

In Figure 5, we examine the validation of Equation (41), which approximates the variance of a frame after quality prediction and temporal prediction. In the experiment, we let $S = \{352 \times 288\}$, $T = \{30fps\}$, and $R = \{r_1, r_2\}$ (corresponding to using two quality layers). Again, for ease of presentation, we assume that if the streams have the same quality resolution, they also have the same QP value. The results depicted in the bottom row of Figure 5 show that when the quantization step of the quality predicting stream is large $((352 \times 288, 30fps, r_1)$ in the case), the propagated distortion dominates the variance calculation, and the rate distortion curves can be well approximated. However, when the quantization step of the quality predicting stream becomes small, the constant term or the temporally propagated distortion may dominate the variance, as shown in the top row of Figure 5. If none of the terms in Equation (41) dominates the process, our variance calculation model becomes less accurate, as shown in the middle row of Figure 5.

The results in Figures 3, 4, and 5 show that when the distortion of the dominating term is not significantly larger than that of the other terms, the variances are not well approximated by (39), (40), and (41). A precise distortion model could be obtained by setting all the values of h as 1; however, the analysis of the error propagation for such a model would be overwhelmingly complex. Deriving a more precise variance model whose error propagation can be analyzed easily would be an interesting topic for future research.

C. Performance comparison

In this subsection, we compare the coding efficiency of different QP selection schemes. We denote our bit allocation method (discussed in Section VI-B) as *Proposed*, and compare its performance with that of the state-of-the-art Lagrangian-based method (denoted as *Lagrangian*) proposed in [8], [7]. The latter uses the weighting of each stream to indicate the importance of the resolution in deriving the optimal bit-allocation; however, the authors do not explain how the weighting values are selected.

The *Lagrangian* method selects the QP values by minimizing

$$J = \sum_{s \in S, t \in T, r \in R} w_{[s,t,r]} \{ \sum_{l=0}^{t} \sum_{i=nu(l-1)+1}^{nu(l)} J_{i,(s,l,r)} \}, \tag{77}$$

in which $w_{[s,t,r]}$ is the weighting of the resolution $[s,t,r]$. Here, $J_{i,(s,l,r)}$ denotes the objective function of the frame i in the stream (s,l,r). It is formulated as follows:

$$J_{i,(s,l,r)} = \mathit{(SSD)}_{i,(s,l,r)} + \lambda_{i,(s,l,r)} b_{i,(s,l,r)}, \tag{78}$$

where *SSD* denotes the sum of the squared differences. According to the analysis in [15], if we assign the same QP values (denoted by $QP_{(s,l,r)}$) to all the frames in the stream (s,l,r), the value of λ in Equation (78) will be $0.85 \times 2^{(QP_{(s,l,r)}-12)/3}$. In our comparison, we let $w_{[s,t,r]}$ be $\mu_{[s,t,r]}$ because both of them are supposed to indicate the importance of the resolution in the bit-allocation process. Thus, we compare the optimization with the following objective function:

$$J = \sum_{s \in S, t \in T, r \in R} \mu_{[s,t,r]} \{ \sum_{l=0}^{t} \sum_{i=nu(l-1)+1}^{nu(l)} J_{i,(s,l,r)} \}. \tag{79}$$

In the following experiments, we measure the performance by averaging the coding gain of *Proposed* over *Lagrangian* on four sequences: Foreman, News, Dancer and Coastguard. First, we compare the user preference profiles assigned to different temporal resolutions. We let $S = \{88 \times 72\}$, $T = \{7.5fps, 15fps, 30fps\}$, and $R = \{r_1\}$. Various values are given to the three preferences, $\mu_{[88\times72,7.5fps,r_1]}$, $\mu_{[88\times72,15fps,r_1]}$, and $\mu_{[88\times76,30fps,r_1]}$ so that their sum is equal to 1. We conduct experiments on three rate constraints, namely, $40kbps$, $60kbps$ and $80kbps$ (corresponding to C in Equation (61)). The average *PSNR* gain of *Proposed* over *Lagrangian* is shown in Figure 6. In addition, as shown in Figure 7, we conduct experiments with the same settings as Figure 6, except that the spatial resolution $S = \{352 \times 288\}$ and the rate constraints are $80kbps$, $120kbps$, and $160kbps$. From Figures 6 and 7, we observe that *Proposed* outperforms *Lagrangian*. The average *PSNR* gain of *Proposed* over *Lagrangian* for temporal scalability is 0.25db in Figure 6 and 0.06db in Figure 7.

Next, we compare the rate allocation schemes in terms of different spatial resolutions. In the experiment, we let $S = \{88 \times 72, 176 \times 144, 352 \times 288\}$, $T = \{7.5fps, 15fps, 30fps\}$, and $R = \{r_1\}$. Various values are given to $\mu_{[88\times72,30fps,r_1]}$, $\mu_{[176\times144,30fps,r_1]}$, and $\mu_{[352\times288,30fps,r_1]}$ so that their sum is equal to 1. The average *PSNR* gain of *Proposed* over *Lagrangian* is shown in Figure 8. In addition, as shown in Figure 9, we conduct the experiments with the same settings as Figure 8, except that the spatial resolution $S = \{176 \times 144, 352 \times 288, 704 \times 576\}$, and the rate constraints become $320kbps$, $640kbps$, and $1280kbps$. For all user preference distributions, the average *PSNR* gain of *Proposed* over *Lagrangian* is 1.38db in Figure 8 and 0.79db in Figure 9.

The coding gains in Figures 8 and 9 are much larger than those in Figures 6 and 7. Thus, under our method, the coding gain for spatial scalability is higher than that for temporal scalability. From Figures 8 and 9, we observe that the coding gain depends on the distribution of the user preferences. If most users prefer a lower spatial resolution, the coding gain may be larger than 8db; conversely, if most users prefer a higher spatial resolution, the coding gain may be less than 0.5db.

We also compare the rate allocation schemes in terms of different quality resolutions. We let $S = \{88 \times 72\}$, $T = \{7.5fps, 15fps, 30fps\}$, and $R = \{r_1, r_2, r_3\}$ (three quality layers); and assign various values to $\mu_{[88\times72,30fps,r_1]}$, $\mu_{[88\times72,30fps,r_2]}$, and $\mu_{[88\times72,30fps,r_3]}$ so that the sum of their preferences is equal to 1. The average *PSNR* gain of *Proposed* over *Lagrangian* is shown in Figure 10. In addition, as shown in Figure 11, we conduct experiments with the same settings as Figure 10, except that the spatial resolution $S = \{352 \times 288\}$ and the rate constraints are $80kbps$, $120kbps$, and $160kbps$. For all user preference distributions, the average *PSNR* gain of *Proposed* over *Lagrangian* is 0.2db in Figure 10 and 0.05db in Figure 11.

Finally, we consider the complexity of the three coding schemes. As shown in Figure 2, motion estimation and mode selection, which are the most time consuming parts of the encoding process in different coding schemes, are only performed once for each macroblock in a *GOP*. The optimal bit-allocation process of all the coding schemes has the same computational complexity order $\mathcal{O}(\mathcal{M}\mathcal{N})$, where M represents the number of streams and N denotes the possible QP values for each stream. If the modes and motion vectors are given, motion compensation and quantization can be executed efficiently in all the schemes. In our experiments, at a high coding rate, the computation time of the *Proposed* method is about 3 times longer than that of the JSVM encoder; however, at a low coding rate, it is only 1.5 times longer.

VIII. Conclusion

We present a theoretical analysis of joint R-D optimization for mult-layer coding. The data dependency structure of temporal, spatial, and quality prediction is fully explored in the analysis. In addition, we demonstrate the importance of the end user's preference to the coding performance of SVC. We derive that the average *PSNR* of SVC is the weighted average of the bit rates assigned to individual streams. The weighting factor is a function of all the affected layers and their corresponding preference factors. We also propose an optimal bit allocation algorithm that controls the encoder rate with subscribers' preference information. Comparison of the algorithm's performance with that of a state-of-the-art coder shows that it achieves a significant *PSNR* gain over the compared method. In a future work, we will extend our analysis to study the joint source and channel coding problem under SVC.

Acknowledgments

The authors wish to thank the reviewers and the associate editor for their insightful comments, which have helped us improve the quality of the paper significantly.

Table 1. The proposed optimal bit allocation algorithm with known preference information

```
(1) Let $QP_i = 51$ for each stream and let $F = \{\}$.
(2) Run SVC to obtain the actual number of bits used
in the current $QP$ assignment.
(3) Derive the values of $h$, $\kappa$, and $\gamma$ from $QP$,
and compute the weight $w_i$.
(4) Based on the weight, assign a stream to either
$S_{\lambda>0}$ or $S_{\lambda=0}$.
(5) If all the streams are in $F$, the algorithm stops.
    Otherwise, for a stream $i$ not in $F$, there are two
possibilities: $i \in S_{\lambda>0}$ or $i \in S_{\lambda=0}$.
(6) Case $S_{\lambda>0}$:
(6.1) Let $w_i = \max_{k\in S_{\lambda>0}, k\notin F} w_k$; that is, the
coding error of the stream $i$ has the largest weight.
    If $QP_i = 1$, the stream has the largest bit assignment
and it is added to $F$.
    Otherwise, increase the number of bits assigned to
the stream by reducing its $QP$ value;
    $QP_i = \max\{QP_i - 1, 1\}$.
(6.2) Run SVC to obtain the actual number of bits for
the new value of $QP$. If $b_i > B_i$,
    let $QP_i = QP_i + 1$, add stream $i$ to $F$, and go to
step 5; otherwise, go to step 3.
(7) Case $S_{\lambda=0}$:
(7.1) Reduce $QP_i$ until $b_i > \frac{C - \sum_{k\in S_{\lambda>0}} B_k}{\sum_{k\in S_{\lambda=0}} w_k} w_i$,
or $b_i > B_i$, or $QP_i = 0$.
    Then, add $i$ to $F$ and let $QP_i = QP_i + 1$;
and go to step 3.
```

Appendix 1: Variance approximation

In this appendix, we provide a simple example to illustrate how the dependency of frames in video coding can affect the rate-distortion analysis significantly. In the following, the texture means the residual obtained after predicting a frame. We consider two cases and use three frames (frames 1, 2, and 3) to demonstrate the benefits derived by re-writing Equation (24) as Equation (25).

Case 1: *Independent* rate distortion curves

Assuming the three frames are encoded separately with the variances of the texture σ_1^2, σ_2^2, σ_2^2 and the model coefficients γ_1 and γ_2 and γ_3 respectively (according to Equation (3)), the optimal rate allocation problem involves minimizing the following equation:

$$D_1(b_1)D_2(b_2)D_3(b_3)$$

$$= \sigma_1^2 \exp(-\gamma_1 b_1)\sigma_2^2 \exp(-\gamma_2 b_2)\sigma_3^2 \exp(-\gamma_3 b_3) \quad (80a)$$

$$= \sigma_1^2\sigma_2^2\sigma_3^2 \exp(-(\gamma_1 b_1 + \gamma_2 b_2 + \gamma_3 b_3)). \quad (80b)$$

Thus, after taking the logarithm on both sides of the equation, we obtain the linear relationship between the log-distortion and the bit rate allocated to each frame. As a result, the optimal rate allocation solution can be derived efficiently by using linear programming methods.

Case 2: *Dependent* rate distortion curves

In this case, we show how the dependency affects the rate allocation results based on the ρ-domain source model. We assume that frame 2 is temporally predicted by frame 1 and frame 3. Because the texture of frame 2 is dependent on the reconstructed referred (predicting) frames in the prediction steps, σ_2^2 should be a function of the rates allocated to frame 1 and frame 3. The optimal rate allocation problem involves minimizing the following equation:

$$\begin{aligned} & D_1(b_1)D_2(b_1,b_2,b_3)D_3(b_3) \\ &= \sigma_1^2 \exp\left(-\gamma_1 b_1\right)\sigma_2^2(b_1,b_3)\exp\left(-\gamma_2 b_2\right)\sigma_3^2 \exp\left(-\gamma_3 b_3\right) \quad \text{(81a)} \\ &= \sigma_1^2\sigma_2^2(b_1,b_3)\sigma_3^2 \exp\left(-(\gamma_1 b_1 + \gamma_2 b_2 + \gamma_3 b_3)\right), \quad \text{(81b)} \end{aligned}$$

where $\sigma_2^2(b_1,b_3)$ implies that the variance of frame 2 is dependent on the rates b_1 and b_3. The detailed derivation of the effects of b_1 and b_3 on σ_2^2 is given in Appendix 2. Using the result of Appendix 2, we have

$$\sigma_2^2(b_1,b_3) \;=\; h_2^T \begin{pmatrix} var(C_2) \\ \{C_1^\alpha C_3^\beta D_1(b_1)D_3(b_3)\}^{\frac{1}{2}} \end{pmatrix}, \quad (82)$$

where $var(C_2)$, C_1^α, and C_3^α are constants, and $D_1(b_1)$ and $D_3(b_3)$ represent the distortion of frame 1 and frame 3 respectively. Let h_2^1 and h_2^2 be the first and second components of column vector h_2. Equation (82) can be written as

$$\sigma_2^2(b_1,b_3) = h_2^1 var(C_2) + h_2^2\{C_1^\alpha C_3^\beta D_1(b_1)D_3(b_3)\}^{\frac{1}{2}}. \quad (83)$$

The ρ-domain source model is used for the distortions $D_1(b_1)$ and $D_3(b_3)$ in Equation (83). We obtain

$$\begin{aligned} & D_1(b_1)D_2(b_1,b_2,b_3)D_3(b_3) \\ &= \sigma_1^2\sigma_3^2(h_2^1 var(C_2) + h_2^2 A(b_1,b_2,b_3)), \quad (84) \end{aligned}$$

where

$$\begin{aligned} & A(b_1,b_2,b_3) = \\ & \{C_1^\alpha C_3^\beta \sigma_1^2 \exp\left(-\gamma_1 b_1\right)\sigma_3^2 \\ & \exp\left(-\gamma_3 b_3\right)\}^{\frac{1}{2}} \exp\left(-(\gamma_1 b_1 + \gamma_2 b_2 + \gamma_3 b_3)\right). \quad (85) \end{aligned}$$

Note that, in Equation (84), the two leading terms begin with h_2^1 and h_2^2 respectively. Thus, *we cannot obtain the simple linear relationship between the log-distortion and the rates allocated to frames by taking the logarithm on both sides of the equation.* This example shows that the complexity of optimal rate-allocation analysis of the three dependent frames can increase. If more frames are involved, the distortion may be comprised of several terms; hence, the rate-allocation analysis would be even more complicated. Moreover, if we consider spatial, temporal, and quality dependency simultaneously, as in H.264/SVC, the optimal rate allocation problem would become overwhelmingly complicated and impossible to solve efficiently.

The above analysis explains why we only allow the vector h in Equations (39),(40), and (41) to have one non-zero component (i.e., dominating component). As a result, the variance of each predicted frame is comprised of only one term, so we can maintain the simple linear relationship between the log-distortion and the bit rate in the analysis. In our example, σ_2^2 is approximated as either $\kappa_1 var(C_2)$ or $\kappa_2\{C_1^\alpha C_3^\beta \sigma_1^2 \exp\left(-\gamma_1 b_1\right)\sigma_3^2 \exp\left(-\gamma_3 b_3\right)\}^{\frac{1}{2}}$, not as a linear combination of them, where κ_1 and κ_2 are scaling factors that adjust the variance of the respective dominating terms to the actual variance. Depending on which term in h is the dominating term, the distortion in Equation (84) becomes either

$$D_1(b_1)D_2(b_1,b_2,b_3)D_3(b_3) = \kappa_1\sigma_1^2\sigma_3^2 var(C_2), \quad (86)$$

or

$$D_1(b_1)D_2(b_1,b_2,b_3)D_3(b_3) = \kappa_2\sigma_1^2\sigma_3^2 A(b_1,b_2,b_3), \quad (87)$$

where $A(b_1,b_2,b_3)$ is given in Equation (85). Thus, our approach can preserve the simple linear relationship between the log-distortion and the allocated bit rates of dependent frames. Using a simpler R-D analysis of distortion propagation from referred frames/layers to referring frames/layers makes the optimal rate allocation process much more straightforward.

Appendix 2: The two-stream relation of temporal prediction at a low bit rate

The two-stream relation explores the data-dependency between the predicting and predicted stream in SVC. We now derive the distortion between the two streams due to temporal prediction at a low bit rate. In the following analysis, the pixels of a frame are arranged as a vector.

For temporal prediction at a low bit rate, Equation (15) can be approximated as

$$(\Delta_{i,(0,l)})^{1,\mathbf{t}} = P_{i,i-m}\Delta_{i-m,(0,l-1,1)} + P_{i,i+m}\Delta_{i+m,(0,l-1,1)}, \quad (88)$$

where $P_{i,i-m}$ and $P_{i,i+m}$ are the matrices of the motion vectors. Without loss of generality, we assume that a pixel in $(\Delta_{i,(0,l)})^{1,\mathbf{t}}$ is estimated by a linear combination of one pixel in $\Delta_{i-m,(0,l-1,1)}$ and one pixel in $\Delta_{i+m,(0,l-1,1)}$. Thus, the p-th pixel in $(\Delta_{i,(0,l)})^{1,\mathbf{t}}$ can be written as

$$(\Delta_{i,(0,l)})^{1,\mathbf{t}}(p) = a\Delta_{i-m,(0,l-1,1)}(f_1(p)) + b\Delta_{i+m,(0,l-1,1)}(f_2(p)), \quad (89)$$

where a and b are the prediction weights of $\Delta_{i-m,(0,l-1,1)}$ and $\Delta_{i+m,(0,t-1,1)}$ respectively; and $f_1(p)$ and $f_2(p)$ are the corresponding pixels in the predicting residuals. As usual, the pixels can be derived from the motion vectors. Because motion estimation finds the most similar blocks in the predicting residual in order to estimate the target block, we can assume that there are several pairs of pixels in which $\Delta_{i-m,(s,t-1,1)}(f_1(p))$ and $\Delta_{i+m,(0,t-1,1)}(f_2(p))$ have similar values; that is, for several pairs of $(f_1(p), f_2(p))$,

$$\Delta_{i-m,(0,l-1,1)}(f_1(p)) \approx \Delta_{i+m,(0,l-1,1)}(f_2(p)). \quad (90)$$

Substituting the above result into Equation (89), we have

$$((\Delta_{i,(0,l)})^{1,\mathbf{t}})^2 \approx (a+b)^2(\Delta_{i-m,(0,l-1,1)}(f_1(p)))^2. \quad (91)$$

Let N^2 denote the number of pixels in $(\Delta_{i,(0,l)})^{1,\mathbf{t}}$. If the interpolation kernel is used, then $a+b=1$, and we have

$$\begin{aligned}
\sigma^2_{i,(0,l,1)} &= \frac{1}{N^2}\sum_{p=1}^{N^2}((\Delta_{i,(0,l)})^{1,\mathbf{t}}(p))^2 && (92)\\
&\approx \frac{(C^\alpha)^{1,\mathbf{t}}_{i-m,(0,l-1)}}{N^2}\sum_{p=1}^{N^2}(\Delta_{i-m,(0,l-1,1)}(f_1(p)))^2 && (93)\\
&= (C^\alpha)^{1,\mathbf{t}}_{i-m,(0,l-1)}D_{i-m,(0,l-1,1)}, && (94)
\end{aligned}$$

where $(C^\alpha)^{1,\mathbf{t}}_{i-m,(0,l-1)}$ is related to the proportion of the pixels used to perform forward prediction in $\Delta_{i-m,(0,l-1,r)}$:

$$\frac{\sum_{p=1}^{N^2}(\Delta_{i-m,(0,l-1,1)}(f_1(p)))^2}{\sum_{p=1}^{N^2}(\Delta_{i-m,(0,l-1,1)}(p))^2}. \qquad (95)$$

The value of $(C^\alpha)^{1,\mathbf{t}}_{i-m,(0,l-1)}$ depends on the motion vectors. It becomes a constant after the motion vectors have been obtained. For the frames of slow motion objects, almost all the pixels in $\Delta_{i-m,(0,l-1,r)}$ will be used in the prediction process; thus, $(C^\alpha)^{1,\mathbf{t}}_{i-m,(0,l-1)} \approx 1$.

In a similar way, we can derive that

$$\sigma^2_{i,(0,l,1)} \approx (C^\beta)^{1,\mathbf{t}}_{i+m,(0,l-1)}D_{i+m,(0,l-1,1)}, \qquad (96)$$

where $(C^\beta)^{1,\mathbf{t}}_{i+m,(0,l-1)}$ is calculated as follows:

$$\frac{\sum_{p=1}^{N^2}(\Delta_{i+m,(0,l-1,1)}(f_2(p)))^2}{\sum_{p=1}^{N^2}(\Delta_{i+m,(0,l-1,1)}(p))^2}. \qquad (97)$$

$(C^\beta)^{1,\mathbf{t}}_{i+m,(0,l-1)}$ is a constant after the motion vectors have been obtained. For frames that contain slow motion objects, $(C^\beta)^{1,\mathbf{t}}_{i+m,(0,l-1)} \approx 1$. Combining Equations (94) and (96), we have

$$\begin{aligned}
&\sigma^2_{i,(0,l,1)} \approx\\
&(C^\alpha)^{1,\mathbf{t}}_{i-m,(0,l-1)}(C^\beta)^{1,\mathbf{t}}_{i+m,(0,l-1)}\\
&D_{i-m,(0,l-1,r)}D_{i+m,(0,l-1,r)})^{\frac{1}{2}}, && (98)
\end{aligned}$$

which is the geometric mean of the results of Equations (94) and (96).

Appendix 3: The two-stream relation of spatial residual prediction at a low bit rate

In the following, we derive the relation between the distortion of two frames during inter-layer spatial prediction at a low bit rate. Let the size of a frame in spatial layer $s-1$ be N^2. Without loss of generality, we assume a dyadic spatial scalability structure, where the number of pixels of a frame in spatial layer identifier s is four times greater than that of the corresponding frame in spatial layer identifer $s-1$.

Let $\Delta^{r,\mathbf{s}}_{i,(s,l)}(p)$ denote the p-th pixel in $\Delta^{r,\mathbf{s}}_{i,(s,l)}$, and let $\Delta_{i,(s-1,l,1)}[p]$ denote the pixels in $\Delta_{i,(s-1,l,1)}$ involved in the spatial prediction of pixel $\Delta^{r,\mathbf{s}}_{i,(s,l)}(p)$. We use the vec operator to change the pixels in the block $\Delta^{s}_{i,(s-1,l,1)}[p]$ into a column vector U, and use the latter to represent the spatial prediction method. As shown in Equation (29), at a low bit rate and $var(U(\Delta_{i,(s-1,l,1)})) > var(P_{i,i-m}\Delta_{i-m,(s,l-1,1)} + P_{i,i+m}\Delta_{i+m,(s,l-1,1)})$, the residual $\Delta^{1,\mathbf{s}}_{i,(s,l)}(p)$ can be approximated as

$$\Delta^{1,\mathbf{s}}_{i,(s,l)}(p) \approx U^T(vec(\Delta_{i,(s-1,l,1)}[p])), \qquad (99)$$

where T is the transpose operation. Taking the square of $\Delta^{1,\mathbf{s}}_{i,(s,l)}(p)$, we can derive that

$$\begin{aligned}
(\Delta^{1,\mathbf{s}}_{i,(s,l)}(p))^2 &= trace(UU^T(vec(\Delta_{i,(s-1,l,1)}[p]))\\
&\quad (vec(\Delta_{i,(s-1,l,1)}[p]))^T) && (100)\\
&\le trace(U^TU)trace((vec(\Delta_{i,(s-1,l,1)}[p]))\\
&\quad (vec(\Delta_{i,(s-1,l,1)}[p]))^T) && (101)\\
&= trace(U^TU)\|vec(\Delta_{i,(s-1,l,1)}[p])\|_F && (102)\\
&\le \|U\|_2^2\ \|vec(\Delta_{i,(s-1,l,1)}[p])\|_F. && (103)
\end{aligned}$$

Equations (101), (102), and (103) are derived by using the properties of the Frobenius norm:

$$\|AB\|_F^2 \le \|A\|_F^2\|B\|_F^2, and\ \ \|A\|_F^2 = trace(A^TA), \qquad (104)$$

for any real matrices A and B. If U represents an interpolation, then $\|U\|_2^2 \le \|U\|_1^2 = 1$. As a result, Equation (103) becomes

$$(\Delta^{1,\mathbf{s}}_{i,(s,l)}(p))^2 \le \|vec(\Delta_{i,(s-1,l,1)}[p])\|_F. \qquad (105)$$

Let $|U|$ denote the size of the vector U. Because any pixel in $\Delta_{i,(s-1,l,1)}$ is used at most $|U|$ times in the spatial prediction process, the variance of $\Delta^{1,\mathbf{s}}_{i,(s-1,l)}$ can be calculated as

$$\begin{aligned}
\sigma^2_{i,(s,l,1)} &= \frac{1}{4N^2}\sum_{p=1}^{4N^2}(\Delta^{1,\mathbf{s}}_{i,(s-1,l)}(p))^2 && (106)\\
&\le \frac{1}{4N^2}\sum_{p=1}^{4N^2}\|vec(\Delta_{i,(s-1,l,1)}[p])\|_F^2 && (107)\\
&\le \frac{|U|}{4N^2}\|vec(\Delta_{i,(s-1,l,1)})\|_F^2 && (108)\\
&= \frac{|U|}{4}D_{i,(s-1,l,1)}. && (109)
\end{aligned}$$

Equation (109) gives the distortion between two corresponding frames in the spatial prediction process. If bilinear interpolation is used, (*i.e.*, 4 pixels are involved in the interpolation), we have $|U| = 4$. Then, we can introduce a constant $(C^\omega)^{1,\mathbf{s}}_{i,(s-1,l)}$ and re-write Equation (109) as follows:

$$\sigma^2_{i,(s,l,1)} = (C^\omega)^{1,\mathbf{s}}_{i,(s-1,l)}D_{i,(s-1,l,1)}. \qquad (110)$$

Appendix 4: The two-stream relation of quality residual prediction at a low bit rate

In the following, we derive the relation between the distortion of two frames during inter-layer quality prediction at a low bit rate. Let the size of a frame in spatial layer s be N^2. Let $\Delta^{r,\mathbf{q}}_{i,(s,l)}(p)$ denote the p-th pixel in $\Delta^{r,\mathbf{q}}_{i,(s,l)}$, and let $\Delta_{i,(s,l,r-1)}[p]$ denote the pixels in $\Delta_{i,(s,l,r-1)}$ involved in the quality prediction of pixel $\Delta^{r,\mathbf{q}}_{i,(s,l)}(p)$. We use the vec operator to change the pixels in the block $\Delta^{s}_{i,(s,l,r-1)}[p]$ into a column vector Y, and use the latter to represent the quality prediction method. As shown in Equation (29), at a low bit rate and $var(Y(\Delta_{i,(s,l,r-1)})) > var(P_{i,i-m}\Delta_{i-m,(s,l-1,r)} + P_{i,i+m}\Delta_{i+m,(s,l-1,r)})$, the residual $\Delta^{r,\mathbf{q}}_{i,(s,l)}(p)$ can be approximated as

$$\Delta^{r,\mathbf{q}}_{i,(s,l)}(p) \approx Y^T(vec(\Delta_{i,(s,l,r-1)}[p])), \tag{111}$$

where T is the transpose operation. Taking the square of $\Delta^{r,\mathbf{q}}_{i,(s,l)}(p)$, we can derive that

$$\begin{aligned}
(\Delta^{r,\mathbf{q}}_{i,(s,l)}(p))^2 &= trace(YY^T(vec(\Delta_{i,(s,l,r-1)}[p])) \\
&\quad (vec(\Delta_{i,(s,l,r-1)}[p]))^T) && (112)\\
&\leq trace(Y^TY)trace((vec(\Delta_{i,(s,l,r-1)}[p])) \\
&\quad (vec(\Delta_{i,(s,l,r-1)}[p]))^T) && (113)\\
&= trace(Y^TY)\|vec(\Delta_{i,(s,l,r-1)}[p])\|_F && (114)\\
&\leq \|Y\|^2_2\ \|vec(\Delta_{i,(s,l,r-1)}[p])\|_F. && (115)
\end{aligned}$$

Equations (113), (114), and (115) are derived by using the properties of the Frobenius norm:

$$\|AB\|^2_F \leq \|A\|^2_F\|B\|^2_F, and\ \ \|A\|^2_F = trace(A^TA), \tag{116}$$

for any real matrices A and B. Because Y represents selection of pixels, we have $\|Y\|^2_2 \leq 1$. As a result, Equation (115) becomes

$$(\Delta^{r,\mathbf{q}}_{i,(s,l)}(p))^2 \leq \|vec(\Delta_{i,(s,l,r-1)}[p])\|_F. \tag{117}$$

The variance of $\Delta^{r,\mathbf{q}}_{i,(s,l)}$ can be calculated as

$$\begin{aligned}
\sigma^2_{i,(s,l,r)} &= \frac{1}{N^2}\sum_{p=1}^{N^2}(\Delta^{r,\mathbf{q}}_{i,(s,l)}(p))^2 && (118)\\
&\leq \frac{1}{N^2}\sum_{p=1}^{N^2}\|Y\|^2_2\|vec(\Delta_{i,(s,l,r-1)}[p])\|^2_F && (119)\\
&\leq \frac{1}{N^2}\|vec(\Delta_{i,(s,l,r-1)})\|^2_F && (120)\\
&= D_{i,(s,l,r-1)}. && (121)
\end{aligned}$$

Equation (121) gives the distortion between two corresponding frames in the quality prediction process. Then, we can introduce a constant $(C^{\omega})^{r-1,\mathbf{q}}_{i,(s,l)}$ and re-write Equation (121) as follows:

$$\sigma^2_{i,(s,l,r)} = (C^{\omega})^{r-1,\mathbf{q}}_{i,(s,l)}D_{i,(s,l,r-1)}. \tag{122}$$

References

[1] P. Steenkiste, "Adaptation models for network-aware disributed computation," in *Workshop on Communication, Architecture, and Applicatios for Network-based Parallel Computing*, January 1999, pp. 16–31.

[2] B. Tierney, D. Gunter, J. Lee, M. Stoufer, and J. B. Evans, "Enabling network-aware applications," in *IEEE International Symposium on High Performance Distributed Computing*, 2001, pp. 281–288.

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

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

[5] M. Wien, H. Schwarz, and T. Oelbaum, "Performance analysis of SVC," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 12, pp. 1771–1771, 2007.

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

[7] K. Ramchandran, A. Ortega, and M. Vetterli, "Bit allocation for dependent quantization with applications to multiresolution and MPEG video coders," *IEEE Transactions on Image Processing*, vol. 3, no. 5, pp. 533–545, 1994.

[8] H. Schwarz and T. Wiegand, "R-D optimized multi-layer encoder control for SVC," in *IEEE International Conference on Image Processing*, Sepetember 2007, pp. 281–284.

[9] M. Koziri and A. Eleftheriadis, "Joint quantizer optimization for scalable coding," in *IEEE International Conference on Image Processing*, October 2010.

[10] J. Reichel, H. Schwarz, and M. Wien, "Joint Scalable Video Model 11 (jsvm 11)," *Joint Video Team, Doc, JVT-X202*, July 2007.

[11] Z. He and S. K. Mitra, "Optimum bit allocation and accurate rate control for video coding via rho domain source modeling," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 10, pp. 840–849, 2002.

[12] Q. Zhang, Q. Guo, Q. Ni, W. Zhu, and Y.-Q. Zhang, "Sender-adaptive and receiver driven layered multicast for scalable video over the internet," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 4, pp. 482–495, April 2005.

[13] H. Li, Z. Li, and C. Wen, "Fast mode decision algorithm for inter-frame coding in fully scalable video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 16, no. 7, pp. 889 –895, july 2006.

[14] L. Czúni, G. Császár, and A. Licsár, "Estimating the optimal quantization parameter in H.264," in *IEEE International Conference on Pattern Recognition*, August 2006, pp. 330–333.

[15] T. Wiegrand, H. Schwarz, A. Joch, F.Kossentini, and G. J. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 7, July 2003.

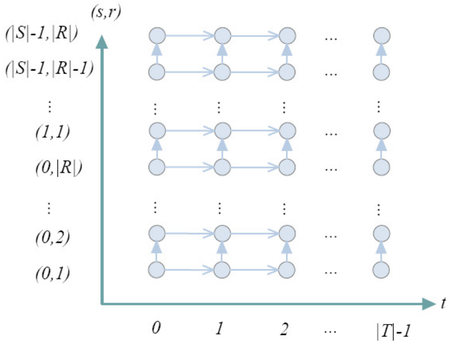

Fig. 1. The data dependency structure for coarse-grain quality prediction of H.264/SVC.

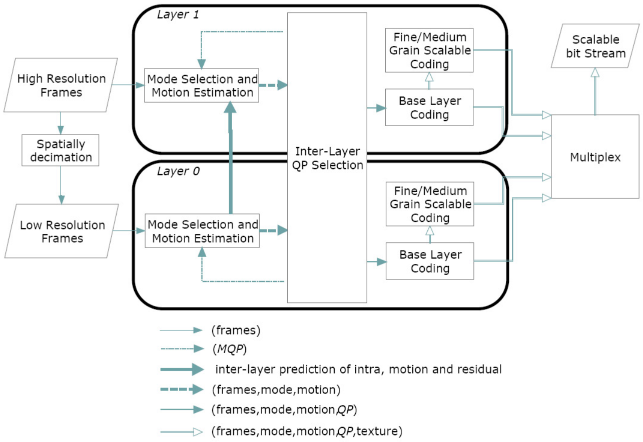
Layer 1
High Resolution Frames
Mode Selection and Motion Estimation
Inter-Layer QP Selection
Fine/Medium Grain Scalable Coding
Base Layer Coding
Scalable bit Stream
Multiplex
Spatially decimation
Layer 0
Low Resolution Frames
Mode Selection and Motion Estimation
Fine/Medium Grain Scalable Coding
Base Layer Coding
(frames)
(MQP)
inter-layer prediction of intra, motion and residual
(frames,mode,motion)
(frames,mode,motion,QP)
(frames,mode,motion,QP,texture)

Fig. 2. The proposed encoding structure supports the combined scalability. In our implementation, motion estimation and mode selection are based on JSVM 9.18. Each layer performs its own motion estimation and mode selection. Note that if the inter-layer QP selection step is removed, then the coding structure is exactly the same as that of H.264/SVC. MQP is defined as the QP values used to derive the modes and the motion vectors of macroblocks. The model parameters for Inter-Layer QP Selection are h, κ, and γ. They are updated in each iteration of our optimal bit allocation algorithm.

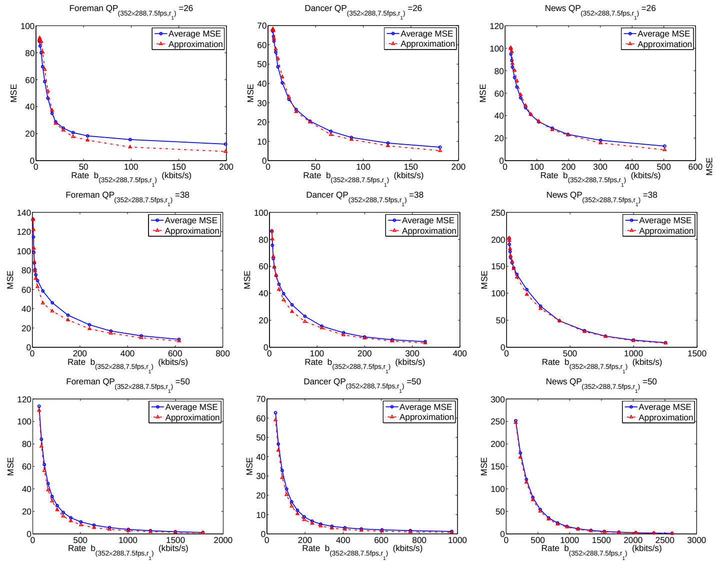

Fig. 3. Approximation of the distortion of the predicted layer due to temporal prediction based on Equation (39). The prediction accuracy depends on the QP values of the predicting layer.

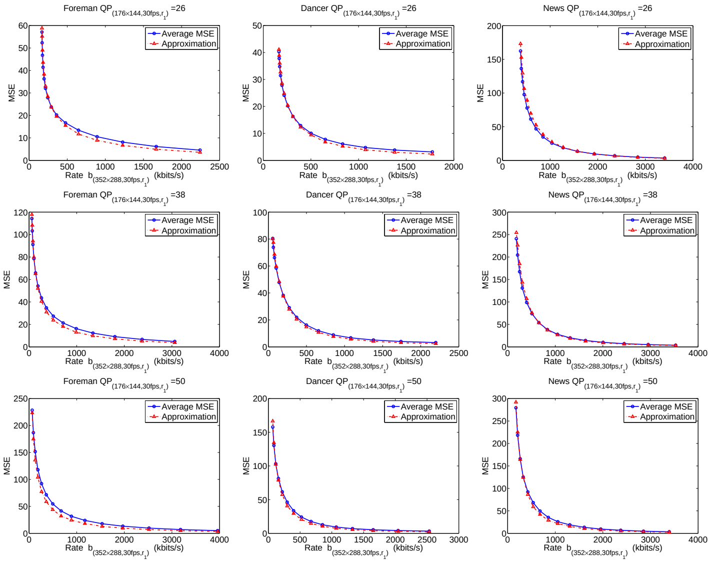

Fig. 4. Approximation of the distortion of the predicted layer due to spatial prediction and temporal prediction based on Equation (40). The prediction accuracy depends on the QP values of the predicting layer.

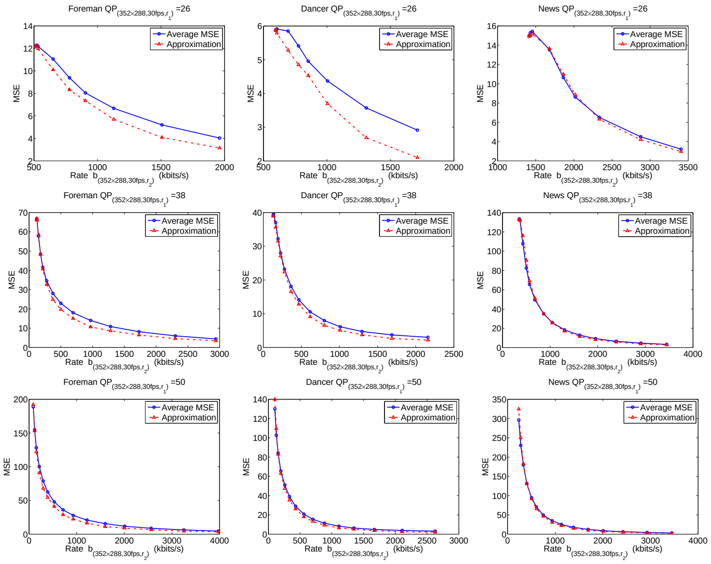

Fig. 5. Approximation of the distortion of the predicted layer due to quality prediction and temporal prediction based on Equation (41). The prediction accuracy depends on the QP values of the predicting layer.

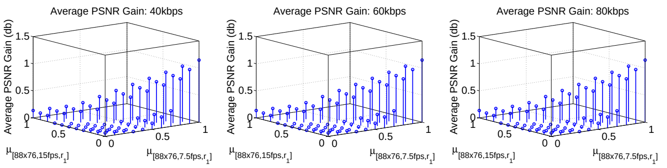

Fig. 6. The *PSNR* gain of *Proposed* over *Lagrangian* in terms of temporal scalability. The preferences $\mu_{[88\times72,7.5fps,r_1]}$ and $\mu_{[88\times72,15fps,r_1]}$ are shown; and, the preference $\mu_{[88\times72,30fps,r_1]}$ can be obtained by $(1-\mu_{[88\times72,7.5fps,r_1]}-\mu_{[88\times72,15fps,r_1]})$.

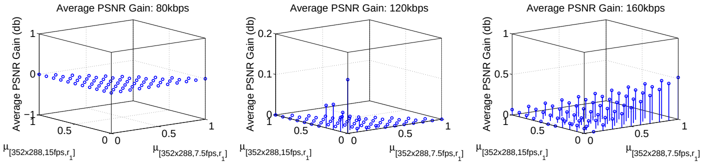

Fig. 7. The *PSNR* gain of *Proposed* over *Lagrangian* in terms of temporal scalability. The preferences $\mu_{[352\times288,7.5fps,r_1]}$ and $\mu_{[352\times288,15fps,r_1]}$ are shown; and, the preference $\mu_{[352\times288,30fps,r_1]}$ can be obtained by $(1-\mu_{[352\times288,7.5fps,r_1]}-\mu_{[352\times288,15fps,r_1]})$.

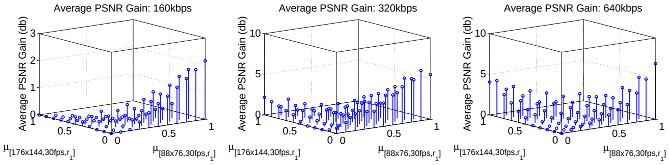

Fig. 8. The *PSNR* gain of *Proposed* over *Lagrangian* in terms of spatial scalability. The preferences $\mu_{[88\times72,30fps,r_1]}$ and $\mu_{[176\times144,30fps,r_1]}$ are shown; and, the preference $\mu_{[352\times288,30fps,r_1]}$ can be computed by $(1-\mu_{[88\times72,30fps,r_1]}-\mu_{[176\times144,30fps,r_1]})$.

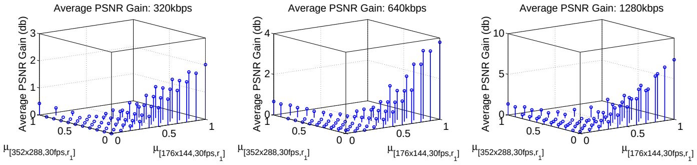

Fig. 9. The *PSNR* gain of *Proposed* over *Lagrangian* in terms of spatial scalability. The preferences $\mu_{[176\times144,30fps,r_1]}$ and $\mu_{[352\times288,30fps,r_1]}$ are shown; and, the preference $\mu_{[704\times576,30fps,r_1]}$ can be computed by $(1-\mu_{[176\times144,30fps,r_1]}-\mu_{[352\times288,30fps,r_1]})$.

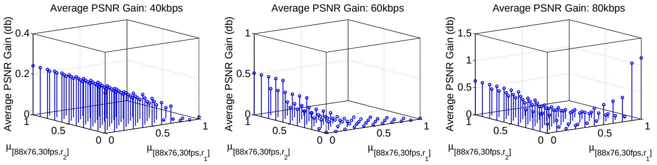

Fig. 10. The *PSNR* gain of *Proposed* over *Lagrangian* in terms of quality scalability. The preferences $\mu_{[88\times72,30fps,r_1]}$ and $\mu_{[88\times72,30fps,r_2]}$ are shown; and, the preference $\mu_{[88\times72,30fps,r_3]}$ can be computed by $(1-\mu_{[88\times72,30fps,r_2]}-\mu_{[88\times72,30fps,r_3]})$.

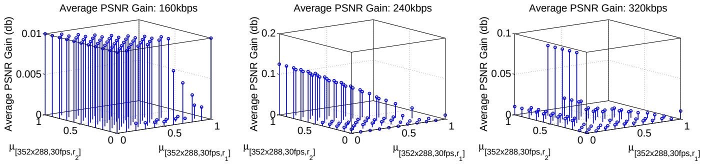

Fig. 11. The *PSNR* gain of *Proposed* over *Lagrangian* in terms of quality scalability. The preferences $\mu_{[352\times288,30fps,r_1]}$ and $\mu_{[352\times288,30fps,r_2]}$ are shown; and, the preference $\mu_{[352\times288,30fps,r_3]}$ can be computed by $(1-\mu_{[352\times288,30fps,r_2]}-\mu_{[352\times288,30fps,r_3]})$.

Guan-Ju Peng was born in Taiwan in 1982. He received a B.S. degree in Computer Science and Information Engineering, and an M.S. degree in Electrical Engineer from National Taiwan University in 2004 and 2006 respectively. Since 2007, he has been a research assistant in Dr. Wen-Liang Hwangs laboratory at the Institute of Information Science, Academia Sinica. Currently, he is studying for his Ph.D under the direction of Prof. Sao-Jie Chen at the Graduate Institute of Electronic Engineering, National Taiwan University. His research interests include video coding, video transmission, and wavelets.

Wen-Liang Hwang received his B.S. Degree in Nuclear Engineering from National Tsing Hua University, Hsinchu, Taiwan; his M.S. Degree in Electrical Engineering from the Polytechnic Institute of New York, New York; and, in 1993, his Ph.D. in Computer Science from New York University, New York. He was a postdoctoral researcher with the Department of Mathematics, University of California, Irvine in 1994. In January 1995, he became a member of the Institute of Information Science, Academia Sinica, Taipei, Taiwan, where he is currently a Research Fellow. He is co-author of the book: ”Practical Time-Frequency Analysis,” Academic Press, 1998. Currently, Dr. Hwang is an associate editor of the Journal of Wavelet Theory and Applications and International Journal of Wavelets, Multiresolution and Information Processing. His research interests include wavelet analysis, signal and image processing, and multimedia compression and transmission. In 2001, he was awarded the Academia Sinica Research Award for Junior Researchers.

Sao-Jie Chen received the B.S. and M.S. degrees in electrical engineering from the National Taiwan University, Taipei, Taiwan, ROC, in 1977 and 1982 respectively, and the Ph.D. degree in electrical engineering from the Southern Methodist University, Dallas, USA, in 1988. Since 1982, he has been a member of the faculty in the Department of Electrical Engineering, National Taiwan University, where he is currently a full professor. During the fall of 1999, he was a visiting professor in the Department of Computer Science and Engineering, University of California, San Diego, USA. During the fall of 2003, he held an academic visitor position in the Department of System Level Design, IBM Thomas J. Watson Research Center, Yorktown Heights, New York, USA. He obtained the ”Outstanding Electrical Engineering Professor Award” by the Chinese Institute of Electrical Engineering in December 2003 to recognize his excellent contributions to EE education. During the falls of 2004 to 2009, he was a visiting professor in the Department of Electrical and Computer Engineering, University of Wisconsin, Madison, USA. He was also an International Adjunct Professor in the Department of Electrical and Computer Engineering, University of Illinois, Urbana-Champaign, for the Spring Semester, 2010 and 2011. His current research interests include: VLSI physical design, SOC hardware/software co-design, Network-on-Chip, and Wireless LAN and Bluetooth IC design. Dr. Chen is a member of the Chinese Institute of Engineers, the Chinese Institute of Electrical Engineering, the Institute of Taiwanese IC Design, the Association for Computing Machinery, a senior member of the IEEE Circuits and Systems and the IEEE Computer Societies