Audio Engineering Society

Convention Paper 6183

Presented at the 117th Convention
2004 October 28–31 San Francisco, CA, USA

MPEG-4 Scalable to Lossless Audio Coding

Rongshan Yu[1], Ralf Geiger[2], Susanto Rahardja[1], Juergen Herre[3], Xiao Lin[1], and Haibin Huang[1]

[1] Institute for Infocomm Research, 21 Heng Mui Keng Terrace, Singapore
{rsyu,rsusanto,linxiao,hhuang}@i2r.a-star.edu.sg

[2] Fraunhofer IDMT, Ilmenau, Germany
ggr@idmt.fraunhofer.de

[3] Fraunhofer IIS, Erlangen, Germany
hrr@iis.fraunhofer.de

ABSTRACT

As the latest extension of MPEG-4 Audio coding, MPEG-4 Lossless Audio Coding includes a scalable audio coding solution (SLS) that integrates the functionalities of lossless audio coding, perceptual audio coding, and fine granular scalable audio coding into a single coder framework while providing backward compatibility to MPEG Advanced Audio Coding (AAC) at the bit-stream level. Despite its abundant functionalities, SLS still achieves a compression performance that is comparable to state-of-the-art non-scalable lossless audio coding algorithms. As a result, SLS provides a universal digital audio format for a variety of application domains including professional audio, Internet music, consumer electronics, broadcasting and others. This paper presents the structure of SLS and its latest developments during the MPEG standardization process.

1. INTRODUCTION

The digital audio format has now essentially superceded its analog counterpart for audio applications due to its unprecedented quality and flexibility. However, the large bit-rates of digital audio signals, e.g. 705.6 kbps/channel for a CD quality digital audio signal sampled at 44.1 kHz with 16 bit/sample word length, could be a heavy burden for many applications with constrained bandwidth or storage resources. For this reason, considerable effort has been devoted to the development of audio compression algorithms. Most audio compression algorithms, such as MPEG-1 Layer III (mp3) [1] and MPEG-2/4 AAC [2], belong to the category of lossy compression where the original audio signal is modified after compression. However, perceptual coding techniques [3] are usually employed in these lossy audio compression algorithms to minimize the perceptual effects of the introduced distortion and, possibly, to achieve "transparent" audio quality such that distortion introduced during compression is inaudible to the human auditory system. Nowadays many perceptual audio compression algorithms can achieve excellent compression ratio performance, e.g., 7 ~ 14:1 times compression, while maintaining the transparent quality in the compressed audio. However, due to their lossy nature, these algorithms are not suitable for high-end applications with lossless reconstruction requirement, e.g., studio archiving applications. They are also not suitable for applications where the compressed audio is expected to be post-processed, as the post-processing may alter the spectral characteristics of the compressed audio and hence make the compression noise no longer inaudible (masked). For these cases the desired solutions are audio compression algorithms that support lossless or near-lossless compression.

Recently, with the advances in broadband access networking and storage technologies there are an increasing number of digital audio applications that could provide high quality audio services with high sampling rate, high amplitude resolution (e.g., 96 kHz, 24 bit/sample) audio and lossless quality. Meanwhile, there will still be many applications that require highly compressed digital audio. Clearly, a solution that provides interchangeability across these two application domains would greatly simplify the problem of migrating audio content in these domains, and facilitate the transition from lossy to lossless digital audio service. In response to this need, the international standard body ISO/IEC JTC1/SC29/WG11, also known as the Moving Picture Experts Group (MPEG), has recently issued a Call for Proposal (CfP) [4] on lossless audio coding to invite contributions for a solution for scalable to lossless (SLS) audio compression. The Reference Model (RM) [5] for the MPEG-4 SLS work was selected at the MPEG meeting in July 2003, and its compression performance has been significantly improved through a Core Experiment (CE) process subsequently.

In this paper, we present the overall structure for MPEG-4 SLS. Unlike most lossless audio codecs that use the predictive coding approach, the MPEG-4 SLS codec employs a transform coding approach that is based on the Integer Modified Discrete Cosine Transform (IntMDCT) [6]. In order to achieve the backward compatibility to existing MPEG-4 lossy audio codecs, the MPEG-4 SLS codec adopts a two-layer structure where the core layer is an MPEG-4 AAC encoder which generates the AAC compliant bit-stream embedded in the final lossless bit-stream. This scalable perceptual and lossless coding approach in frequency domain has been introduced in [24],[25]. In addition to this, MPEG-4 SLS provides a fine grain scalability from lossy to lossless, which is achieved by a lossless enhancement (LLE) bit-stream which is generated with bit-plane coding technique. Moreover, in order to achieve best perceptual quality at intermediate rates when the LLE bit-stream is truncated, the order of bit-plane coding is selected in such a way that the spectral shape of the quantization noise of the core AAC bit-stream, which has been perceptually optimized during the AAC coding process, is preserved during the sequential bit-plane coding. In this way, an increasing noise-to-mask ratio headroom is added with each additional bit-plane. This allows for a flexible near-lossless signal representation with constant bit-rate.

It is worth to note that the performance of the MPEG-4 SLS codec has been significantly improved during the collaborative phase for this MPEG work item. Several key techniques have contributed to this improvement, including the Multi-Dimensional Lifting (MDL) scheme based IntMDCT implementation [7], refinements in entropy coder design such as Low energy mode coding [8], context-based bit-plane arithmetic coding [9], and over-sampling IntMDCT techniques [10].

This paper is organized as follows. Before going into the implementation details of MPEG-4 SLS, several possible application scenarios for this new MPEG Audio tool are listed in Section 2. The structure of MPEG-4 SLS codec is briefed in Section 3; and more detailed descriptions for the blocks used in MPEG-4 SLS are elaborated in the subsequent sections. These include Section 4 on the Integer MDCT, Section 5 on the error mapping process and Section 6 on the entropy coder design. The performance of MPEG-4 SLS is presented in Section 6. Finally, Section 7 concludes this paper.

2. APPLICATION SCENARIOS

2.1. Studio Operations

The MPEG-4 SLS provides a good solution for storage of audio at various points in the studio operations such as recording, editing, mixing and pre-mastering, as studio procedures are designed to preserve the highest levels of quality. In addition, it also gives a standardized solution for transferring audio between sites for collaborative working, which is increasingly common in studio. The scalability of MPEG-4 SLS also makes it suitable for situations where the bandwidth is not sufficient to support lossless compression. In addition, the AAC bit-streams that are contained in the

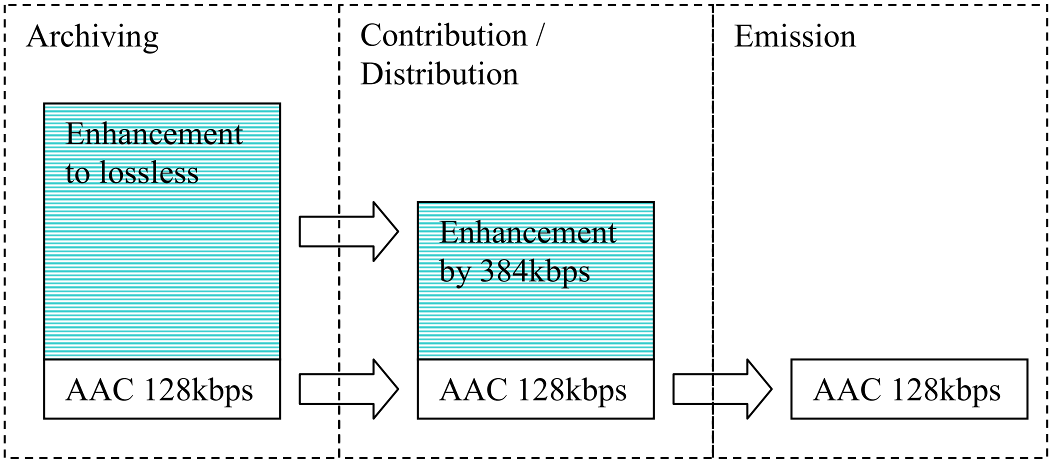

Figure 1. MPEG-4 Audio SLS in the broadcast chain

MPEG-4 SLS lossless streams can be very useful for transferring "previews" for work in progress, or remote monitoring of a recording session in real-time over low capacity networks.

2.2. Archival

Archives of sound recordings are very common in studios, record labels, libraries, and etc. MPEG-4 SLS provide the lossless compression capability, which is important to these archiving systems. In addition, the scalability of MPEG-4 SLS enables the possibility that low bit-rate versions of the archives lossless audio items can be extracted at any time to allow applications such as remote data browsing.

2.3. Consumer disc-based delivery

The MPEG-4 SLS can be used in consumer disc-based delivery applications; and its scalability feature enables delivering both lossless and lossy audio on the same disc in a very efficient way compared with simulcasting solutions such as DVD-Audio.

2.4. Internet Delivery of Files

The online music downloading service has attracted more and more attention recently thanks to the success of the Apple iTune service, and others. In such an application the bandwidth condition can vary dramatically over different access network technologies. As a result, same audio contents at a variety of bitrates and qualities may need to be provisioned at the server side and MPEG-4 SLS provide a nice "one-file" solution for this requirement. In addition, it also provides the possibility that user may initially download a low-bit-rate file, and "pay for quality" and download an enhancement later.

2.5. Streaming Applications

MPEG-4 SLS delivers the fine granular bit-rate scalability, which makes it an ideal solution for streaming applications on channel with variable QoS conditions. Examples for this type of streaming applications include the Internet audio streaming, multicast streaming applications that feeds several channels of differing capacity, possibly with a parser at the gateway controlling feed rates, and mobile streaming applications.

2.6. Broadcast Contribution/Distribution Chain

In a broadcast environment, MPEG-4 SLS could be used in all stages comprising archiving, contribution/distribution and emission. For archiving the codec can operate in a lossless way, for contribution/distribution a constant, high bit rate (e.g. 512 kbps) can be used, and finally the AAC core can be used for emission. This is illustrated in Figure 1.

In this broadcast chain, one main feature of the MPEG-4 SLS architecture can be used: In every stage where lower bit rates are required, the bit stream is just truncated, and no re-encoding is therefore required. Clearly, re-encoding is still required after post-processing steps, but for the sake of transcoding to

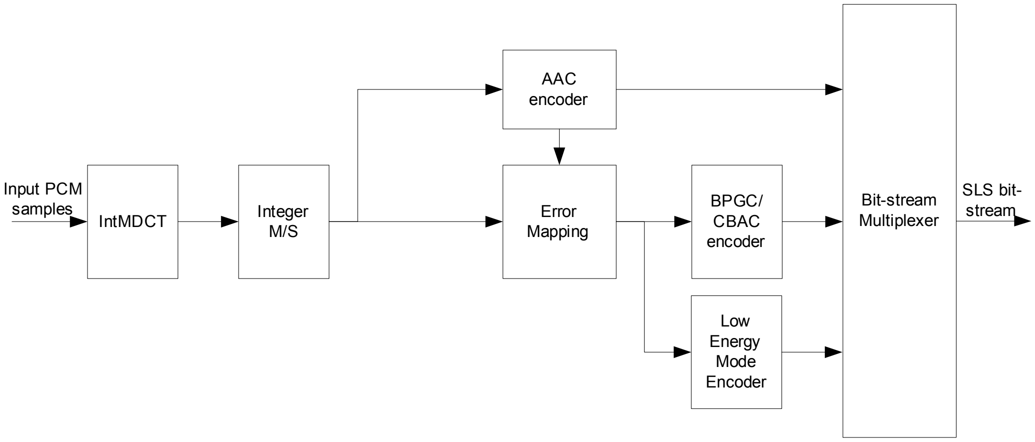

Figure 2. Diagram of MPEG-4 Audio SLS encoder

lower bit rates only a truncation, and no re-encoding is required.

3. STRUCTURE OF MPEG-4 SLS

The MPEG-4 SLS codec adopts Advanced Audio Zip (AAZ) [11] as Reference Model. It utilizes the IntMDCT based lossless coding approach where the IntMDCT spectral data are coded with two complementary layers, namely, a core MPEG-4 AAC layer which generates an AAC compliant bit-stream at a pre-defined bit-rate which constitutes the minimum rate/quality unit of the lossless bit-stream, and a Lossless Enhanced (LLE) layer that makes use of bit-plane coding method to produce the fine grain scalable to lossless portion of the lossless bit-stream. The structure of MPEG-4 SLS encoder is shown in Figure 2.

The core-layer AAC encoder in MPEG-4 SLS follows the informative AAC encoding specification described in [2][12]. The information that has been coded in the core AAC bit-stream is then removed from the IntMDCT spectral data by an error mapping process; and the resulted IntMDCT spectral residuals are coded in the LLE encoder. This error mapping process also manages to preserve the probability distribution skew of the original IntMDCT coefficients, which is approximately Lapacian distributed, in the IntMDCT residuals so that they can be very efficiently coded by the entropy coder used in the LLE layer.

In particular, for high sampling rates (96 kHz and higher) inputs, the performance of this scalable system is further enhanced by the so-called oversampling technique. By this approach the LLE encoder can operate at a preferable longer transform length while the AAC core codec can operate at a more appropriate, lower sampling rate. For example, for an oversampling factor (osf) of 2, the AAC core can operate at 48 kHz while the LLE encoder operates at 96 kHz with the frame length doubled compared to the AAC core (i.e. 2048 samples). In this case the lower 1024 IntMDCT spectral values can be used as an approximation of the MDCT spectral values necessary for the AAC encoder. The error mapping process is still valid. In this case the quantized AAC spectrum is mapped to the lower part of the oversampled IntMDCT spectrum.

The MPEG-4 SLS codec provides a non-core mode for applications that require only lossless quality. This is achieved by simply disabling the AAC core used in the MPEG-4 SLS codec. It is found in our test that the lossless compression performance of MPEG-4 SLS will be improved by 1% ~ 5% if the AAC core is not presented, and its computational complexity as well as the implementation cost has also been significantly reduced since the AAC encoder and decoder do not need to be implemented in the MPEG-4 SLS non-core mode codec.

4. INTEGER MDCT

4.1. Decomposition of MDCT

The Integer Modified Discrete Cosine Transform (IntMDCT), introduced in [6], is an invertible integer approximation of the Modified Discrete Cosine Transform (MDCT), which is obtained by utilizing the “Lifting Scheme” [13] or “Ladder Network” [14].

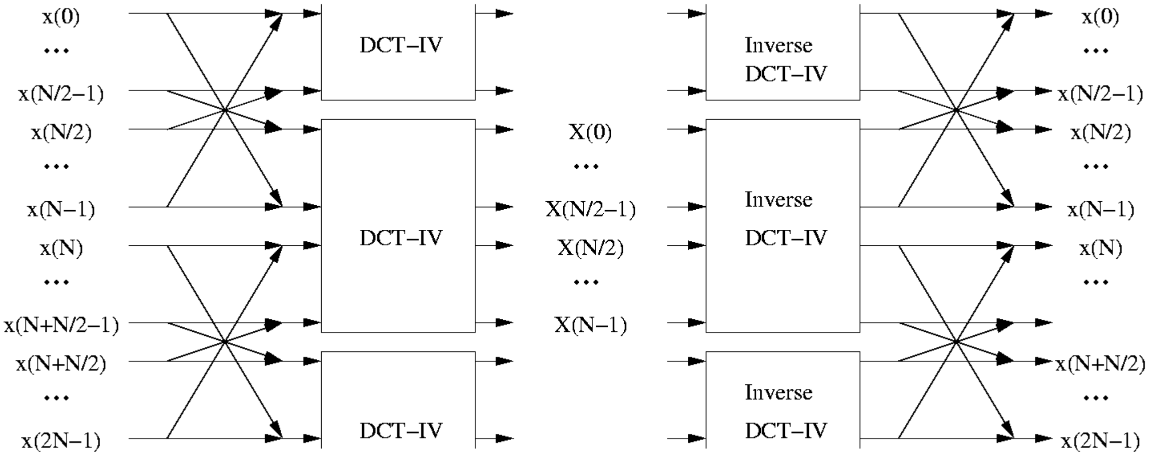

Figure 3: MDCT and inverse MDCT by Windowing/TDA and DCT-IV

The MDCT, defined by

$$X(m)=\sqrt{\frac{2}{N}}\sum_{k=0}^{2N-1} w(k)x(k)\cos\frac{(2k+1+N)(2m+1)\pi}{4N} \quad ,$$

$m=0,...,N-1$

is decomposed into the two blocks:

- Windowing and Time Domain Aliasing (TDA)
- Discrete Cosine Transform of Type IV (DCT-IV)

This is illustrated in Figure 3 for the MDCT and the inverse MDCT.

In the forward IntMDCT the Windowing/TDA block is calculated by 3N/2 so-called lifting steps:

$$\begin{pmatrix} x(k)\\ x(N-1-k)\end{pmatrix}\mapsto$$

$$\begin{pmatrix}1 & -\frac{w(N-1-k)-1}{w(k)}\\ 0 & 1\end{pmatrix}\begin{pmatrix}1 & 0\\ -w(k) & 1\end{pmatrix}\begin{pmatrix}1 & -\frac{w(N-1-k)-1}{w(k)}\\ 0 & 1\end{pmatrix}\begin{pmatrix}x(k)\\ x(N-1-k)\end{pmatrix}$$

$k=0,...,N/2-1$

After each lifting step, a rounding operation is applied to stay in the integer domain. Every lifting step can be inverted by simply adding the subtracted value, and vice versa.

4.2. Integer DCT-IV

For the IntMDCT, the DCT-IV is calculated in an invertible integer fashion, called the Integer DCT-IV. The Multi-Dimensional Lifting (MDL) Scheme [15][16] is applied in order to reduce the required rounding operations in the invertible integer approximation as much as possible.

The following block matrix decomposition for an invertible matrix *T* and the identity matrix *I* shows the basic principle behind the MDL scheme:

$$\begin{pmatrix}T & 0\\ 0 & T^{-1}\end{pmatrix}=\begin{pmatrix}-I & 0\\ T^{-1} & I\end{pmatrix}\begin{pmatrix}I & -T\\ 0 & I\end{pmatrix}\begin{pmatrix}0 & I\\ I & T^{-1}\end{pmatrix}$$

The three blocks in this decomposition are so-called Multi-Dimensional Lifting Steps. Similar to the conventional lifting steps, they can be transferred to invertible integer mappings by rounding the floating-point values after being processed by T resp. T^{-1}, and they can be inverted by subtracting the values that have been added.

By applying the MDL scheme to the DCT-IV, the following decomposition is obtained:

$$\begin{pmatrix} I & 0 \\ 0 & -I \end{pmatrix}\begin{pmatrix} I & \frac{1}{2}\sqrt{2}DCTIV_N \\ 0 & I \end{pmatrix}\begin{pmatrix} I & 0 \\ -\sqrt{2}DCTIV_N & I \end{pmatrix}$$

$$\begin{pmatrix} I & \frac{1}{2}\sqrt{2}DCTIV_N \\ 0 & I \end{pmatrix}\begin{pmatrix} I & -\frac{1}{2}I \\ 0 & I \end{pmatrix}\begin{pmatrix} I & 0 \\ I & I \end{pmatrix}$$

$$= \begin{pmatrix} \frac{1}{2}\sqrt{2}DCTIV_N & \frac{1}{2}\sqrt{2}DCTIV_N \\ \frac{1}{2}\sqrt{2}DCTIV_N & -\frac{1}{2}\sqrt{2}DCTIV_N \end{pmatrix}$$

$$= \begin{pmatrix} \frac{1}{2}\sqrt{2}*I & \frac{1}{2}\sqrt{2}*I \\ \frac{1}{2}\sqrt{2}*I & -\frac{1}{2}\sqrt{2}*I \end{pmatrix}\begin{pmatrix} DCTIV_N & 0 \\ 0 & DCTIV_N \end{pmatrix}$$

In case of stereo signals this decomposition is used to obtain an integrated calculation of the M/S matrix and the Integer DCT-IV for the left and the right channel. The number of required rounding operations is 3N per channel pair, i.e. 3N/2 per channel, which is the same number as for the Windowing/TDA stage. Overall, the Stereo IntMDCT including M/S requires only 3 rounding operations per sample. In case of mono signals the same structure can be utilized. It only has to be extended by some additional lifting steps to obtain the Integer DCT-IV of one block, see [16]. For this Mono IntMDCT 4 rounding operations per sample are required.

4.3. Noise Shaping

The lossless coding efficiency of the IntMDCT is further improved by utilizing a noise shaping technique, introduced in [17]. In the lifting steps where time-domain signals are processed, the rounding operations are connected to an error feedback to provide a spectral shaping of the approximation noise.

This approximation noise affects the lossless coding efficiency mainly in the high frequency region where audio signals usually contain a very small amount of energy, especially at sampling rates of 96 kHz and higher. Hence, a low-pass characteristic of the approximation noise improves the lossless coding efficiency. A first-order noise shaping filter is used, as illustrated in Figure 4.

For the IntMDCT this filter is applied in the three stages of lifting steps in the Windowing/TDA processing and in the first rounding stage of the Integer DCT-IV processing. Figure 5 compares the resulting approximation error between the IntMDCT values and the MDCT values rounded to integer, where the IntMDCT operates both with and without noise shaping.

5. ERROR MAPPING

The error mapping process tries to make the information contained in the AAC core bit-stream and that in the LLE one highly complementary to minimize the overhead of embedding the AAC core. To this end, we notice that the AAC core encoder can be taken as a quantization and coding process [2], where the IntMDCT spectral data are first grouped into different scale factor bands (sfb), which are then quantized with different quantization step size and coded to produce the AAC bit-stream. Assuming that an IntMDCT coefficient $c[k]$ from a particular sfb s is quantized at the AAC core encoder, which generates an output $i[k]$, the following property holds:

$$|thr(i[k])| \le |c[k]| < |thr(i[k])| + \Delta(i[k]),\ k \in s,$$

where $thr(i[k])$ and $\Delta(i[k])$ are, respectively, the quantization threshold closer to zero and the quantization step size for $i[k]$ [2]. That is,

$$thr(i[k]) = \begin{cases} \operatorname{sgn}(i[k])\left[\sqrt[4]{2^{scale\_factor(s)}}\left(|i[k]| - C\right)^{4/3}\right] & ,i[k] \ne 0 \\ 0 & ,i[k] = 0 \end{cases}$$

and

$$\Delta(i[k]) = thr(|i[k]| + 1) - thr(|i[k]|).$$

Here, $scale\_factor[s]$ is the scale factor that determines the quantization step size for sfb s, and the rounding offset is given by $C = 0.4054$.

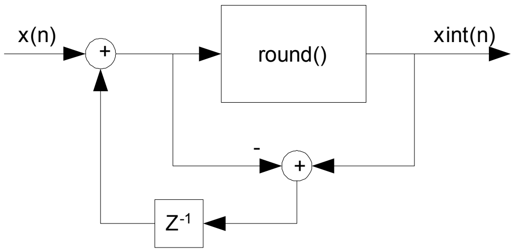

Figure 4. Noise shaping filter for IntMDCT

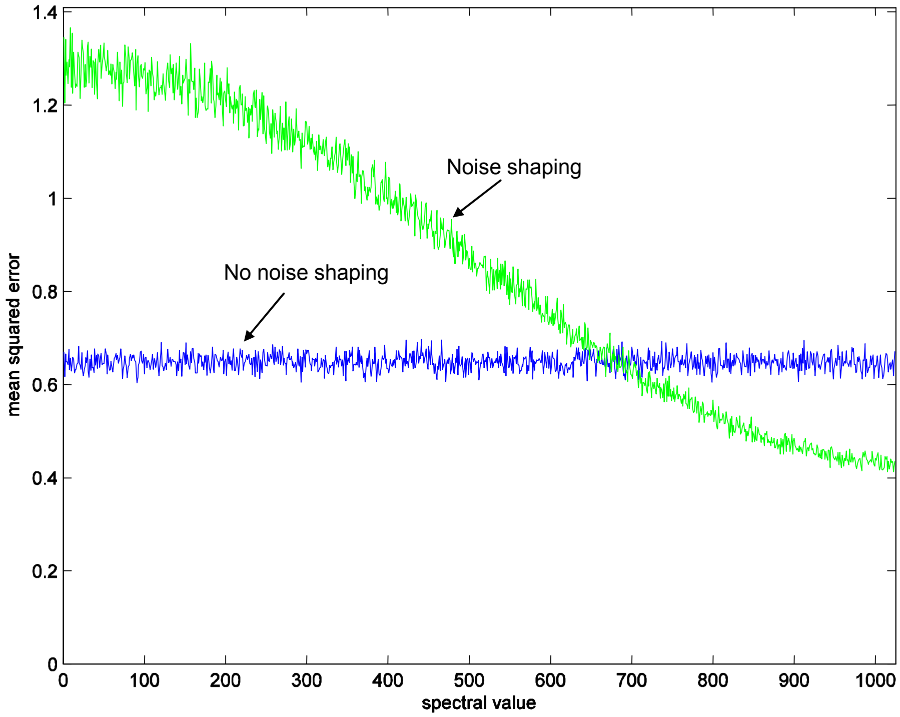

Figure 5. Mean squared approximation error of Stereo IntMDCT (including M/S) with and without noise shaping

Intuitively, the error mapping process should produce a residual signal $e[k]$ with the smallest entropy, which in most cases is simply the Minimum Mean Square Error (MMSE). This is obtained by subtracting the IntMDCT coefficient $c[k]$ to its MMSE reconstruction, i.e., $e[k]=c[k]-\hat{c}[k]$, where the MMSE reconstruction is given by $\hat{c}[k]=E\{c[k]\,|\,i[k]\}$, where $E\{\bullet\}$ denotes the expectation operation. However, such an error mapping process will generally produce uniformly distributed errors $e[k]$ that are virtually incompressible. In order to fully utilize the statistical properties of $c[k]$, the MPEG-4 SLS adopts a somewhat anti-intuitive approach where the residual signal $e[k]$ is given by:

$$e[k]=\begin{cases} c[k]-\lfloor thr(i[k])\rfloor & ,i[k]\neq 0 \\ c[k] & ,i[k]=0 \end{cases} \qquad (1)$$

$k=1,...,1023$.

Here $\lfloor\bullet\rfloor:\mathbb{R}\to\mathbb{Z}$ is the flooring operation that rounds a floating-point value to its nearest integer with smaller amplitude. The advantages of the above error mapping process are two-fold. Firstly, if $c[k]$ is significant in the core layer, i.e., $i[k]\neq 0$, the sign of the IntMDCT residual $e[k]$ will be identical to $i[k]$ and hence it does not need to be coded. Secondly, as illustrated in Figure 6 we notice that if $c[k]$ is a Laplacian RV, from the "memoryless" property of a Laplacian pdf, the amplitude of $e[k]$ will be approximately one-side Laplacian (Geometrical) distributed.

6. SCALABLE CODING OF INTMDCT SPECTRAL DATA

6.1. Perceptual Bit-Plane Coding

In MPEG-4 SLS, the bit-plane coding technology is used in coding the IntMDCT spectral residual to produce the scalable to lossless LLE bit-stream. Consider an input data vector $\mathbf{e}=\{e[0],...,e[N-1]\}$,

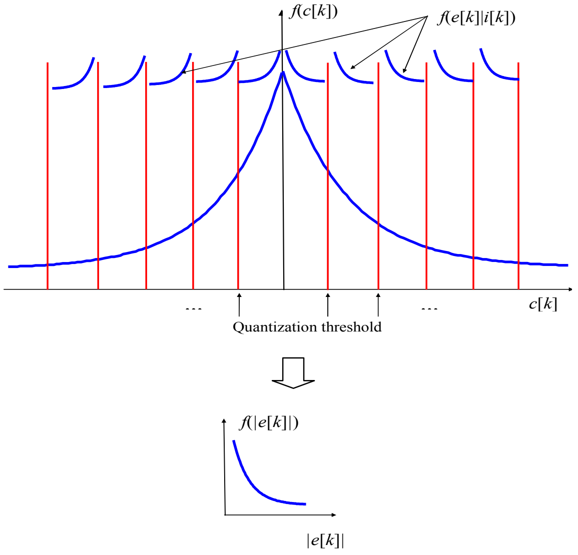

Figure 6. Illustration of the error mapping process. Top: The pdf of the Laplacian distributed IntMDCT coefficient and the conditional pdf of the IntMDCT residual; bottom: The unconditioned pdf of the amplitude of the IntMDCT residual.

for which N is the dimension of $\mathbf{e}$. In a bit-plane coding scheme, each element $e[k]$ in $\mathbf{e}$ is first represented in a binary format as s

$$e[k]=(2s[k]-1)\cdot\sum_{j=0}^{M-1}b[k,j]\cdot 2^{j}, k=0,...,N-1,$$

which comprises of a sign symbol

$$s[k]\overset{\Delta}{=}\begin{cases}1 & ,e[k]\geq 0\\ 0 & ,e[k]<0\end{cases}, \; k=0,...,N-1$$

and bit-plane symbols $b[k,j]\in\{0,1\}$, $i=1,...,k$. Here, M is the Most Significant Bit (MSB) for $\mathbf{e}$ that satisfies $2^{M-1}\leq\max\{|e[k]|\}<2^{M}, k=0,...,N-1$. The bit-planes symbols are then scanned and coded from the MSB to the Least Significant Bit (LSB) for all the elements in $\mathbf{e}$ in a certain order to produce the compressed bit-stream. Clearly, the bit-plane code can be taken as an Entropy Constrained Scalar Quantizer (ECSQ) with successive refined quantization step sizes 2^{T}, $T=M,...,0$; and it is at the same time a lossless entropy coder for an integer source e if the finest quantization step is 1. It can be seen that the bit-plane code provides a very natural approach to generate an FGS to lossless bit-stream; and a lossy bit-stream at any given intermediate rate can be obtained by simply performing a truncation operation on the resulted bit-stream.

It is an essential that the scalability of a scalable audio coding scheme is provided in terms of the perceptual quality. That is, better perceptual quality of the reconstructed audio should be obtained for higher bit-rates of the scalable bit-stream. Generally, this feature is particularly relevant to the cases when the core bit-stream is working at rates that are lower than that for perceptual transparent quality, where the scalability in perceptual quality is easily appreciated by the end users. In the context of MPEG-4 SLS however, this is also relevant to cases when AAC is working at higher rates, for which the scalability is on the coding margin which is important for high-end applications or applications where the reconstructed audio is to be post-processed.

In order to achieve the desirable scalability in perceptual quality, MPEG-4 SLS adopts a rather straightforward perceptual embedding coding principle, which is illustrated in Figure 7. It can be seen that bit-plane coding process is started from the most significant bit-planes (i.e. the first non zero bit-planes) of all the sfb, and progressively moves to lower bit-planes after coding the current for all sfb. Consequently, during this process, the energy of the quantization noise of each sfb is gradually reduced by the same amount. As a result, the spectral shape of the quantization noise, which has been perceptually optimized by the core AAC encoder, is preserved during bit-plane coding process.

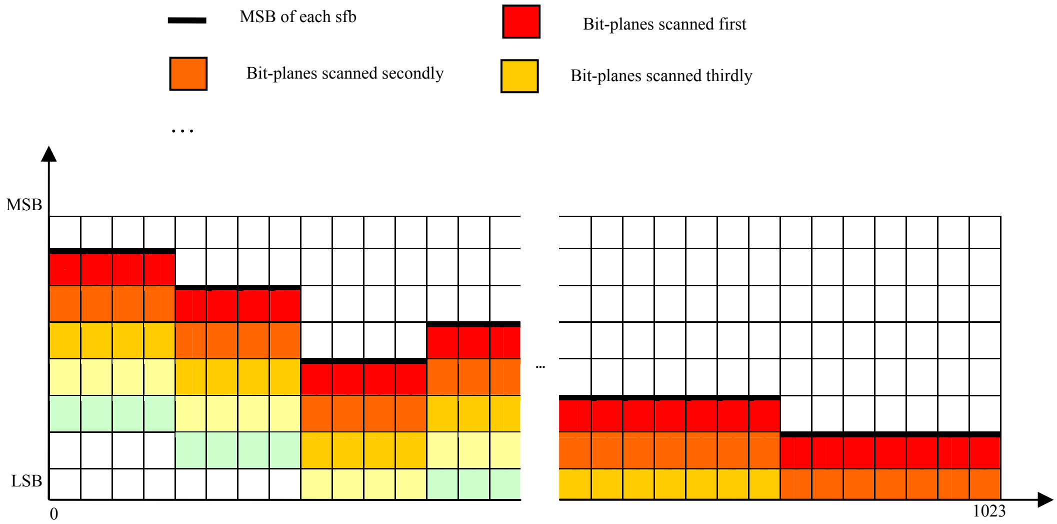

Figure 7. The bit-plane order adopted in MPEG-4 SLS.

As an example, Figure 8 gives the maximum Noise to Mask Ratio (NMR) [21] as a function of audio frames for a piece of symphony audio decoded from SLS bit-streams coded with different intermediate bit-rates at the LLE bit-stream. Evidently, as can be seen from Figure 8, the LLE layer improves the perceptual quality of the AAC core bit-stream as higher bit-rates in the LLE layer always result in smaller MaxNMR over all the audio frames.

6.2. Bit-Plane Coding with BPGC/CBAC

In MPEG-4 SLS, the bit-plane symbols are coded with arithmetic code with fixed frequency tables. There are two different types of frequency tables used in the current SLS RM. The first one is followed the Bit-Plane Golomb Code (BPGC) [18] frequency assignment, for which the frequency assignment rule is derived from the statistical properties of a geometrically distributed source. In BPGC, a bit-plane symbol at bit-plane j is coded with probability assignment $Q(j)$ given by:

$$Q(j)=\begin{cases}\dfrac{1}{1+2^{2^{j-lazy\_bp}}} & ,j\geq lazy\_bp\\ \dfrac{1}{2} & ,j<lazy\_bp\end{cases},$$

where the parameter *lazy_bp* can be selected using the adaptation rule [18] given as follows:

$$lazy\_bp=\min\left\{L'\in\mathbb{Z}\,|\,2^{L'+1}N\geq A\right\}$$

Here N and A are the length and the absolute sum of the data vector to be bit-plane coded respectively. It is shown in [18] that despite its simplicity, BPGC achieves excellent lossless compression ratio performance for a geometrically distributed integer source, and its Rate-Distortion (R-D) performance is very close to that of an optimal ECSQ for that source.

To further improve the coding efficiency, MPEG-4 SLS also introduces a more sophisticated probability assignment, namely, Context-Based Arithmetic Code (CBAC) to complement the BPGC coding method. The idea of CBAC is inspired by the fact that the probability distribution of bit-plane symbols is usually correlated with their frequency location and the significance state of the adjacent spectral lines. In order to capture these correlations, various contexts are employed in CBAC. These include the location of the IntMDCT spectral data and significant states of their adjacent spectral lines, and the bit-plane symbols are then arithmetic coded with frequencies specified in these contexts.

MPEG-4 SLS uses three types of the contexts, namely, the frequency band (FB) context, the distance to lazy (D2L) context, and the significant state (SS) context. The guidance of selecting these contexts is trying to find those contexts that are most “correlated” to the distribution of the bit-plane symbols. In addition, cares

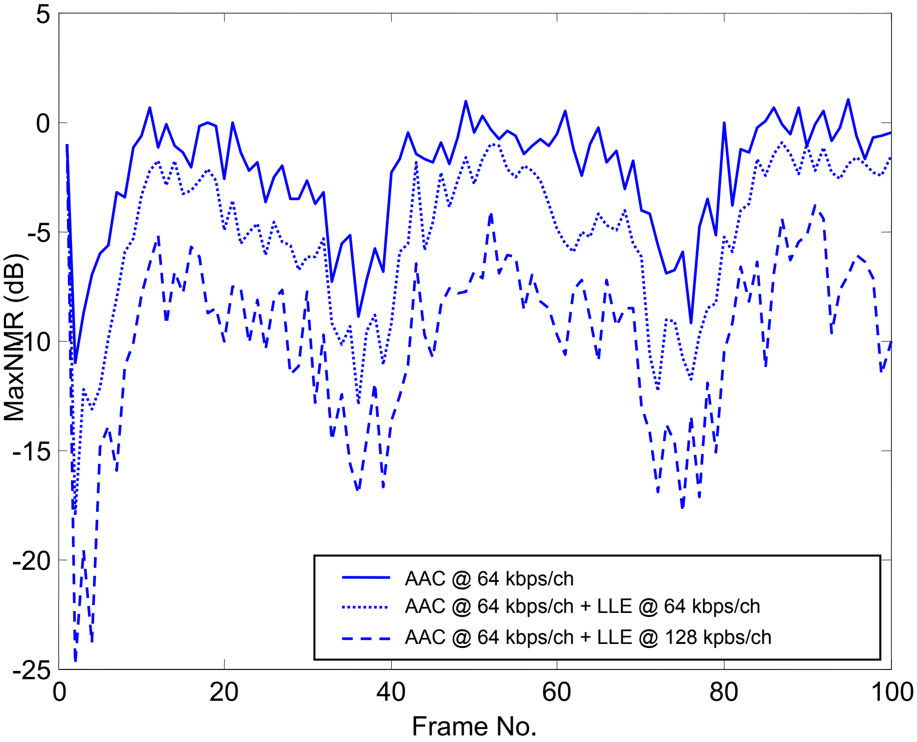

Figure 8. MaxNMR as a function of audio frames (Symphony, 48 kHz/16 bits)

have also taken to avoid "overpopulation" of the number of the contexts, which may deteriorate the coding efficiency performance, and introduce unnecessary implementation burdens. The detailed context assignments are given as follows:

- Context 1: *frequency band (FB)*

It is found in our experiments that the probability distribution of bit-plane symbols varies for different frequency bands. Therefore, we classify the residual spectrum into three different FB contexts, namely, Low Band (0 ~ 4 kHz, *FB* = 0), Mid Band (4 kHz ~ 11 kHz, *FB* = 1) and High Band (above 11 kHz, *FB* = 2).

- Context 2: *distance to lazy (D2L)*

The D2L context is defined as the distance of the current bit-plane to the BPGC *lazy_bp* parameter, which is defined as follows:

$$D2L = \begin{cases} 3 - j + lazy\_bp & , j - lazy\_bp \geq -2 \\ 6 & , else \end{cases}$$

The rationale behind this context is based on the fact that the skew of the probability distribution of the bit-plane symbols from a source with near-Laplacian distribution tends to decrease as the number of D2L decrease. To reduce the total number of this context, all $DL2 \geq 6$ are grouped into one context and coded with probability 0.5 since the probability skew in these contexts is so small that further arithmetic coding of them would only lead to negligible coding gain.

- Context 3: *significant state (SS)*

The SS context is designed to capture the correlation among the amplitude of the current IntMDCT spectral data, those of the adjacent IntMDCT spectral lines and the quantization interval of the AAC core quantizer. To elaborate, we define a vector $sig\_cx(k, j)$ to represent the significance states for adjacent IntMDCT spectral line of bit-plane symbol $b[k, j]$, $k = 0,..., N-1$:

$$sig\_cx(k,j) = \{sig(k-2,j), sig(k-1,j), sig(k+1,j), sig(k+2,j)\}$$

where the significance state $sig(k, j)$ is defined as,

$$sig(k,j)=\begin{cases}0 & \hat{c}_j[k]=0\\ 1 & \hat{c}_j[k]\neq 0\end{cases},$$

and $sig(k,j)$ is taken as 0 if *k* is outside the IntMDCT spectrum. Here $\hat{c}_j[k]$ is the partial reconstruction for $c[k]$ up to bit-plane *j*. That is:

$$\hat{c}_j[k]=\sum_{i=j+1}^{M-1} b[k,i]\cdot 2^i + thr\left(i[k]\right).$$

Note that this definition gives us 2^4 = 16 possible contexts, however, by taking the symmetric property of $sig\_cx(k,j)$, e.g., context $sig\_cx(k,j)=\{1,1,0,0\}$ and context $sig\_cx(k,j)=\{0,0,1,1\}$ can be combined as one context, the actual number of contexts can be reduced to 10.

The context $sig\_cx(k,j)$ is only used in coding bit-plane symbols from insignificant IntMDCT spectral lines. For those from significant ones, we further introduce the *sig_core* context whose value is determined by whether these coefficients are from a sfb that has been quantized and coded at the core AAC encoder (significant sfb) or not (insignificant sfb). The value of *sig_core* context is given by:

$$sig\_core(k)=\begin{cases}0 & c[k]\,\textit{is from an insignificant sfb}\\ 1 & c[k]\,\textit{is from a significant sfb}\end{cases}$$

Furthermore, for the latter case ($sig\_core(k)=1$), from Eq. (1) we further have:

$$0\le e[i]\le \Delta\left(i[k]\right). \qquad (2)$$

As a result, the probability distribution of the bit-plane symbols will be affected by the size of the $\Delta\left(i[k]\right)$ and the following context is employed to capture this dependency:

$$quant\_int(k,j)=\begin{cases}0 & \hat{e}_j[k]+2^{j+1}\le\Delta\left(i[k]\right)\\ 1 & \hat{e}_j[k]+2^{j}\le\Delta\left(i[k]\right)<\hat{e}_j[k]+2^{j+1}\\ 2 & \Delta\left(i[k]\right)<\hat{e}_j[k]+2^{j}\end{cases}$$

where $\hat{e}_j[k]$ is the partial reconstruction of $e[k]$ up to bit-plane $j+1$. Note that if $quant\_int(k,j)=2$, from Eq. (2) it follows that the bit-plane symbol will be 0 with probability 1 hence it does not need to be coded (null context).

In the MPEG-4 SLS decoder, the BPGC or CBAC bit-plane coding process described above is reversed to reconstruct the bit-plane of **e** ; and the lossless reconstruction of **e** is obtained if a full LLE bit-stream is received so that all the bit-stream symbols of **e** are decoded. In the case when the LLE bit-stream is truncated, the MMSE reconstruction $\hat{\mathbf{e}}=\left\{\hat{e}[0],...,\hat{e}[N-1]\right\}$ is given by [18]:

$$\hat{e}[k]=\begin{cases}\left(2s[k]-1\right)\left[\sum_{j=T}^{M-1}b[k,j]2^j+f_k\left(L,T\right)\right] & , \quad \sum_{j=T}^{M-1}b[k,j]2^j\neq 0\\ 0 & , \quad else\end{cases}$$

$$k=0,...,N-1$$

where the fill element $f_k\left(L,T\right)$ is given by the following iteration:

$$f\left(L,T\right)=\sum_{j=0}^{T-1}Q\left(j\right)2^j ,$$

which is implemented in MPEG-4 SLS by using a table lookup.

6.3. Low Energy Mode Coding

The BPGC/CBAC bit-plane coding process used in the current RM for MPEG-4 SLS provides optimal compression performance only for sources with Laplacian or near-Laplacian distributions. Indeed, it is found that for most scale factor band, the IntMDCT spectral data are closely approximated by the Laplacian distribution [26]. However, there exist some “silence” T/F regions, such as those scale factor bands in the high frequencies or those for silence portions of the coded audio, at which the IntMDCT spectral data are in fact dominated by the rounding errors accumulated during the IntMDCT algorithm. The distribution of these IntMDCT spectral data is far away from the Laplacian distribution. In order to improve the coding efficiency MPEG-4 SLS uses a different coding method, namely, the low energy mode coding, for IntMDCT spectral data from sfb with extremely low energy.

The low energy mode coding is used in sfb’s for which the BPGC parameters *lazy_bp* are smaller or equal to 0. At low energy mode coding, the amplitude of residual

Amplitude of $e[k]$	Binary string { $b[pos]$ }
0	0
1	1 0
2	1 1 0
3	1 1 1 0
4	1 1 1 1 0
…	…
$2^M - 2$	1 1 … … … 1 0
$2^M - 1$	1 1 … … … 1 1
pos	0 1 2 3 …

Table 1. Binarization of IntMDCT error spectrum at low energy mode

spectral data $e[k]$ is first converted into a unitary binary string $\mathbf{b} = \{b[0], b[1], \ldots, b[pos], \ldots\}$ as in Table 1

Clearly, the probability distributions of the binary symbols in **b** are jointly determined by its position *pos*, and the distribution of $e[k]$ which is in turn related to the *lazy_bp* parameter:

$$\Pr\{b[pos]=1\} = \Pr\{e[k] > pos \mid e[k] \geq pos\}$$
$$0 \leq pos < 2^M$$

Therefore, $b[pos]$ is arithmetic coded conditioned on its position *pos* in **b** and the *lazy_bp* parameter. For simplicity, fixed frequency assignments are used here which are trained from long audio training sequences for different *pos* and *lazy_bp*.

7. PERFORMANCE

The lossless compression performance of MPEG-4 SLS is evaluated by using the audio testing sets from the MPEG lossless audio coding task group [4]. These testing sets comprise audio sequences of popular sampling rates and word lengths combinations, namely, 48/16 (kHz/bit per sample), 48/24, 96/24 and 192/24, and for each combination audio sequences of a variety of music styles such as vocal, instrumental, jazz and classical music are included. Some of these items are recordings of the New York Symphonic Ensemble. In our tests, the core AAC encoder is working at 48 kHz sampling rate by adjusting the oversampling factor (osf). That is, osf = 1 for 48 kHz testing sets, osf = 2 for the 96 kHz testing set, and osf = 4 for the 192 kHz testing sets; and it operates at a bit-rate for "(nearly) perceptually transparent quality", that is, 64 kbits/s per channel. The MPEG-4 SLS encoder non-core mode is also included in our tests, where the osf is set to 4 for best performance. For comparison, Monkey's Audio 3.97 (MA397) [22], a popular open source lossless only audio codec that delivers the state-of-the-art compression performance [23], is used as the benchmarks in our tests, where it is set to its highest compression setting. The comparison results are listed in Table 2. It can be seen that for most items the MPEG-4 SLS non-core mode encoder achieves similar or better compression ratio performances compared to those of MA397. In addition, these results also show that despite the abundant functionalities of MPEG-4 SLS if the AAC core encoder is presented, the overhead for embedding such a core in terms of compression ratio is quite trivial, which is smaller than 5% in the worst case (48/16 testing set) compared to the non-core mode encoders.

Items	MAC397	MPEG-4 SLS (-osf 1)	MPEG-4 SLS (Non-core)
avemaria	2.68	2.55	2.69
blackandtan	1.85	1.76	1.83
broadway	2.11	1.96	2.05
cherokee	1.92	1.83	1.90
clarinet	2.19	2.08	2.17
cymbal	3.36	3.03	3.45
dcymbals	1.69	1.60	1.66
etude	2.48	2.38	2.49
flute	2.60	2.46	2.63
fouronsix	2.22	2.09	2.19
haffner	1.87	1.79	1.86
mfv	3.54	3.27	3.55
unfo	2.01	1.91	2.00
violin	2.15	2.04	2.14
waltz	1.94	1.84	1.92
Overall	2.21	2.09	2.20
w.r.t. MAC397	100%	94.57%	99.54%

a) Compression ratio for 48 kHz/16 bit testing set

Items	MAC397	MPEG-4 SLS (-osf 1)	MPEG-4 SLS (Non-core)
avemaria	1.73	1.69	1.73
blackandtan	1.43	1.40	1.44
broadway	1.54	1.49	1.52
cherokee	1.47	1.44	1.47
clarinet	1.57	1.53	1.57
cymbal	2.07	2.01	2.09
dcymbals	1.36	1.33	1.36
etude	1.66	1.64	1.67
flute	1.70	1.66	1.71
fouronsix	1.58	1.54	1.58
haffner	1.44	1.42	1.45
mfv	1.93	1.88	1.94
unfo	1.50	1.47	1.51

violin	1.55	1.52	1.55
waltz	1.47	1.44	1.47
Overall	1.58	1.54	1.58
w.r.t. MAC397	100.00%	97.47%	100%

b) Compression ratio for 48 kHz/24 bit testing set

Items	MAC397	MPEG-4 SLS (-osf 2)	MPEG-4 SLS (Non-core)
avemaria	2.00	1.98	2.01
blackandtan	2.09	2.18	2.22
broadway	1.77	1.73	1.76
cherokee	2.14	2.21	2.25
clarinet	2.32	2.34	2.42
cymbal	2.21	2.17	2.22
dcymbals	1.68	1.65	1.68
etude	1.94	1.92	1.95
flute	2.36	2.35	2.41
fouronsix	2.38	2.41	2.46
haffner	2.03	2.05	2.10
mfv	2.03	2.00	2.03
unfo	2.20	2.28	2.33
violin	2.18	2.19	2.24
waltz	2.16	2.24	2.28
Overall	2.08	2.09	2.13
w.r.t. MAC397	100%	100.48%	102.40%

c) Compression ratio for 96 kHz/24 bit testing set

Items	MAC397	MPEG-4 SLS (-osf 4)	MPEG-4 SLS (non-core)
avemaria	2.79	2.85	2.87
broadway	2.50	2.49	2.52
cymbal	2.18	2.18	2.20
dcymbals	2.46	2.54	2.58
etude	2.71	2.78	2.80
mfv	2.82	2.89	2.92
Overall	2.56	2.60	2.62
w.r.t. MAC397	100%	101.56%	102.34%

d) Compression ratio for 192 kHz/24 bit testing set

Table 2. Lossless compression performance of MPEG-4 SLS RM4.

8. CONCLUSION

This paper presents the design and the structure for MPEG-4 Audio Scalable to Lossless (SLS) coding - the latest MPEG work item in audio. MPEG-4 SLS provides the support for the full spectrum of digital audio applications ranging from high compression low bit-rates applications to those that require lossless quality in a single framework. This is achieved by adopting the scalable coding approach in the Integer MDCT domain, and integrating a perceptually fine-granular scalable layer on top of the core layer MPEG-4 AAC, the state-of-the-art perceptual audio coder. Experimental results show that despite its abundant functionalities, MPEG-4 SLS still maintains excellent lossless compression performance. In addition, MPEG-4 SLS also introduces a "light-weight" non-core mode for applications that require lossless coding only; and its lossless compression performance rivals other best lossless only audio codecs available nowadays.

9. REFERENCES

[1] ISO/IEC JTC1/SC29/WG11, "Information technology – Coding of moving pictures and associated audio for digital storage media at up to about 1.5 Mbit/s – Part3: Audio," IS 11172-3, 1992.

[2] ISO/IEC JTC1/SC29/WG11, "Coding of Audiovisual Objects, Part 3. Audio, Subpart 4 Time/Frequency Coding," International Standard 14496-3, 1999.

[3] T. Painter and A. Spanias, "Perceptual Coding of Digital Audio", Proceeding of the IEEE, vol.88, no. 2, pp. 451 – 513, April 2000.

[4] ISO/IEC JTC1/SC29/WG11 (MPEG), "Call for Proposals on MPEG-4 Lossless Audio Coding," N5040, Shanghai, China, Oct. 2002.

[5] R. Yu, X. Lin, S. Rahardja and H. Huang, "Technical description of I2R's proposal for MPEG-4 audio scalable lossless coding (SLS): advanced audio zip (AAZ)," ISO/IEC JTC1/SC29/WG11, M10035, October 2003, Brisbane, Australia.

[6] R. Geiger, T. Sporer, J. Koller and K. Brandenburg, "Audio Coding based on Integer Transforms," 111th AES Convention preprint 5471, New York, USA, Sep. 2001.

[7] R. Geiger, "Information on Improved Integer MDCT", ISO/IEC JTC1/SC29/WG11, M9873, July 2003, Trondheim, Norway.

[8] R. Yu, X. Lin, S. Rahardja, and H. Haibin, "Proposed Core Experiment for improving coding efficiency in MPEG-4 audio scalable coding (SLS)", ISO/IEC JTC1/SC29/WG11, M10136, Oct. 2003, Brisbane, Australia.

[9] R. Yu, X. Lin, S. Rahardja, and H. Haibin, "Proposed Core Experiment for improving coding efficiency in MPEG-4 audio scalable coding (SLS)", ISO/IEC JTC1/SC29/WG11, M10683, March. 2004, Munich, Germany.

[10] R. Geiger, M. Schmidt, J. Herre, "Proposed Core Experiment on MPEG-4 SLS", ISO/IEC JTC1/SC29/WG11, M10711, March. 2004, Munich, Germany.

[11] R. Yu, X. Lin, S. Rahardja, C. C. Ko, "A scalable lossy to lossless audio coder for MPEG-4 lossless audio coding," *Proc. ICASSP*, 2004

[12] M. Bosi, K. Brandenburg, S. Quackenbush, L. Fielder, K. Akagiri, H. Fuchs, and M. Dietz, "ISO/IEC MPEG-2 advanced audio coding,", J. Audio Eng. Soc., pp. 789 – 813, Oct. 1997.

[13] I. Daubechies and W. Sweldens, "Factoring Wavelet Transforms into Lifting Steps," Tech. Rep., Bell Laboratories, Lucent Technologies, 1996.

[14] F. Bruekers and A. Enden, "New networks for perfect inversion and perfect reconstruction," IEEE JSAC, vol. 10, no. 1, pp. 130–137, Jan. 1992.

[15] R. Geiger, Y. Yokotani, G. Schuller, "Improved Integer Transforms for Lossless Audio Coding", Asilomar Conf. on Signals, Systems, and Computers, Pacific Grove, California, USA, November 9 - 12, 2003

[16] R. Geiger, Y. Yokotani, G. Schuller, J. Herre, "Improved Integer Transforms using Multi-Dimensional Lifting", International Conference on Acoustics, Speech, and Signal Processing (ICASSP), May 17-21, 2004, Montreal, Quebec, Canada

[17] Y. Yokotani, R. Geiger, G. Schuller, S. Oraintara, K. R. Rao, "Improved Lossless Audio Coding using the Noise-Shaped IntMDCT", IEEE 11th DSP Workshop, August 1-4, 2004, Taos Ski Valley, New Mexico, USA.

[18] R. Yu, C.C. Ko, S. Rahardja and X. Lin, "Bit-plane Golomb code for sources with Laplacian distributions," Proc. of the ICASSP 2003, pp. 277 – 280, 2003.

[19] M. Davis, "The AC-3 multichannel coder," in Proc. 95th Conv. Aud. Eng. Soc., Oct. 1993, preprint 3774.

[20] N. S. Jayant and P. Noll, Digital Coding of Waveforms: Principles and Application to Speech and Video, Prentice Hall, 1990.

[21] B. Beaton and et al, "Objective perceptual measurement of audio quality," in Collected papers on digital audio bit-rate reduction, pp. 126 – 152, AES, 1996.

[22] Monkey's Audio, open source lossless audio codec (http://www.monkeysaudio.com)

[23] Comparison of lossless audio coders, http://flac.sourceforge.net/comparison.html

[24] Ralf Geiger, Jürgen Herre, Jürgen Koller, Karlheinz Brandenburg, "IntMDCT – A link between perceptual and lossless audio coding", International Conference on Acoustics Speech and Signal Processing (ICASSP), May 13-17, 2002, Orlando, Florida

[25] Ralf Geiger, Gerald Schuller, Jürgen Herre, Ralph Sperschneider and Thomas Sporer, "Scalable Perceptual and Lossless Audio Coding based on MPEG-4 AAC", 115th Convention of Audio Engineering Society (AES), October 10-13, 2003, New York, USA

[26] R. Yu, X. Lin, S. Rahardia, and C. C. Ko, "A Statistics Study of the MDCT Coefficient Distribution for Audio," presented at International Conference on Multimedia and Expo, Taipei, China, 2004