Joint Optimization of Base and Enhancement Layers in Scalable Audio Coding

Emmanuel Ravelli, Vinay Melkote, *Member, IEEE*, Tejaswi Nanjundaswamy, *Student Member, IEEE*, and Kenneth Rose, *Fellow, IEEE*

***Abstract*—Current scalable audio coders typically optimize performance at a particular layer without regard to impact on other layers, and are thus unable to provide a performance trade-off between different layers. In the particular case of MPEG Scalable Advanced Audio Coding (S-AAC) and Scalable-to-Lossless (SLS) coding, the base-layer is optimized first followed by successive optimization of higher layers, which ensures optimality of the base-layer but results in a scalability penalty that progressively increases with the enhancement layer index. The ability to trade-off performance between different layers enables alignment to the real world requirement for audio quality commensurate with the bandwidth afforded by a user. This work provides the means to better control the performance tradeoffs, and the distribution of the scalability penalty, between the base and enhancement layers. Specifically, it proposes an efficient joint optimization algorithm that selects the encoding parameters for each layer while accounting for the rate-distortion costs in all layers. The efficacy of the technique is demonstrated in the two distinct settings of S-AAC, and SLS High Definition Advanced Audio Coding. Objective and subjective tests provide evidence for substantial gains, and represent a significant step toward bridging the gap with the non-scalable coder.**

***Index Terms*—Iterative optimization, joint optimization, rate-distortion optimization, scalable audio coding, scalable-to-lossless.**

I. Introduction

In scalable audio coding, the signal is encoded in a multi-layered or embedded bitstream. The base layer provides a coarse reconstruction, that can be successively refined on receipt of additional enhancement layers. Such an embedded bit-stream is particularly useful for client-server applications in heterogeneous networks, where clients could have diverse bit-rate or quality constraints. In such a scenario, the server needs to store only the scalable bit-stream for each audio sample, in lieu of multiple versions encoded at different bit-rates or quality levels. Another benefit of an embedded bitstream is that it provides network nodes the ability to simply drop packets corresponding to higher layers to satisfy link capacity constraints. Scalable audio coding has been incorporated into the MPEG-4 General Audio [1] coding standard in various forms such as scalable advanced audio coding (AAC) [2], bit-sliced arithmetic coding [3], and scalable-to-lossless (SLS) audio coding [4].

The selection of parameters in the scalable encoding process is critical to the rate-distortion (RD) performance at each layer. The prevalent approach of several scalable encoders is to optimize each layer successively and regardless of impact on higher layers: the residual signal after each stage or layer is encoded using the same optimization techniques employed in single layer or non-scalable coding. For instance, in two-layered scalable AAC (illustrated in the left part of Fig. 1), the audio signal is divided into frames similar to non-scalable AAC. Each frame is transformed to the frequency domain, and the transform coefficients are perceptually quantized and coded following the same process as in the non-scalable AAC encoder. A psychoacoustic model generates the per-frequency band masking thresholds required in the quantization module. These quantized coefficients now form the coarse base layer. The residue or quantization error spectrum from the base layer is then encoded with finer resolution into the enhancement layer, via the same non-scalable AAC encoding process. Note that this scheme can be easily extended to include more enhancement layers. Formal evaluation of such a scalable AAC codec [5] has shown that although the enhancement layers improve the perceptual quality beyond that of the base layer, there exists a significant performance gap compared to a non-scalable AAC encoder operating at the same cumulative bit-rate. In [6] and [7] it was demonstrated that despite employing an RD optimal AAC coder at every layer, the scalable AAC fails to match the performance of a non-scalable codec. This limitation critically impedes the practical deployment of such systems. For instance, in a client-server application that utilizes such a scalable codec, clients subscribing to a higher bit-rate version are impacted with a more severe "scalability penalty", and are in fact experiencing signal quality inferior to the bit rate expended. This led providers to revert to the wasteful practice of storing multiple copies of the audio file encoded at different quality levels/bit-rates.

Motivated by these observations we propose a method that enables performance tradeoffs between different layers of the scalable coding scheme. Rather than minimize separate RD cost functions for each layer successively and regardless of impact on higher layers, the proposed technique endeavors to minimize a single cost function that incorporates the relative importance of different layers via a weighted combination of their individual RD costs. Since the exact joint optimization of this amal-

Manuscript received January 23, 2012; revised June 30, 2012 and October 05, 2012; accepted October 15, 2012. Date of publication November 30, 2012; date of current version January 11, 2013. This work was supported in part by the National Science Foundation (NSF) under Grant CCF-0917230, and by Qualcomm, Inc.. The research was performed while E. Ravelli and V. Melkote were at The University of California, Santa Barbara. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Laurent Daudet.

E. Ravelli is with Fraunhofer IIS, 91058 Erlangen, Germany (e-mail: ravelli@ece.ucsb.edu).

V. Melkote is with Dolby Laboratories, Inc., San Francisco, CA 94103-4813 USA (e-mail: melkote@ece.ucsb.edu).

T. Nanjundaswamy and K. Rose are with the Department of Electrical and Computer Engineering, University of California, Santa Barbara (UCSB), CA 93106-9560 USA (e-mail: rose@ece.ucsb.edu).

Digital Object Identifier 10.1109/TASL.2012.2231071

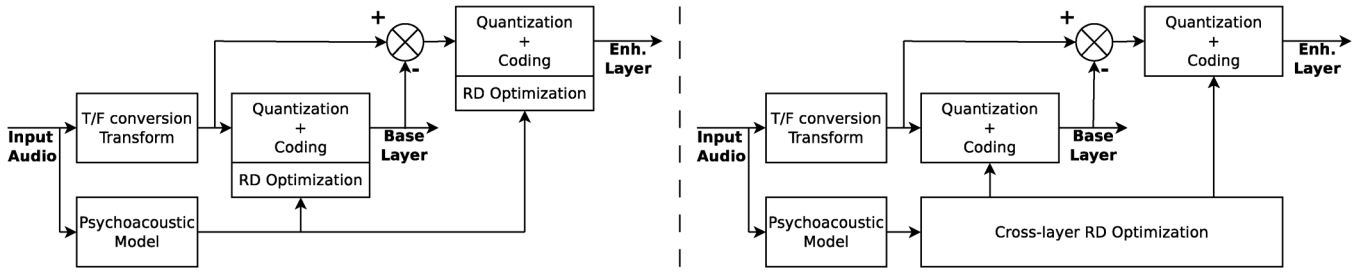


Fig. 1. Existing scalable coder with successive optimization of layers (left) versus proposed join optimization of the layers (right).

gamated cost function over encoding parameters of all layers incurs complexity exponential in the number of layers, an alternative iterative optimization technique is proposed. Similar to existing scalable coding techniques, the proposed iterative optimization works on one layer at a time. However, in a significant departure from current techniques, the cost function minimized at each layer accounts for the impact of encoding decisions made in that layer on the rate and distortion in *all* layers. Each iteration refines the parameter choices, successively for each layer, until convergence.

The proposed approach is first demonstrated in the setting of scalable AAC (illustrated in the right part of Fig. 1). The optimal AAC encoding parameters in each layer (scalefactors and Huffman code books) are chosen such that a single cost function (a weighted combination of costs in different layers) is minimized under multiple constraints (one per layer, either on the bit-rate or the distortion). Trellis-based techniques for optimal non-scalable AAC parameter selection, developed previously by our research group in [6]–[8] are leveraged and incorporated as building blocks for the proposed algorithm. Objective and subjective evaluations demonstrate the effectiveness of the proposed approach. The results also demonstrate the technique's capability to achieve improved performance at higher layers, at a cost to the lower layers: a characteristic clearly appealing to the server-client application mentioned previously. Note that the proposed technique is an encoder only modification and retains compatibility with the standard bitstream syntax.

Subsequently, we extend the proposed approach to encompass High Definition AAC (HD-AAC), a scalable-to-lossless (SLS) encoding standard. In HD-AAC, audio is encoded into a base AAC layer that ensures state-of-the-art perceptual quality and backward compatibility with legacy decoders, while the residue from this base layer is encoded in fine-grained SLS enhancements that enable bit-exact reconstruction. HD-AAC without the perceptually coded base layer forms the SLS "non-core" (NC) codec, which reportedly achieves better lossless compression than HD-AAC [9], albeit with a poor perceptual quality at intermediate (lossy) bit-rates. The parameters of HD-AAC are currently selected in a manner similar to other scalable coders, i.e., the parameters of the base layer and the SLS enhancements are selected successively and regardless of impact on higher layers. This clearly ignores the fact that the residual signal after base-layer coding, and in turn its lossless compression by SLS, is largely determined by the choice of AAC encoding parameters. Thus we propose selecting AAC encoder parameters, while explicitly accounting for its effects on subsequent SLS encoding via a single combined cost function, in line with our approach for scalable AAC. This algorithm clearly enables a trade-off between the perceptual (and lossy) coding performance at the AAC core, and the overall lossless compression. Simulations provide evidence that careful optimization achieves "the best of both worlds", namely, an AAC core of very good perceptual quality, and lossless coding performance comparable to (and in some cases surpassing) that of the SLS NC codec.

Preliminary results for disparate components of the proposed approach have appeared in [10] (scalable AAC) and [11] (scalable-to-lossless). This paper subsumes the earlier publications within a unifying Lagrangian framework, including complete and detailed optimization techniques, extensive subjective listening test results, and explicit treatment of convergence and complexity of the proposed approach.

The related literature includes prior work on RD optimized audio coding and scalable audio coding. In [6] and [7] a trellis-based approach was proposed for optimal encoding parameter selection for each frame in single layer audio coding. A low-complexity sub-optimal alternative was proposed in [12]. A mixed integer linear programming-based solution to the same problem was proposed by Bauer and Vinton in [13]. The approach in [7] was extended in [8] to incorporate delayed decisions, i.e., optimizing decisions over multiple frames. Notable prior work on scalable audio coding include [14] on optimal scalable quantization for the weighted mean square error criterion, [15] where embedded zerotree algorithms are employed, [16] which compares different fine-grained scalable audio coding schemes, and [17] where a perceptually scalable TWIN-VQ based audio coding algorithm is proposed. While an extensive search of the existing literature yielded no prior work that considered joint optimization of multiple layers in scalable audio coding, it is noteworthy that this research direction has been pursued in scalable video coding [18], [19].

The paper is structured as follows. Background on scalable AAC and HD-AAC is provided in Section II. The RD problem is formulated in Section III. The current approach of successive optimization of each layer regardless of impact on higher layers is described in Section IV. The proposed iterative joint optimization technique is detailed in Section V. Results are presented in Section VI, and the paper concludes in Section VII.

II. Background

This section provides background information on scalable audio coding schemes most relevant to this paper: i) scalable AAC [1], which is a large-step scalable coding scheme that consists of several AAC layers; and ii) SLS audio coding [20],

which provides fine-grained enhancements over a core perceptual codec, typically AAC, and which can ensure lossless reconstruction. We first describe the AAC base layer codec employed in either case.

A. Base Layer Codec: AAC

MPEG AAC is a highly flexible codec which can be operated according to several profiles and with many optional tools. For simplicity, but without loss of generality, this work employs the AAC Low Complexity (AAC LC) profile and excludes optional tools such as block switching, temporal noise shaping, etc. The description of the reference base-layer codec is as follows.

1) Base-Layer Encoder: The AAC encoder first segments the original audio signal into 50% overlapped frames of $2K$ samples each ($K = 1024$). Let an audio file consist of N such frames. A modified discrete cosine transform (MDCT) is employed to produce K transform coefficients per frame. We denote by $c_n[k]$, $0 \leq n < N$, $0 \leq k < K$ the k^{th} transform coefficient of frame n. These transform coefficients are grouped into L frequency bands, referred to as scalefactor bands (SFBs). All the coefficients in an SFB of a frame are now quantized using the same quantizer, which is a generic AAC quantizer scaled by a parameter called the scalefactor (SF), to produce the quantized coefficient indices $i_n^{(b)}[k]$. The quantization indices in a frame n, $\forall k \in$ SFB l are given as:

$$i_n^{(b)}[k] = \operatorname{sgn}\left(c_n[k]\right)\left\lfloor \left|2^{-s_n^{(b)}[l]/4} c_n[k]\right|^{\frac{3}{4}} + 0.4054 \right\rfloor \qquad (1)$$

where $\lfloor \cdot \rfloor$ is the floor function and $s_n^{(b)}[l]$ is the SF index (the superscript (b) specifies that the variable under consideration belongs to the base layer). The purpose of these scalefactors is to shape the quantization noise differently in each frequency band. This noise shaping is generally performed with the help of a psychoacoustic model, and aims to minimize the perceived distortion, typically calculated as the noise-to-mask ratio. The quantization indices within an SFB are encoded with the same Huffman codebook (HCB), denoted as $h_n^{(b)}[l]$. The AAC bitstream consists of the entropy-coded quantized coefficients and, as side information, one SF and one HCB index per SFB. The SF values are differentially encoded, and the HCB indices are run-length encoded across SFBs. Note that when all coefficients of SFB l are quantized to zero, then $h_n^{(b)}[l]$ is set to 0, and no additional information is sent for this SFB. It is important to note that the standard only dictates the decoder part of the codec. The encoder is thus not standardized and may be implemented in various ways, for instance, a different quantization rule could possibly be used [21]. We use in this paper the public-domain encoder described in the informative part of the MPEG standard.

2) Base-Layer Decoder: The AAC decoder unpacks the entropy coded quantization indices, and inverse quantizes them using the decoded SFs, resulting in the reconstructed coefficients $\hat{c}_n^{(b)}[k]$ given by

$$\hat{c}_n^{(b)}[k] = \operatorname{sgn}\left(i_n^{(b)}[k]\right)\left(2^{s_n^{(b)}[l]/4}\left|i_n^{(b)}[k]\right|^{\frac{4}{3}}\right) \qquad (2)$$

An inverse MDCT transforms the quantized coefficients back to the time domain.

B. Scalable AAC

In the scalable AAC codec the base layer reconstruction error is successively refined by the enhancement layers, each time employing the same quantization and coding process as in the base layer. For simplicity of exposition we focus in this paper on a two-layered scalable AAC codec. The base layer is encoded as previously described. The enhancement AAC layer is encoded/decoded as follows.

1) Enhancement-Layer Encoder: A base-layer AAC decoder within the scalable AAC encoder reconstructs the coefficients via (2). The reconstruction error in the base layer is

$$e_n[k] = c_n[k] - \hat{c}_n^{(b)}[k]. \qquad (3)$$

This reconstruction error is requantized using the same generic AAC quantizer in each SFB, but with possibly different SFs $s_n^{(e)}[l]$, thus producing the enhancement layer indices

$$i_n^{(e)}[k] = \operatorname{sgn}\left(e_n[k]\right)\left\lfloor \left|2^{-s_n^{(e)}[l]/4} e_n[k]\right|^{\frac{3}{4}} + 0.4054 \right\rfloor. \qquad (4)$$

The enhancement layer quantization indices are encoded with HCBs $h_n^{(e)}[l]$. The enhancement bitstream is similar to the base-layer AAC bitstream, and consists of entropy-coded quantization indices, as well as enhancement layer SF and HCB indices as side information.

2) Enhancement-Layer Decoder: The scalable AAC decoder can either decode only the base-layer to produce the coefficients $\hat{c}_n^{(b)}[k]$, or decode both layers to obtain refined coefficients $\hat{c}_n^{(e)}[k] = \hat{c}_n^{(b)}[k] + \hat{e}_n[k]$, where the enhancement correction $\hat{e}_n[k]$ is the requantized base layer reconstruction error, given by

$$\hat{e}_n[k] = \operatorname{sgn}\left(i_n^{(e)}[k]\right)\left(2^{s_n^{(e)}[l]/4}\left|i_n^{(e)}[k]\right|^{\frac{4}{3}}\right). \qquad (5)$$

An inverse MDCT is finally employed to obtain the decoded time domain signal.

3) Notation for Scalable AAC Parameters: We denote the base layer encoding parameters by $\mathbf{p}_n^{(b)} = (\mathbf{s}_n^{(b)}, \mathbf{h}_n^{(b)})$ and the enhancement layer parameters by $\mathbf{p}_n^{(e)} = (\mathbf{s}_n^{(e)}, \mathbf{h}_n^{(e)})$, for every frame n, with $\mathbf{s}_n^{(x)} = \{s_n^{(x)}[0], \ldots, s_n^{(x)}[L-1]\}$, and $\mathbf{h}_n^{(x)} = \{h_n^{(x)}[0], \ldots, h_n^{(x)}[L-1]\}$, $x \in \{b, e\}$. The complete set of base and enhancement layer encoding parameters is summarized in $\mathcal{P}^{(b)} = \{\mathbf{p}_0^{(b)}, \ldots, \mathbf{p}_{N-1}^{(b)}\}$, and $\mathcal{P}^{(e)} = \{\mathbf{p}_0^{(e)}, \ldots, \mathbf{p}_{N-1}^{(e)}\}$, respectively, with the overall set $\mathcal{P} = \{\mathcal{P}^{(b)}, \mathcal{P}^{(e)}\}$.

C. MPEG SLS Coding

In MPEG SLS, the residue from the base layer (or core) codec is encoded in fine-grained enhancements that enable an ultimate lossless reconstruction, when needed. Although the MPEG standard [20] does not mandate a fixed choice of the core codec, we describe our approach in the setting of the popular HD-AAC profile [22] that employs an AAC LC coder in the base layer. Since our primary focus in the SLS case is on an approach that

trades-off the *perceptual quality of the base layer* with the *lossless compression performance* of the SLS encoder, we henceforth employ the term 'SLS layer' or 'enhancement layer' in conjunction with HD-AAC loosely, to refer to the *entirety* of fine-grained enhancements. Note, however, that the SLS standard does allow partial decoding of fine-grained enhancements, corresponding to a lossy reconstruction. For simplicity, we do not consider performance at these fine-grained enhancements in our optimization problem.

1) SLS Encoder: The lossless compression feature necessitates the use of a reversible integer transform known as IntMDCT in the base layer encoder, that closely approximates standard MDCT. The integer coefficients are then coded by the AAC quantization and coding module to produce a base layer bitstream compatible with the basic AAC standard. Subsequently, an error mapping process calculates the residual coefficients that are coded into the SLS enhancement layer. If all the base layer quantized coefficients in an SFB are zeros (i.e., $h_n^{(b)}[l] = 0$), the band is referred to as an explicit band and the mapped error $e_n[k]$ is simply equal to the original coefficients, i.e., $e_n[k] = c_n[k]$. If not, the band is said to be an implicit band, and the mapped error is

$$e_n[k] = c_n[k] - \left\lfloor \mathrm{thr}\left(i_n^{(b)}[k]\right) \right\rfloor \tag{6}$$

where $i_n^{(b)}[k]$ denotes the base layer quantized coefficient index and $\mathrm{thr}(i_n^{(b)}[k])$ is the boundary closer to zero of the AAC quantizer cell with index $i_n^{(b)}[k]$, and is given $\forall k \in$ SFB l, as

$$\mathrm{thr}\left(i_n^{(b)}[k]\right) = \mathrm{sgn}\left(i_n^{(b)}[k]\right)\left(2^{s_n^{(b)}[l]/4}\left|i_n^{(b)}[k] - 0.4054\right|^{\frac{4}{3}}\right) \tag{7}$$

for $i_n^{(b)}[k] \neq 0$, and $\mathrm{thr}(0) = 0$. Finally, the mapped error is encoded using either bit-plane Golomb codes (BPGC) or low energy mode codes (LEMC) [20], both of which are bit-plane arithmetic codes (note that another option of context-based arithmetic coding, standardized as part of [20], is not permitted in HD-AAC [22]). In bit-plane coding, the magnitude of error coefficients in SFB l is expressed in $\mathcal{M}_n[l]$-bit binary representation as

$$|e_n[k]| = \sum_{j=0}^{\mathcal{M}_n[l]} b[k,j]2^j, \tag{8}$$

where $\mathcal{M}_n[l]$ is the most significant bit (MSB)-plane in SFB l. The parameter $\mathcal{M}_n[l]$ is deduced from the AAC quantizer cell widths if SFB l is an implicit band, else it is differentially encoded and sent to the decoder as side information. Note that the mapped error $e_n[k]$ and the MSB-planes $\mathcal{M}_n[l]$ are determined by the AAC core. Thus the only free encoding parameters in the SLS enhancement layer are the so called 'lazy bit-plane' parameters $\mathcal{L}_n[l]$ [4], that characterize the binary distribution of bits in each SFB, and control the arithmetic coding operation. The bits $b[k,j]$ in each SFB are encoded using BPGC if $\mathcal{M}_n[l] - \mathcal{L}_n[l] > 0$, else LEMC is used. The parameter $\mathcal{L}_n[l]$ for each SFB is Huffman coded and sent as side information. Thus the SLS enhancement layer bitstream consists of the arithmetic-coded mapped error and the side information of the MSB-planes and the lazy bit-plane parameters.

2) SLS Decoder: The SLS decoder converts the arithmetic coded bits in the enhancement layer to a bit-exact reconstruction of the mapped error which, along with the AAC base layer information, provides an exact reproduction of the IntMDCT coefficients. Finally, an inverse IntMDCT results in a lossless time-domain reconstruction of the original audio signal.

3) Notation for SLS Parameters: The same notation as in the scalable AAC case applies at the base layer. The encoding parameters in the SLS enhancement layer are given by $\mathbf{p}_n^{(e)} = \boldsymbol{\mathcal{L}}_n = \{\mathcal{L}_n[0], \ldots, \mathcal{L}_n[L-1]\}$ for every frame n.

III. The Scalable Encoder Optimization Problem

A prerequisite for specification of a rate-distortion optimization problem is the definition of the distortion metric. We define the overall distortion of an audio file in terms of the widely employed noise-to-mask ratio (NMR), which is calculated for each SFB as the ratio of the quantization noise energy to a noise masking threshold provided by a psychoacoustic model [23], [24]. The per-SFB NMR, for the base layer, depends on the SF value and is defined as

$$d_{n,l}^{(b)}\left(s_n^{(b)}[l]\right) = \frac{\sum_{k\in\mathrm{SFB}\ l}\left(c_n[k] - \hat{c}_n^{(b)}[k]\right)^2}{\mu_n[l]} \tag{9}$$

where $\mu_n[l]$ is the masking threshold in SFB l of frame n. While NMR is commonly accepted as a simple and effective estimate of the perceptual distortion in a given SFB of a given frame, there are various ways to combine such NMRs, computed for each SFB in each frame, into a single overall distortion for the entire audio signal. Recent work (e.g. [8]) proposed several simple distortion metrics obtained by averaging or maximizing NMR over SFBs and/or frames. We use in this paper the max-max NMR (MMNMR), defined as:

$$\mathcal{D}^{(b)}\left(\mathcal{P}^{(b)}\right) = \max_{0\le n<N}\max_{0\le l<L} d_{n,l}^{(b)}\left(s_n^{(b)}[l]\right). \tag{10}$$

It can be argued that minimizing the maximum NMR across SFBs and frames ensures that the NMR is approximately the same in all time-frequency tiles, which has a perceptually pleasing effect as confirmed via informal listening tests, and hence the choice of MMNMR as the distortion metric. Moreover, this metric simplified the derivation of optimization techniques proposed in this paper.

Similarly, the per-SFB NMR in the enhancement layer is given by

$$d_{n,l}^{(e)}\left(s_n^{(b)}[l], s_n^{(e)}[l]\right) = \frac{\sum_{k\in\mathrm{SFB}\ l}\left(c_n[k] - \hat{c}_n^{(e)}[k]\right)^2}{\mu_n[l]}. \tag{11}$$

It is important to note that the enhancement layer distortion $d_{n,l}^{(e)}$ is obtained after decoding both the base-layer and the enhancement-layer. This distortion thus depends on the choice of the base layer SF, $s_n^{(b)}[l]$, as well. The enhancement layer MMNMR is then,

$$\mathcal{D}^{(e)}\left(\mathcal{P}^{(b)}, \mathcal{P}^{(e)}\right) = \max_{0\le n<N}\max_{0\le l<L} d_{n,l}^{(e)}\left(s_n^{(b)}[l], s_n^{(e)}[l]\right). \tag{12}$$

Recall that we consider the entirety of the fine grained enhancements in SLS coding as a single layer that provides lossless reconstruction (and thus zero distortion). Therefore the above definition of enhancement layer distortion is relevant only to the case of the scalable AAC codec.

We denote by $\mathcal{R}_n^{(b)}(\mathbf{p}_n^{(b)})$ the bit-rate consumed in the base layer of frame n and by $\mathcal{R}_n^{(e)}(\mathbf{p}_n^{(b)}, \mathbf{p}_n^{(e)})$ the cumulative bit-rate consumed by the base and enhancement layers of frame n. Note that the cumulative bit-rate is influenced by base layer parameters in part because the rate contribution from the enhancement layer depends on the base layer reconstruction error it codes, which in turn is determined by $\mathbf{p}_n^{(b)}$. The overall base layer and cumulative bit-rates are denoted by $\mathcal{R}^{(b)}(\mathcal{P}^{(b)})$ and $\mathcal{R}^{(e)}(\mathcal{P}^{(b)}, \mathcal{P}^{(e)})$, respectively, which are, naturally, averaged over the frames.

$$\mathcal{R}^{(x)}(\cdot) = \frac{1}{N}\sum_{n=0}^{N-1}\mathcal{R}_n^{x}(\cdot), \quad x \in \{b, e\}. \tag{13}$$

A. The Rate-Constrained Problem

In most applications, the bit-rate at different layers is constrained and a weighted sum of distortion in both layers needs to be minimized. In the case of scalable AAC this rate-constrained optimization is specified as follows:

$$\mathcal{P}^* = \arg\min_{\mathcal{P}} \left\{(1-\beta)\mathcal{D}^{(b)}\left(\mathcal{P}^{(b)}\right) + \beta\mathcal{D}^{(e)}(\mathcal{P})\right\} \quad \text{s.t. } \mathcal{R}^{(b)}\left(\mathcal{P}^{(b)}\right) \le \mathcal{R}_t^{(b)} \text{ and } \mathcal{R}^{(e)}(\mathcal{P}) \le \mathcal{R}_t^{(e)} \tag{14}$$

where $\mathcal{R}_t^{(b)}, \mathcal{R}_t^{(e)}$ are the target base layer and cumulative bit-rates. The weight β enables a performance trade-off between the two layers.

In the MPEG SLS case, a bit-rate constraint is specified only for the base-layer. The distortion in the enhancement layer is zero, and the aim is to achieve a trade off between the lossless compression bit-rate (the cumulative bit-rate of the AAC base layer and SLS enhancement) and the base-layer distortion. Thus, the optimization problem is now specified as:

$$\mathcal{P}^* = \arg\min_{\mathcal{P}} \left\{(1-\beta)\mathcal{D}^{(b)}\left(\mathcal{P}^{(b)}\right) + \beta\mathcal{R}^{(e)}(\mathcal{P})\right\} \quad \text{s.t. } \mathcal{R}^{(b)}\left(\mathcal{P}^{(b)}\right) \le \mathcal{R}_t^{(b)} \tag{15}$$

These optimization problems can be extended in a straightforward manner to include bit-rates and distortions of multiple layers, e.g., more than two layers in the scalable AAC. However, we restrict this paper to two layers for notational and presentation simplicity.

B. The Distortion-Constrained Problem

A dual specification of the rate-distortion optimization problem in the scalable AAC case is in terms of distortion constraints:

$$\mathcal{P}^* = \arg\min_{\mathcal{P}} \left\{(1-\alpha)\mathcal{R}^{(b)}\left(\mathcal{P}^{(b)}\right) + \alpha\mathcal{R}^{(e)}(\mathcal{P})\right\} \quad \text{s.t. } \mathcal{D}^{(b)}\left(\mathcal{P}^{(b)}\right) \le \mathcal{D}_t^{(b)} \text{ and } \mathcal{D}^{(e)}(\mathcal{P}) \le \mathcal{D}_t^{(e)} \tag{16}$$

where $\mathcal{D}_t^{(b)}, \mathcal{D}_t^{(e)}$ are the target distortions in each layer. Note that the solution to the rate-constrained problem in (15) can be obtained by iteratively solving (17) under different distortion constraints and weights: if the solution to (17) under an initial set of constraints does not meet the given rate-constraints of (15), $\mathcal{D}_t^{(b)}$, $\mathcal{D}_t^{(e)}$, then α can be altered suitably and the optimization in (17) redone. The process is repeated until the required constraints are met.

The appeal of such an approach to solving a rate-constrained problem as a sequence of distortion-constrained problems lies in the following. By the definition of MMNMR, the overall distortion constraints in (17) are equivalent to the per-SFB constraints below:

$$d_{n,l}^{(b)}\left(s_n^{(b)}[l]\right) \le \mathcal{D}_t^{(b)} \text{ and } d_{n,l}^{(e)}\left(s_n^{(b)}[l], s_n^{(e)}[l]\right) \le \mathcal{D}_t^{(e)} \tag{17}$$

for all $0 \le n < N$ and $0 \le l < L$. This observation, along with (13), significantly simplifies the overall optimization problem in (17), by decomposing it into N per-frame minimizations of the form

$$\mathbf{p}_n^* = \arg\min_{\mathbf{p}_n}(1-\alpha)\mathcal{R}_n^{(b)}\left(\mathbf{p}_n^{(b)}\right) + \alpha\mathcal{R}_n^{(e)}(\mathbf{p}_n) \quad \text{s.t. } d_{n,l}^{(b)}\left(s_n^{(b)}[l]\right) \le \mathcal{D}_t^{(b)} \text{ and } d_{n,l}^{(e)}\left(s_n^{(b)}[l], s_n^{(e)}[l]\right) \le \mathcal{D}_t^{(e)}. \tag{18}$$

In contrast, a direct solution of (15) would entail considerable complexity as the rate constraint is for the entire duration of the signal.

In the MPEG SLS case, as the enhancement layer distortion is exactly zero, there is only one condition for the distortion-constrained problem, i.e.,

$$\mathcal{P}^* = \arg\min_{\mathcal{P}} \left\{(1-\alpha)\mathcal{R}^{(b)}\left(\mathcal{P}^{(b)}\right) + \alpha\mathcal{R}^{(e)}(\mathcal{P})\right\} \quad \text{s.t. } \mathcal{D}^{(b)}\left(\mathcal{P}^{(b)}\right) \le \mathcal{D}_t^{(b)} \tag{19}$$

Similar to the scalable AAC case, the overall optimization problem in (19) is decomposed into N per-frame minimizations of the form

$$\mathbf{p}_n^* = \arg\min_{\mathbf{p}_n} \left\{(1-\alpha)\mathcal{R}_n^{(b)}\left(\mathbf{p}_n^{(b)}\right) + \alpha\mathcal{R}_n^{(e)}(\mathbf{p}_n)\right\} \quad \text{s.t. } d_{n,l}^{(b)}\left(s_n^{(b)}[l]\right) \le \mathcal{D}_t^{(b)} \tag{20}$$

IV. Current Approach: Separate Layer Optimization

The prevalent method for selecting encoding parameters in a two-layer scalable audio codec is to optimize each layer independently and successively. This section briefly describes the application of this technique in the frameworks of scalable AAC and SLS audio coding.

A. Independent Optimization of Layers in Scalable AAC

The AAC coding parameters for the base-layer are chosen first, such that a criterion measuring the base-layer distortion

is minimized and the base-layer rate constraint, $\mathcal{R}_t^{(b)}$, is met. With the base-layer coding parameters now fixed, the residue to be coded at the enhancement layer is known. The enhancement layer parameters are now chosen such that the distortion at that layer is minimized, while meeting the constraint $\mathcal{R}_t^{(e)}$ on the cumulative bit-rate. Since the layers are optimized successively and regardless of impact on higher layers, the same optimization tool as in the non-scalable AAC encoder can be employed at each layer. Typically, existing encoders analyze each frame of the audio file individually, and select the SFs and HCBs for each layer of the frame via sub-optimal low-complexity techniques (see e.g. [25], or [26]). These approaches provide a selection of encoding parameters that meets the rate-constraints, however it cannot ensure that the choice is optimal in terms of minimizing the distortion criterion.

In [6] and [7] a trellis search-based technique was proposed that is an RD optimal technique for parameter selection in a single AAC layer. This trellis approach analyzed the audio file one frame at a time, and ensured the optimal selection of parameters for individual frames. The intra-frame approach in [7] was further enhanced in [8] to jointly optimize encoding decisions for *all* the frames of the audio file. In other words, the latter work solved the base-layer optimization problem defined by (15) with $\beta = 0$. Note that the trellis-based single layer optimization tool of [8] can itself be employed successively at each layer in scalable AAC, with resulting improved performance over existing scalable encoders that employ sub-optimal techniques at each layer. Since the emphasis of this paper is on the benefits of joint optimization of all layers in scalable coding as opposed to the independent and successive optimization of each layer, we will consider as leading competitor or reference a scalable AAC encoder that employs the trellis method of [8] to successively optimize each layer. We emphasize again that this reference is in itself an improvement over existing scalable encoders that employ suboptimal techniques. As an aside, note that the mixed integer linear programming-based approach proposed in [13] can serve as an alternative to the intra-frame trellis of [7].

In [8] the rate-constrained optimization problem (15) with $\beta = 0$, is solved via the intermediate distortion-constrained problem (17) with $\alpha = 0$ (or equivalently the per-frame minimizations specified by (19)). Since $\alpha = 0$, the distortion cost in (19) is independent of the choice of enhancement layer parameters $\mathbf{p}_n^{(e)}$, and can thus be solved exactly with the intra-frame trellis in [7]. While we briefly describe this technique here, the interested reader is deferred to [7] or [8] for a detailed explanation.

For each audio frame, a trellis with L stages, each stage corresponding to one SFB, and nodes in each stage corresponding to all possible combinations of SF and HCB values, is constructed. The algorithm then consists of three steps:

1) The MDCT coefficients in each SFB are quantized and entropy coded with all allowed combinations of SF and HCB values. The corresponding distortions (per-band NMRs) and number of bits to encode the coefficients in the SFB are calculated.
2) The trellis nodes are populated with these rate and distortion values. Only the states that satisfy the distortion constraint in (19) are retained. Transitions between states are associated with bit costs required to differentially encode SFs and run-length encode HCB values.
3) Viterbi algorithm is then employed to find the path through this trellis that minimizes the cost function in (19). With $\alpha = 0$ this is simply the base-layer rate. The optimal path gives the optimal base layer SFs and HCBs for the frame that satisfy the distortion constraint in (19).

This trellis-based optimization is employed in each frame to solve (17). If the base-layer rate $\mathcal{R}^{(b)}(\mathcal{P}^{(b)})$, thus obtained, does not meet the rate constraint $\mathcal{R}_t^{(b)}$ in (15), the distortion constraint $\mathcal{D}_t^{(b)}$ is modified and the trellis-based optimization is redone. Details of the $\mathcal{D}_t^{(b)}$ updating mechanism is described below:

1) Solve (17) with distortion constraint $\mathcal{D}_1^{(b)} = \mathcal{D}_2^{(b)} = \mathcal{D}_{init}^{(b)}$.
2) If $\mathcal{R}^{(b)} > \mathcal{R}_t^{(b)}$, set $\mathcal{D}_2^{(b)} = \mathcal{D}_1^{(b)}$, assign $\mathcal{D}_1^{(b)} = \mathcal{D}_1^{(b)} K$, where K is a constant greater than 1, and redo the optimization with the updated distortion constraint $\mathcal{D}_1^{(b)}$.
3) If $\mathcal{R}^{(b)} < \mathcal{R}_t^{(b)}$, set $\mathcal{D}_1^{(b)} = \mathcal{D}_2^{(b)}$, assign $\mathcal{D}_2^{(b)} = \mathcal{D}_2^{(b)}/K$, and redo the optimization with the updated distortion constraint $\mathcal{D}_2^{(b)}$.
4) Repeat either 2) or 3) until $\mathcal{D}_1^{(b)}$ results in a rate less than the constraint and $\mathcal{D}_2^{(b)}$ results in a rate greater than the constraint.
5) Update $\mathcal{D}_t^{(b)} = \sqrt{\mathcal{D}_1^{(b)} \mathcal{D}_2^{(b)}}$ and redo optimization.
6) If $\mathcal{R}^{(b)} > \mathcal{R}_t^{(b)}$, then $\mathcal{D}_2^{(b)} = \mathcal{D}_t^{(b)}$, else $\mathcal{D}_1^{(b)} = \mathcal{D}_t^{(b)}$.
7) End algorithm if $\mathcal{R}^{(b)} < \mathcal{R}_t^{(b)}$ and $\mathcal{R}^{(b)} > \mathcal{R}_t^{(b)} - \delta$, else goto 5).

Once the base-layer target rate is achieved, the resulting residual signal is encoded at the enhancement layer by application of the same technique so that the bit-rate for that layer alone meets the constraint $\mathcal{R}_t^{(e)} - \mathcal{R}_t^{(b)}$, thus satisfying the cumulative rate constraint $\mathcal{R}_t^{(e)}$.

Clearly, this reference scalable AAC framework that successively optimizes the base and enhancement layers cannot ensure that a overall cost function, such as the weighted sum of per-layer distortions in (15) or the weighted sum of per-layer rates in (17), is minimized. Since the base layer optimization only considers the minimization of the base layer distortion $\mathcal{D}^{(b)}(\cdot)$ or rate $\mathcal{R}^{(b)}(\cdot)$, optimal coding performance at the enhancement layer is always limited by the choice of the base layer parameters $\mathcal{P}^{(b)}$. Consequently, the overall distortion with both layers $\mathcal{D}^{(e)}(\cdot)$ is significantly higher than the optimal performance of a single layer coder operating at bit-rate $\mathcal{R}_t^{(e)}$.

B. Independent Optimization of Layers in SLS

In the SLS case, a non-scalable AAC encoder modified to incorporate the IntMDCT is typically employed at the base-layer. Most existing implementations utilize sub-optimal low-complexity techniques for AAC parameter selection (see Section IV-A). The 'mapped error' from the base-layer (see

Section II-C-1) is then encoded by selecting the lazy bit-plane parameters $\mathcal{L}_n[l]$ via a suitable algorithm such as the heuristic approach suggested in [4]. Note again that the layers are successively optimized.

The reference SLS encoder we employ in this paper improves upon this design by utilizing the trellis-based approach described Section IV-A for optimal AAC parameter selection at the base-layer. Given this choice of base layer parameters, the lazy bit-plane parameter for each SFB (which can only take one of 3 standard prescribed values) is chosen by an exhaustive search over all possible values of $\mathcal{L}_n$ for the set that minimizes the enhancement layer bit-rate, in contrast to the heuristic approach in [4]. Note that the fine-grained enhancements are arithmetic coded, and hence an exact number of bits to encode the enhancement layer can only be obtained by actually coding the residual signal with a given choice of $\mathcal{L}_n$. In order to circumvent the complexity inherent in performing this coding process for all 3^L possible choices of $\mathcal{L}_n$, we approximate the bit-rate calculation thus: the rate for encoding bit $b[k,j]$ is estimated as the entropy $-\log_2[b[k,j]Q^x(j) + (1 - b[k,j])(1 - Q^x(j))]$, where $Q^x(j)$ is the probability assignment of the BPGC (or LEMC) coder for the j^{th} bit-plane with $x = \mathcal{L}_n[l]$. Note that with this approximation in place the SLS bits for individual SFBs can be calculated, and the best value of $\mathcal{L}_n[l]$ can be found independently for each SFB l, thus considerably reducing the exhaustive search complexity.

Obviously, the reference SLS framework provides optimal performance at the AAC base layer, i.e., it minimizes $\mathcal{D}^{(b)}(\cdot)$ under rate constraint $\mathcal{R}_t^{(b)}$. However, this optimization is myopic in that it fixes the parameters $\mathcal{P}^{(b)}$ regardless of the impact on subsequent lossless encoding of the residue. Thus, the overall lossless compression rate $\mathcal{R}^{(e)}(\cdot)$ may be considerably higher than in the absence of an AAC base-layer, i.e., if the entire signal was encoded by a 'non-core SLS'.

V. Proposed Approach: Joint Optimization of Layers

From the prior discussion it is obvious that the successive optimization of layers benefits the base-layer at the expense of performance at enhancement layers. In case of a streaming service for customers with disparate bandwidth limitations (or with subscription types of different quality levels), this approach favors the lowest rung customer who can afford only the base-layer. However, in order to maintain economic viability of such a service, a tradeoff needs to be balanced between the streaming quality provided to different customer types. This observation motivates the approach proposed herein, which jointly optimizes the choice of parameters for all layers by accounting for their effects in a single cost function.

The proposed method first considers the solution to the intermediate distortion-constrained problem (17), to eventually find a solution for the original rate-constrained problems (15) or (15). This section describes the proposed optimization technique, separately, for the scenarios of scalable AAC, and the MPEG SLS.

A. Joint Optimization of Layers in Scalable AAC

In Section IV-A we briefly described the trellis-based optimization technique to solve (17) when $\alpha = 0$. One can envision a direct extension for the case $\alpha > 0$, by simply increasing the number of states in each stage of the trellis to correspond to all combinations of both base and enhancement layer SFs and HCBs. The rate cost in each node would now be a weighted (by α) combination of base and enhancement layer bits for that SFB, and only such nodes are retained in the trellis where both base and enhancement layer distortion constraints (see (17)) are satisfied. The path through this extended trellis with minimum total cost corresponds to the *optimal* choice of base and enhancement layer parameters that solve (17). Typically an SF in any layer could be one of 60 different values, and there is a choice of 12 HCBs for each SFB. Therefore, when $\alpha = 0$ (i.e., only base layer optimization) a conservative estimate of the number of states in each stage of the trellis is $60 \times 12 = 720$. When $\alpha > 0$ both base and enhancement layer parameters are to be optimized, and the number of states increases to $(720)^2 = 518400$, and the number of transitions in the trellis increases to 518400^2. In other words, although the trellis search in [7] has complexity linear in the number of SFBs, its extension as described above to scalable AAC has complexity exponential in the number of layers.

In order to circumvent this undesirable complexity we propose an alternative simplified, albeit efficient, trellis-based iterative solution. Specifically, the parameters of the two layers are successively *and iteratively* optimized with a separate trellis for each layer. In any particular iteration, the base layer optimization involves minimization of the cost function in (17) over all $\mathcal{P}^{(b)}$, while assuming that $\mathcal{P}^{(e)}$ is unchanged from the previous iteration, in a trellis which now only requires to be populated with base layer parameters. Note that although only the base layer is optimized in this step, the approach is still significantly different from the existing method in that, *the weighted cost function ensures that the base layer parameters are selected with a conscious accounting of their effect on the enhancement layer coding*. Subsequently, $\mathcal{P}^{(b)}$ is fixed and the same weighted cost function is minimized over all choices of $\mathcal{P}^{(e)}$, in a trellis which now contains only enhancement layer parameters. This latter enhancement layer optimization step is still similar to the standard approach, as given $\mathcal{P}^{(b)}$ (and hence $\mathcal{R}^{(b)}$), the minimization of the cost function in (17) is simply equivalent to minimization of the enhancement layer bit-rate, $\mathcal{R}^{(e)}(\mathcal{P}) - \mathcal{R}^{(b)}$. Multiple iterations of these two steps of base and enhancement layer optimizations are performed until convergence. Since each step of the iteration only optimizes one layer the complexity of the trellis is not exponential in the number of layers anymore.

We note that, unlike the full extended trellis, the simplified iterative approach does not guarantee an optimal solution. However, on account of the weighted cost function it minimizes, and due to the multiple iterations of base and enhancement layer optimization, it still ensures a better RD performance than existing encoders. Obviously the first iteration of base layer optimization requires some initialization of the enhancement layer parameters $\mathcal{P}^{(e)}$. Since the performance of iterative optimization may be sensitive to initialization, we adopt an "informed"

initialization via the preliminary base layer optimization summarized below:

Algorithm I: Scalable AAC—Preliminary base layer optimization

1) The nodes in the base layer trellis for each frame are populated with base layer distortion and rate costs corresponding to different SF and HCB values. In every node of the trellis the quantization error/residual in MDCT coefficients is preserved. Only states that satisfy the base layer distortion constraint in (19) are retained.
2) For each of these retained nodes an enhancement layer SF and HCB value is found to encode the quantization residual, such that the enhancement layer distortion constraint is satisfied, and the number of bits required to entropy code the quantized residual in that node is minimized. Note that, at this point each state of the base layer trellis in a frame is associated with SF and HCB values for both base and enhancement layers.
3) Each state of the base layer trellis is now associated with a new cost, that is a weighted combination (via α) of the number of bits consumed in the two layers, to encode the quantized MDCT coefficients in the SFB.
4) Transitions between states are similarly associated with a weighted bit cost that accounts for the bits to differentially encode corresponding SFs, and run-length code HCB values, at both layers.
5) Viterbi algorithm is employed to find the path through the trellis in frame n that minimizes the cost function in (19), which provides a first approximation of $\mathbf{p}_n^{(b)}$. The procedure repeated in each frame yields a first choice of $\mathcal{P}^{(b)}$.

With this first choice of $\mathcal{P}^{(b)}$ now available, the iterative procedure for optimization can now be followed:

Algorithm II: Scalable AAC—Distortion-constrained optimization

1) Run Algorithm I to find an initial choice of $\mathcal{P}^{(b)}$.
2) Repeat the following steps until an exit condition is satisfied:
   a) Optimize the enhancement layer, i.e., find $\mathcal{P}^{(e)}$ via trellis-based technique to minimize the weighted cost in (17), given the base layer is encoded with the current choice of $\mathcal{P}^{(b)}$.
   b) Optimize the base layer, i.e., find $\mathcal{P}^{(b)}$ via trellis-based technique to minimize the cost in (17), assuming that the residue will be encoded with the choice of $\mathcal{P}^{(e)}$ found in step 2(a).

Note that the Algorithm II is guaranteed to converge, as at each iteration, given the parameters of one layer, the trellis-based technique *optimally selects* the parameters of the other layer to minimize the *overall cost function*. That is, the overall cost at every iteration is monotonically non-increasing. However, note that the solution thus obtained is only guaranteed to be locally optimal.

The solution to the rate-constrained problem in (15) is derived from the solution to the distortion-constrained setting. We simply solve the distortion-constrained problem for multiple distortion constraints, and different values of the parameter α till we find a solution that meets the rate constraints in (15). This algorithm is summarized below:

Algorithm III: Scalable AAC—Rate-constrained optimization

1) Initialize the parameter α, and the distortion constraints $\mathcal{D}_t^{(b)}$ and $\mathcal{D}_t^{(e)}$.
2) Solve (17) for the given distortion constraints and parameter α via Algorithm II.
3) If the base layer rate constraint $\mathcal{R}_t^{(b)}$ is not met, change $\mathcal{D}_t^{(b)}$ and go to step 2.
4) If the cumulative rate constraint $\mathcal{R}_t^{(e)}$ is not met, change $\mathcal{D}_t^{(e)}$ and go to step 2.
5) Calculate the cost function of (15). Retain the solution if the cost has reduced compared to its previous stored value.
6) End algorithm if an exit condition is satisfied, else store current cost value, change α and repeat steps 2 to 5.

We initialize and update $\mathcal{D}_t^{(b)}$ and $\mathcal{D}_t^{(e)}$ via a method similar to the $\mathcal{D}_t^{(b)}$ updating algorithm described in Section IV-A. While α can be updated similarly, to simplify the experiments, we search for the best α within a fixed number of equally spaced values between 0 and 1. Note that as α is selected from a fixed set, Algorithm III is not subject to convergence concerns.

B. Joint Optimization of Layers in SLS

Similar to the case of scalable AAC, the proposed solution for SLS optimization first considers the intermediate problem (19), to eventually solve (15). However, here the intermediate problem (19) is solved without recourse to any iterative procedure. The proposed optimization employs one single-layer trellis for the AAC base-layer, which is modified to include the cost of encoding the SLS enhancement layer. Each node of this trellis is associated with a particular value of SF, and hence also with specific quantized base layer spectral data and 'mapped error' to be coded into the SLS layer for the SFB. Thus, each node of this trellis has a corresponding value of the MSB-plane $\mathcal{M}_n[l]$, and an optimal value of the lazy plane parameter $\mathcal{L}_n[l]$ obtained via the exhaustive search approach described in Section IV-B. Thus the node can also be associated with the number of SLS bits for the SFB estimated via the approximation in Section IV-B. The best encoding parameters (including SFs, HCBs, and the lazy plane parameters) are now found by minimizing the total weighted cost in (19) by the Viterbi algorithm. The overall optimization process for problem (19) is summarized below.

Algorithm IV: SLS—Distortion-constrained optimization

1) The nodes in the base layer trellis are populated with base layer distortion and rate costs corresponding to different SF and HCB values. Only states that satisfy the base layer distortion constraint in (20) are retained.
2) For each of these retained nodes the mapped error is computed and an exhaustive search is performed to find the optimal SLS parameters $\mathcal{L}_n[l]$. For this optimal $\mathcal{L}_n[l]$, the estimated enhancement layer bit-rate for coding the mapped error and the parameter $\mathcal{L}_n[l]$ is preserved.
3) Each state of the base layer trellis is now associated with a new cost, that is a weighted combination (via α) of the number of bits consumed in the two layers, to encode the intMDCT coefficients in the SFB.
4) State transitions are now associated with appropriate bit costs to differentially encode corresponding SFs, HCBs, and (MSB)-planes parameters.
5) Viterbi algorithm is employed to find the path through this trellis that minimizes the cost function in (20). The best path gives the optimal per-frame base layer parameters $\mathbf{p}_n^{(b)}$. This is repeated for each frame to obtain the optimal parameters $\mathcal{P}^{(b)}$.

The solution to the rate-constrained problem (15) is obtained by solving the distortion-constrained problem for multiple distortion levels, and different values of the parameter α until a solution that meets the rate constraint in (15) is obtained. The algorithm is summarized below:

Algorithm V: SLS—Rate-constrained optimization

1) Initialize the parameter α, and the distortion constraint $\mathcal{D}_t^{(b)}$.
2) Solve (19) for the given distortion constraint and parameter α via Algorithm IV.
3) If the base layer rate constraint $\mathcal{R}_t^{(b)}$ is not met, change $\mathcal{D}_t^{(b)}$ and go to step 2.
4) Calculate the cost function of (15). Retain the solution if the cost has reduced compared to its previous stored value.
5) End algorithm if an exit condition is satisfied, else store current cost value, change α and repeat steps 2 to 4.

We initialize and update $\mathcal{D}_t^{(b)}$ and α via a method similar to the scalable AAC case.

VI. Results

In this section, evaluation results of the proposed optimization methods are provided. First, results for the two-layered scalable AAC are discussed. Subsequently results for the SLS case are presented.

A. MPEG Scalable AAC

The experimental setting compared three coders:

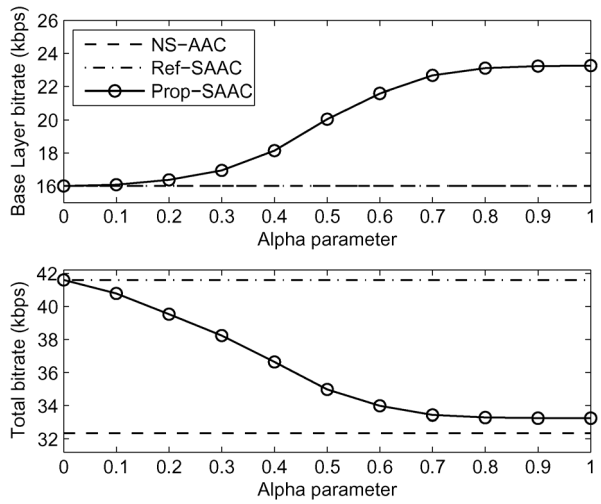


Fig. 2. Base layer bit-rate and total bit-rate of the two layers (averaged over the 4 test items) for scalable AAC experiments with distortion constraints. The proposed coder clearly provides a trade-off between optimality at the two layers.

- A single layer (non-scalable) AAC coder employing trellis based optimization (denoted NS-AAC).
- The reference scalable AAC coder described in Section IV-A that employs *separate optimization of layers* via a per-layer trellis (denoted Ref-SAAC).
- The proposed scalable AAC coder of Section V-A with *joint optimization of layers* (denoted Prop-SAAC).

All coders employed a simple psychoacoustic model with fixed signal-to-mask ratios similar to the MPEG-VM reference software. The coders were evaluated using a standard MPEG dataset, designed for the evaluation of low bit-rate audio coding. For computational and evaluation expediency, a subset of the dataset was created, by selecting one item per category, and extracting the first 5 seconds of each audio file (which are mono at 48 kHz). This resulted in the following test dataset:

- Speech signal: vocal (vega)
- Single instrument: harpsichord (harp)
- Simple sound mixture: plucked strings (stri)
- Complex sound mixture: orchestral piece (orch)

1) Distortion-Constrained Optimization Results: The first experiment considers the optimization problem with distortion constraints (17). The constraints were chosen as $\mathcal{D}_t^{(b)} = 7.1$ dB and $\mathcal{D}_t^{(e)} = 2.1$ dB, as using these in the non-scalable coder resulted in an average bit-rate (over the 4 audio items) of 16 and 32 kbps respectively. The proposed scalable coder was optimized for different values of the parameter $\alpha \in \{0.1, 0.2, \ldots, 1\}$ and the resultant base and total bit-rates averaged over the 4 audio items are shown in Fig. 2. For comparison, the figure also shows the bit-rates achieved by the non-scalable coder and the reference scalable coder.

The results confirm that the proposed scalable coder provides a tradeoff between base layer versus enhancement layer performance, in terms of rate required to achieve the prescribed distortion, which is controlled by the α parameter. The non-scalable coder bounds the rate achievable at each layer distortion. The reference scalable coder achieves one extreme of the tradeoff (optimality only at base layer).

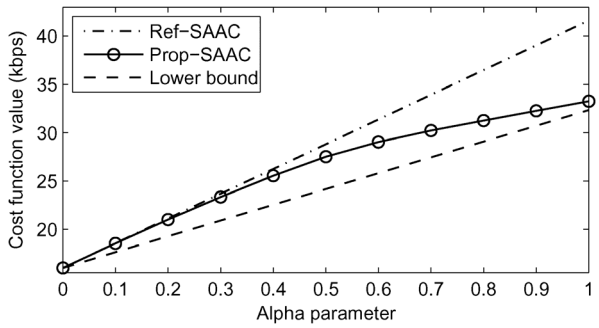


Fig. 3. Cost function value (averaged over the 4 test items) for scalable AAC experiments with distortion constraints. Note that the proposed coder cost is always lower than the reference coder cost.

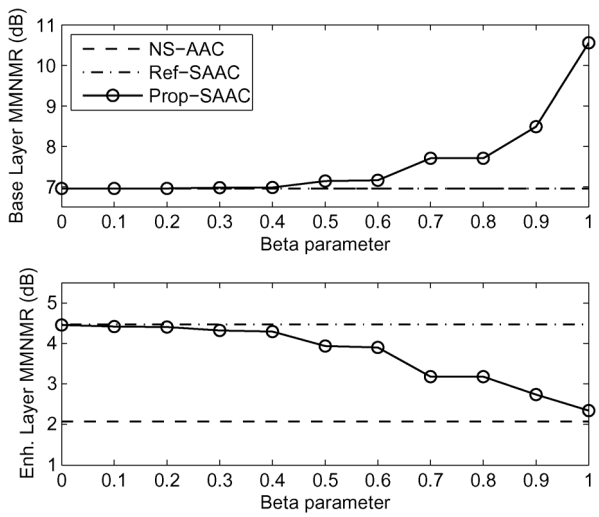


Fig. 4. Distortion at the base and enhancement layers (averaged over the 4 test items) for scalable AAC experiments with rate constraints. The proposed coder clearly controls the performance tradeoff between the two layers.

For completeness, in Fig. 3 we plot versus α the minimum distortion-constrained cost (17), achieved by the proposed scalable coder, as well as the corresponding cost curves for the reference scalable coder, and the lower bound given by the rates achieved by the non-scalable coder. Clearly, the proposed scalable coder always outperforms the reference scalable coder, with performance gains monotonically increasing with α.

2) Rate-Constrained Optimization Results: The second experiment considers the rate constrained problem (15), with constraints set at $\mathcal{R}_t^{(b)} = 16$ kbps and $\mathcal{R}_t^{(e)} = 32$ kbps. The proposed scalable coder was optimized for different values of the parameter $\beta \in \{0.1, 0.2, \ldots, 1\}$ using *Algorithm III* (described in Section V-A). The base and enhancement layer MMNMR distortions (averaged over the 4 audio items) achieved by the proposed scalable coder for different values of β are shown in Fig. 4. The figure also includes for comparison, the MMNMR distortions achieved by the non-scalable coder and the reference scalable coder.

Similar to the case of distortion-constrained optimization, these results demonstrate that the proposed scalable coder provides a tradeoff between base layer and enhancement layer performance, in terms of distortion achieved at the prescribed rate, which is controlled by parameter β. The non-scalable coder bounds the distortion achievable at each layer rate. The reference scalable coder achieves one extreme of the tradeoff (optimality only at base layer).

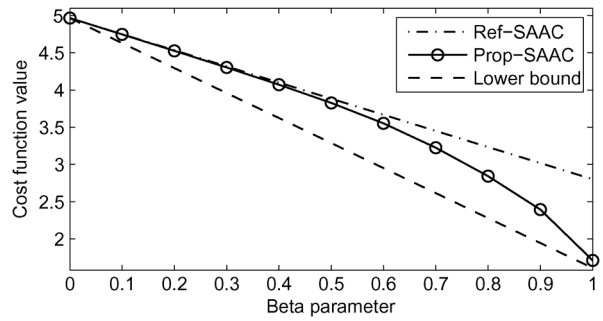


Fig. 5. Cost function (weighted sum of base and enhancement layer MMNMR) for scalable AAC experiments with rate constraints, averaged over the 4 test items. The y-axis is on a linear non-dB scale of MMNMR. Note that the proposed coder cost is always lower than the reference coder cost.

In Fig. 5 we plot versus β the minimum rate-constrained cost (15), achieved by the proposed scalable coder, as well as the corresponding cost curves for the reference scalable coder, and as lower bound the distortions achieved by the non-scalable coder. The figure clearly shows that the proposed scalable coder always outperforms the reference scalable coder, and that the difference increases with β.

Finally, a MUSHRA listening test was conducted to evaluate subjective quality. The proposed scalable coder (at $\beta = 0.8$ and $\beta = 1.0$) was compared with the reference scalable coder and the non-scalable coder. The following 9 versions of each of the 4 audio items were evaluated:

- Hidden reference (Ref)
- 3.5 kHz low-pass anchor (Anc)
- Base layer of the reference scalable coder at 16 kbps (same as the non-scalable coder at 16 kbps)
- Base layer of the proposed scalable coder with $\beta = 0.8$ at 16 kbps
- Base layer of the proposed scalable coder with $\beta = 1.0$ at 16 kbps
- Base + Enhancement layers of the reference scalable coder at cumulative rate of 32 kbps
- Base + Enhancement layers of the proposed scalable coder with $\beta = 0.8$ at cumulative rate of 32 kbps
- Base + Enhancement layers of the proposed scalable coder with $\beta = 1.0$ at cumulative rate of 32 kbps
- The non-scalable coder at 32 kbps

The test items were presented in random order to 10 expert listeners, and scored on a scale of 0 (bad) to 100 (excellent). The average scores for all items and the 95% confidence intervals are given in Fig. 6. The results clearly demonstrate the substantial loss in performance (of more than 20 points on MUSHRA scale) at the enhancement layer of the reference scalable coder compared to the non-scalable coder at the same bit-rate. In contrast, the proposed scalable coder can improve the enhancement layer performance by 10 MUSHRA points with $\beta = 0.8$, and about 20 MUSHRA points with $\beta = 1.0$, the latter statistically indistinguishable from the performance of the non-scalable coder. Note that the impact on the performance of the base layer is minimal at both $\beta = 0.8$ and $\beta = 1.0$.

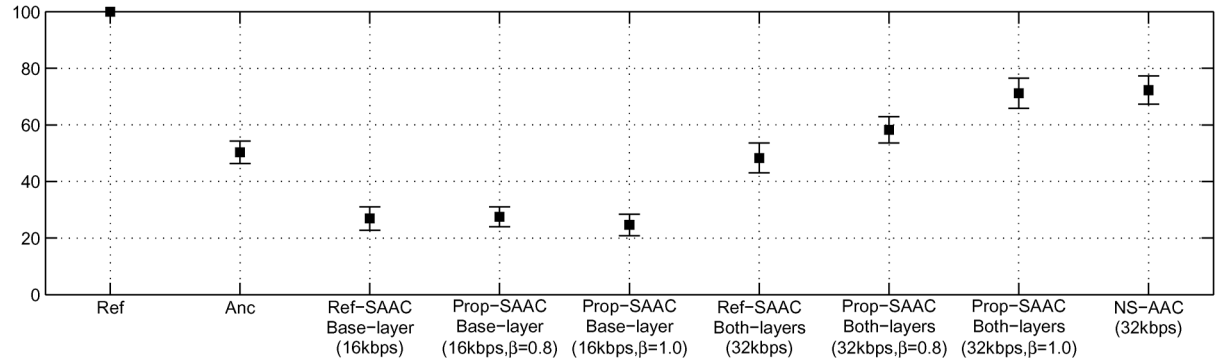


Fig. 6. MUSHRA listening test results for scalable AAC experiments: average scores for all items (with 95% confidence interval). For the scalable coders, bit-rates are cumulative up to the layer indicated.

B. MPEG SLS

In this experimental setting the following three coders were compared:

- The non-core SLS coder (noted NC-SLS).
- The reference SLS coder described in Section IV-B that employs *independent optimization of layers* (noted Ref-SLS).
- The proposed SLS coder of Section V-B with *joint optimization of layers* (noted Prop-SLS).

The AAC core used in the reference SLS coder and the proposed SLS coder employs the same psychoacoustic model as the scalable AAC experiments. All the three encoders use the "optimal" selection of the $\mathcal{L}_n[l]$ parameters in the SLS enhancement layer (as introduced in Section IV-B). In the distortion-constrained setting, i.e., when $\mathcal{D}_t^{(b)}$ is constrained for the reference and proposed SLS coders, the "base-layer" for the non-core SLS is chosen by truncating the SLS bitstream such that for each frame the distortion constraint is satisfied, and the corresponding "base-layer" bit-rate is calculated as the average the number of bits per-frame in the truncated bitstream. In the rate-constrained setting, with $\mathcal{R}_t^{(b)}$ as the base-layer bit-rate constraint, one approach to generate the "base-layer" for the non-core SLS bitstream could be to simply truncate each SLS frame to have exactly $\mathcal{R}_t^{(b)}$ number of bits (i.e., constant bit-rate), and the resulting MMNMR (across SFBs and frames) can be used for comparison. However, this would be unfair to the non-core SLS coder as it would not benefit from a variable bit-rate base layer, as the other coders do. Thus, in order to mimic a variable bit-rate base-layer (with a target average bit-rate of $\mathcal{R}_t^{(b)}$) in the non-core SLS case, we implemented the following approach: the SLS bitstream is truncated under a distortion constraint $\mathcal{D}_\tau^{(b)}$ and the average bit-rate calculated. If the rate constraint $\mathcal{R}_t^{(b)}$ is not met, the distortion constraint $\mathcal{D}_\tau^{(b)}$ is changed and the SLS bitstream re-truncated. When the target constraint $\mathcal{R}_t^{(b)}$ is met, $\mathcal{D}_\tau^{(b)}$ is the "base-layer" MMNMR of the non-core SLS coder that is compared against that of the reference and proposed SLS coders.

The experiments were conducted with audio data test designed for scalable-to-lossless audio coding (as used in previous work on SLS [4], [9]). This testing set originates from the MPEG lossless audio coding task group [27] and consists of 15 audio sequences each 30 seconds long, single channel and sampled at 48 kHz.

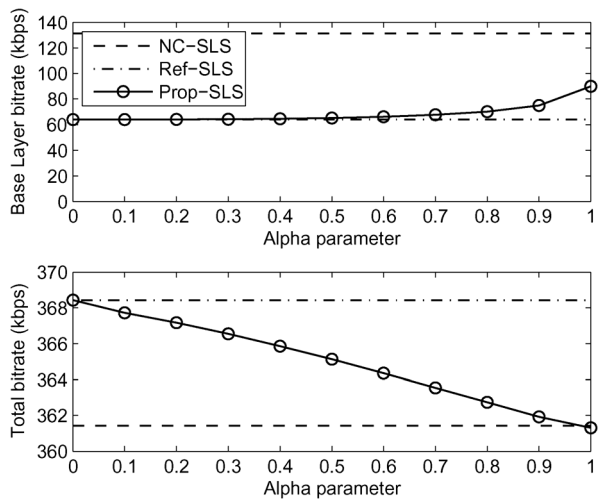


Fig. 7. Base layer rate and total lossless compression rate (averaged over all test items) for SLS experiments with distortion constraint. The proposed coder clearly controls the performance tradeoff between the two layers.

1) Distortion-Constrained Optimization Results: This experiment considers the distortion-constrained optimization problem (19). The constraint for the base layer was chosen as $\mathcal{D}_t^{(b)} = -3$ dB, as using this in the reference SLS coder resulted in an average bit-rate of 64 kbps. The proposed SLS coder was optimized for different values of the parameter $\alpha \in \{0.1, 0.2, \ldots, 1\}$ and the resultant base layer and total bit-rates averaged over all audio items are shown in Fig. 7. For comparison, the figure also includes bit-rates achieved by the non-core SLS coder and the reference SLS coder.

The results first show that, at the base layer, the non-core SLS coder substantially underperforms the reference SLS coder, as it is not optimized to minimize perceptual distortion, and vice versa at the enhancement layer. Then the results confirm that the proposed SLS coder provides a tradeoff between base layer performance, in terms of rate required to achieve the prescribed distortion, and the lossless compression rate, which is controlled by parameter α, whereas the reference SLS coder implements the extreme tradeoff point achieving optimality only at the base layer.

In Fig. 8 we plot, versus α, the difference in distortion-constrained cost (19), relative to the reference SLS coder, achieved by the proposed SLS coder and by the non-core SLS coder. The

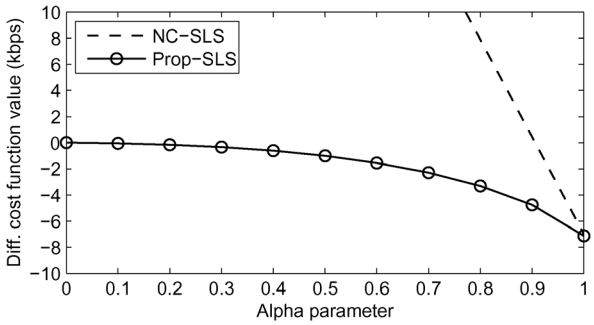


Fig. 8. Distortion-constrained cost differences relative to reference SLS coder, averaged over all test items. Note that the proposed coder cost is always lower than both reference and non-core SLS coder costs.

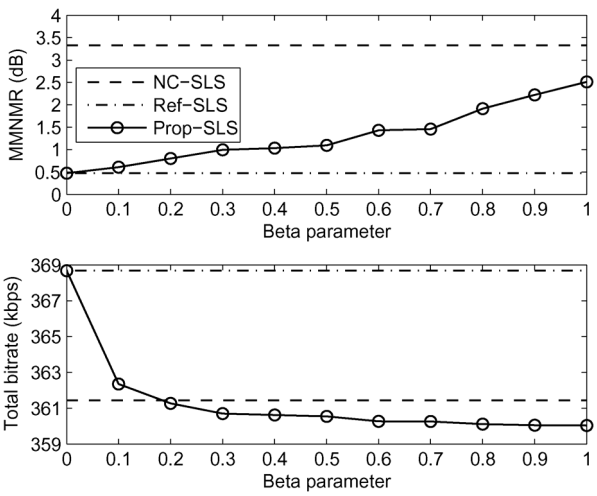


Fig. 9. Distortions at the base layer and total lossless compression rate (averaged over all test items) for SLS experiments with rate constraint. The proposed coder clearly controls the performance tradeoff between the two layers.

figure clearly shows that the proposed SLS coder always outperforms the reference SLS coder in terms of the cost function, with performance gains monotonically increasing with α. Moreover, it always outperforms even the non-core SLS in terms of the same metric.

2) Rate-Constrained Optimization Results: The next experiment considers the problem with base layer rate constraint (15), set at $\mathcal{R}_t^{(b)} = 64$ kbps. The proposed SLS coder was optimized for different values of the parameter $\beta \in \{0.1, 0.2, \ldots, 1\}$ using *Algorithm V* (described in Section V-B). The base layer MMNMR distortions and the total lossless compression rate achieved by the proposed SLS coder for different values of β are shown in Fig. 9. The figure also includes for comparison the MMNMR distortions and total rates achieved by the non-core SLS coder and the reference SLS coder.

Similar to the case of distortion-constrained optimization, the results first show that, at the base layer, the non-core SLS coder substantially underperforms the reference SLS coder as it is not optimized to minimize perceptual distortion, and vice versa at the enhancement layer. Then the results confirm that the proposed SLS coder provides a tradeoff between base layer performance, in terms of distortion achieved at the prescribed rate, and the overall lossless compression rate, which is controlled by parameter β, whereas the reference SLS coder implements

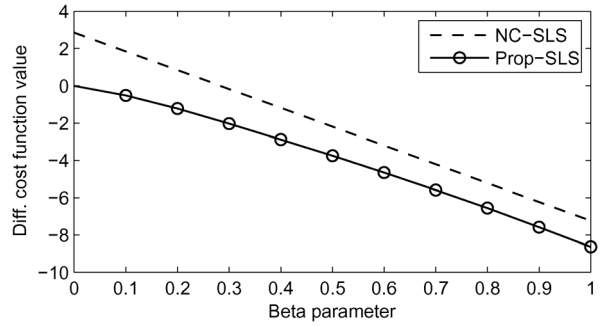


Fig. 10. Rate-constrained cost differences relative to reference SLS coder, averaged over all test items. Note that the proposed coder cost is always lower than both reference and non-core SLS coder costs.

the extreme tradeoff point achieving optimality only at the base layer.

In Fig. 10 we plot, versus β, the difference in rate-constrained cost (15), relative to the reference SLS coder, achieved by the proposed SLS coder and by the non-core SLS coder. It is evident that the proposed SLS coder always outperforms the reference SLS coder (and also non-core SLS) in terms of the cost function optimized, with performance gains monotonically increasing with β.

In the next experiment, we evaluated the base layer quality with a more realistic perceptual measure of Objective Difference Grade (ODG) given by the PEAQ method (ITU-R BS.1387-1 [28], as implemented in the AFsp library [29]), and the lossless performance was evaluated via the compression ratio. Note that similar evaluation was conducted in previous SLS work (e.g., [4]). The average and individual results of all files at base layer constraint of 64 kbps, for the non-core SLS coder, the reference SLS coder, and the proposed SLS coder (at $\beta = 0.2$ and $\beta = 1.0$) is given in Table I.

The results again confirm that the non-core SLS coder's base layer quality is substantially worse than that of the reference SLS coder, while its lossless performance (the compression ratio) is better. The results also show that the proposed SLS coder at $\beta = 0.2$ can achieve 'best of both worlds' with same excellent base layer quality as the reference SLS coder and same lossless compression ratio as the non-core SLS coder. Then by increasing β, the proposed SLS coder continues to trade-off base layer quality to achieve even better lossless compression ratio.

Finally, a MUSHRA listening test was conducted to evaluate subjective quality. The base-layer of the proposed SLS coder (at $\beta = 0.2$ and $\beta = 1.0$) was compared with the base-layer of the reference SLS coder and the non-core SLS coder decoded at the corresponding rate. Amongst the files used for the PEAQ evaluation, we conducted the listening tests with 6 audio items: 3 of these items had better PEAQ scores for the reference SLS coder (cherokee, dcymbals and waltz) and 3 of them had better PEAQ scores for the proposed SLS coder with $\beta = 0.2$ (etude, flute and violin). 10 listeners participated in the test. The average scores for all items and the 95% confidence intervals are given in Fig. 11. The results clearly indicate that the proposed SLS coder with $\beta = 0.2$ achieves subjective quality statistically similar to the reference SLS coder and outperforms the non-core SLS coder, corroborating the PEAQ results in achieving 'best

TABLE I
COMPARISON OF THE REFERENCE SCALABLE SLS CODER, NON-CORE SLS CODER, AND THE PROPOSED SLS CODER (UNDER TWO OPTIMIZATION SETTINGS: $\beta = 0.2$ AND $\beta = 1.0$): THE BASE-LAYER (64 kbps) OF THE CODECS ARE COMPARED IN TERMS OF ODG SCORES W.R.T THE UNCODED ORIGINAL AS CALCULATED BY PEAQ. THE LOSSLESS ENHANCEMENT-LAYER OF THE CODECS IS COMPARED IN TERMS OF THE COMPRESSION RATIO

	ODG measurements				Compression ratio			
	Ref-SLS	**Prop-SLS** ($\beta = 0.2$)	**Prop-SLS** ($\beta = 1.0$)	**NC-SLS**	**Ref-SLS**	**Prop-SLS** ($\beta = 0.2$)	**Prop-SLS** ($\beta = 1.0$)	**NC-SLS**
avemaria	-0.822	-0.688	-1.090	-2.552	2.510	2.569	2.577	2.575
blackandtan	-0.959	-1.021	-2.604	-2.572	1.766	1.796	1.802	1.794
broadway	-1.372	-1.358	-2.917	-3.393	1.876	1.914	1.922	1.907
cherokee	-0.888	-1.088	-2.192	-2.549	1.842	1.870	1.876	1.872
clarinet	-0.894	-0.786	-1.455	-2.134	2.059	2.105	2.113	2.100
cymbal	-1.090	-1.180	-1.180	-1.295	3.091	3.150	3.150	3.191
dcymbals	-0.766	-0.909	-3.048	-3.052	1.624	1.647	1.654	1.649
etude	-0.954	-0.807	-1.859	-3.062	2.303	2.359	2.369	2.357
flute	-0.987	-0.729	-0.975	-3.338	2.442	2.514	2.522	2.506
fouronsix	-1.107	-1.166	-1.945	-2.471	2.122	2.163	2.170	2.162
haffner	-0.893	-0.937	-2.493	-2.944	1.770	1.804	1.811	1.798
mfv	-0.213	-0.276	-0.276	-0.497	3.213	3.299	3.299	3.332
unfo	-1.174	-1.240	-2.440	-2.597	1.938	1.979	1.987	1.974
violin	-1.004	-0.863	-1.916	-3.174	2.030	2.083	2.092	2.069
waltz	-1.014	-1.128	-2.397	-2.412	1.865	1.900	1.907	1.897
Overall	**-0.942**	**-0.945**	**-1.919**	**-2.536**	**2.083**	**2.126**	**2.133**	**2.125**

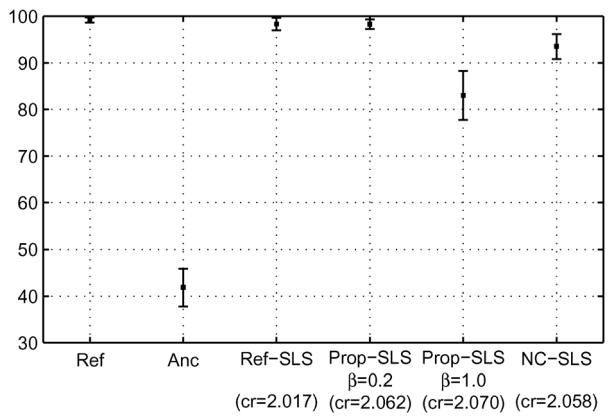


Fig. 11. MUSHRA listening test results comparing base-layer quality of the SLS coders: average scores for all items (with 95% confidence interval). The compression ratio (cr) for each SLS coder is also given.

of both worlds'. The proposed SLS coder can further trade base layer quality for improved lossless compression ratio, which is highlighted by the MUSHRA scores for $\beta = 1.0$, where the base layer quality of the proposed SLS coder is indeed lower than that of the non-core SLS coder albeit at a significantly improved compression ratio.

C. Complexity

The algorithm complexity is evaluated by measuring the computation time on a recent computer (Intel Core i5 750 @ 2.67 GHz, 6 GB RAM). The computation times averaged over all test sequences and normalized by the average test file length, is provided in Table II.

The computation times for distortion-constrained optimization show that the proposed scalable AAC coder is 3 times more complex than the reference scalable AAC coder, which itself is 3 times more complex than the non-scalable AAC coder, and the proposed SLS coder is 2.5 times more complex than the reference SLS coder, which itself is 4 times more complex than the non-core SLS coder.

TABLE II
COMPUTATION TIMES, AVERAGED OVER ALL TEST SEQUENCES AND NORMALIZED BY THE AVERAGE TEST FILE LENGTH

	Distortion-constrained	**Rate-constrained**
Ref-SAAC	0.32	5.52
Prop-SAAC	1.03	79.96
NS-AAC	0.11	1.16
Ref-SLS	0.39	4.83
Prop-SLS	0.99	11.65
NC-SLS	0.09	0.94

For the rate-constrained problem, we evaluated the computation times of the proposed joint-optimization approach for a single α value. Results show that the proposed scalable AAC coder is 15 times more complex than the reference scalable AAC coder, which itself is 5 times more complex than the non-scalable AAC coder, and the proposed SLS coder is 2.5 times more complex than the reference SLS coder, which itself is 5 times more complex than the non-core SLS coder. Note that to get the β-optimized solution for the proposed coder, its computation time will scale up by the number of iterations over α. While we perform an exhaustive search over equally spaced fixed number of α values, a more efficient solution with very few iterations could be used in practice.

VII. CONCLUSION

This paper proposes a novel approach to scalable audio encoding that jointly optimizes the parameters of all layers via a single cost function incorporating the relative importance of different layers, in contrast with the common practice of optimizing each layer successively and regardless of impact on higher layers. The proposed approach is applied in conjunction with two standard scalable audio coding formats, namely scalable AAC and MPEG SLS. Experimental results for scalable AAC show substantial performance gains, in terms of objective and subjective quality metrics, over the commonly employed "myopic" successive optimization of layers, at the cost of a reasonable increase in complexity. Results for the SLS codec

demonstrate that improvement in lossless compression can be achieved at minimal compromise of the perceptual quality of the AAC layer, and conversely that the presence of a superior AAC core does not preclude excellent lossless compression performance.

REFERENCES

[1] *Information Technology—Coding of Audio-Visual Objects—Part 3: Audio—Subpart 4: General Audio Coding (GA)*, ISO/IEC std. ISO/IEC JTC1/SC29 14496-3:2005, 2005.

[2] B. Grill, "A bit rate scalable perceptual coder for MPEG-4 audio," in *Proc. 103rd AES Conv.*, Sep. 1997, Preprint 4620.

[3] S.-H. Park, Y.-B. Kim, S.-W. Kim, and Y.-S. Seo, "Multi-layer bit-sliced bit-rate scalable audio coding," in *Proc. 103rd AES Conv.*, Oct. 1997, Preprint 4520.

[4] R. Yu, S. Rahardja, L. Xiao, and C. C. Ko, "A fine granular scalable to lossless audio coder," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 4, pp. 1352–1363, Jul. 2006.

[5] *MPEG-4 Audio Verification Test Results: Audio on Internet*, ISO/IEC JTC1/SC29/WG11/MPEG98/N2425, Oct. 1998.

[6] A. Aggarwal, S. L. Regunathan, and K. Rose, "Trellis-based optimization of MPEG-4 advanced audio coding," in *Proc. IEEE Workshop Speech Coding*, 2000, pp. 142–144.

[7] A. Aggarwal, S. L. Regunathan, and K. Rose, "A trellis-based optimal parameter value selection for audio coding," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 2, pp. 623–633, Mar. 2006.

[8] V. Melkote and K. Rose, "Trellis-based approaches to rate-distortion optimized audio encoding," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 18, no. 2, pp. 330–341, Feb. 2009.

[9] R. Geiger, R. Yu, J. Herre, S. Rahardja, S.-W. Kim, X. Lin, and M. Schmidt, "ISO/IEC MPEG-4 High-Definition Scalable Advanced Audio Coding," *J. Audio Eng. Soc.*, vol. 55, no. 1/2, pp. 27–43, Jan./Feb. 2007.

[10] E. Ravelli, V. Melkote, T. Nanjundaswamy, and K. Rose, "Cross-layer rate-distortion optimization for scalable advanced audio coding," in *Proc. 128th AES Conv.*, May 2010, Preprint 8084.

[11] E. Ravelli, V. Melkote, T. Nanjundaswamy, and K. Rose, "Joint optimization of the perceptual core and lossless compression layers in scalable audio coding," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Apr. 2010, pp. 365–368.

[12] C.-H. Yang and H.-M. Hang, "Cascaded trellis-based rate-distortion control algorithm for MPEG-4 advanced audio coding," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 3, pp. 998–1007, May 2006.

[13] C. Bauer and M. Vinton, "Joint optimization of scale factors and huffman codebooks for MPEG-4 AAC," in *Proc. 6th IEEE Workshop Multimedia Signal Process.*, Sep. 2004, pp. 111–114.

[14] A. Aggarwal, S. L. Regunathan, and K. Rose, "Efficient bit-rate scalability for weighted squared error optimization in audio coding," *IEEE Trans. Audio, Speech, and Lang. Process.*, vol. 14, no. 4, pp. 1313–1327, Jul. 2006.

[15] B. Leslie, C. Dunn, and M. Sandler, "Developments with zerotree audio codes," in *Proc. AES 17th Int. Conf.*, Sep. 1999.

[16] C. Dunn, "Efficient audio coding with fine grain scalability," in *Proc. 111th AES Conv.*, Sep. 2001, preprint 5492.

[17] S. Kandadai and C. D. Creusere, "Perceptually-weighted audio coding that scales to extremely low bit-rates," in *Proc. IEEE Data Compression Conf.*, Mar. 2006, pp. 382–391.

[18] H. Schwarz and T. Weigand, "R-D optimized multi-layer encoder control for SVC," in *Proc. Int. Conf. Image Process.*, Sep. 2007, vol. 2, pp. 281–284.

[19] X. Li, P. Amon, A. Hutter, and A. Kaup, "One-pass multi-layer rate-distortion optimization for quality scalable video coding," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, Apr. 2010, pp. 637–640.

[20] *Information Technology—Coding of Audio-Visual Objects—Part 3: Audio—Amd. 3: Scalable Lossless Coding (SLS)*, ISO/IEC std. ISO/IEC JTC1/SC29 14496-3:2005/Amd.3:2006, 2006.

[21] O. Derrien, "A new quantization optimization algorithm for the MPEG advanced audio coder using a statistical subband model of the quantization noise," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 14, no. 4, pp. 1328–1339, Jul. 2006.

[22] *Information Technology—Coding of Audio-Visual Objects—Part 3: Audio—Amd. 10: HD-AAC Profile*, ISO/IEC std. ISO/IEC JTC1/SC29 14496-3:2005/Amd.10:2008, 2008.

[23] K. Brandenburg, "Evaluation of quality for audio encoding at low bit rates," in *Proc. 82nd AES Conv.*, Mar. 1987, Paper 2433.

[24] K. Brandenburg and T. Sporer, "NMR and masking flag: Evaluation of quality using perceptual criteria," in *Proc. AES 11th Int. Conf.*, May 1992.

[25] M. Bosi, K. Brandenburg, S. Quackenbush, L. Fielder, K. Akagiri, H. Fuchs, M. Dietz, J. Herre, G. Davidson, and Y. Oikawa, "ISO/IEC MPEG-2 Advanced Audio Coding," *J. Audio Eng. Soc*, vol. 45, no. 10, pp. 789–814, Oct. 1997.

[26] 3GPP TS 26.403, General Audio Codec Audio Processing Functions; Enhanced aacPlus General Audio Codec; Encoder Specification; Advanced Audio Coding (AAC) Part, 2004.

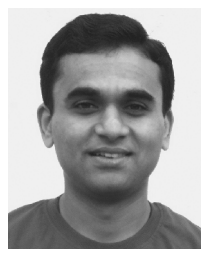
[27] "Call for proposals on MPEG-4 lossless audio coding," in *Proc. 61st MPEG Meeting, Klagenfurt, Austria*, Jul. 2002, ISO/IEC JTC1/SC29/WG11 N5040.

[28] *Method for Objective Measurements of Perceived Audio Quality (PEAQ)*, ITU-R Rec. BS.1387-1, 2001.

[29] P. Kabal, Audio File Programs and Routines, [Online]. Available: http://www-mmsp.ece.mcgill.ca/Documents/Downloads/AFsp/

Emmanuel Ravelli graduated from ENSEEIHT, Toulouse, France, as an electronics and signal processing engineer in 2004, and received the M.Sc. degree from the University Paul Sabatier, Toulouse, France, in 2005 and the Ph.D. degree from the Pierre et Marie Curie University-Paris 6, Paris, France, in 2008. He was a Postdoctoral Researcher with the Department of Electrical and Computer Engineering, University of California, Santa Barbara in 2009. He is currently with the International Audio Laboratories, Fraunhofer IIS, Erlangen, Germany.

Vinay Melkote (S'08–M'10) received the B.Tech degree in electrical engineering from the Indian Institute of Technology Madras, Chennai, India, in 2005 and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of California, Santa Barbara, in 2006 and 2010, respectively. He is currently with the Sound Technology Research Group, Dolby Laboratories, Inc., San Francisco, CA, where he focuses on audio compression and related technologies. He was a recipient of the Best Student Paper Award at the IEEE International Conference on Acoustics, Speech, and Signal Processing 2009, and is a member of the IEEE Signal Processing Society's technical committee for Audio and Acoustic Signal Processing.



Tejaswi Nanjundaswamy (S'11) received the M.S. degree in electrical and computer engineering from the University of California, Santa Barbara (UCSB), in 2009. He is currently pursuing the Ph.D. degree in electrical and computer engineering at UCSB. His research interests include audio and speech processing/coding.

Mr. Nanjundaswamy is a student member of the Audio Engineering Society (AES). He won the Student Technical Paper Award at the AES 129th Convention.

Kenneth Rose (S'85–M'91–SM'01–F'03) received the Ph.D. degree in 1991 from the California Institute of Technology. He is a Professor of Electrical and Computer Engineering at the University of California, Santa Barbara. His main research activities are in the areas of information theory and signal processing, and include rate-distortion theory, source and source-channel coding, audio and video coding and networking, pattern recognition, and nonconvex optimization. Dr. Rose was co-recipient of the 1990 William R. Bennett Prize Paper Award of the IEEE Communications Society, as well as the 2004 and 2007 IEEE Signal Processing Society Best Paper Awards.