A fine granular scalable perceptually lossy
and lossless audio codec

R. Yu, X. Lin, S. Rahardja and C. C. Ko

1Institute for Infocomm Research, A*STAR,
2Department of Electrical and Computer Engineering
National University of Singapore, Singapore 119260, email: elekocc@nus.edu.sg

ABSTRACT
This paper presents Advanced Audio Zip (AAZ), an audio codec that provides the fine granular bit-rate scalability from lossy to lossless coding. Perceptually embedded coding principle is employed in AAZ to provide lossy reconstruction with optimal perceptual quality at intermediate bit-rates. AAZ also provides the backward compatibility where the lossless bit-stream embeds a compliant MPEG-4 AAC bit-stream.

1. INTRODUCTION
During the last decade the development of perceptual audio coding [1] has successfully provided the tools for delivering audio contents of perceptually transparent quality with limited storage or bandwidth resources. However, with the progress of technology it is now becoming realistic to deliver audio contents with high sampling rates, high amplitude resolution (e.g. 96 kHz, 24 bit/sample) with lossless quality. Therefore, it would be very desirable to have audio codecs that provide the fine-granular scalability (FGS) between lossy and lossless coding as a bridge for such an evolution. In addition, this FGS property is also very useful for transmission applications as it provides a hierarchical bit-stream that can be dynamically changed during the transmission to achieve soft degradation of QoS.

In this paper we present Advance Audio Zip (AAZ), a scalable perceptual and lossless audio codec. As will be shown in this paper, AAZ gives an ideal solution for lossless audio compression with the following characteristics:

• Perceptually embedded lossless bit-stream – the lossless bit-stream can be randomly truncated to meet almost any rate/fidelity constraints with granularity down to a single byte. Meanwhile the perceptual optimality is preserved in the truncated lossless bit-stream.
• Low complexity – AAZ consumes only very limited CPU and memory requirement in both its encoder and decoder implementations.

The lossless bit-stream of AAZ can be utilized in several manners to meet the various rate/quality requirements for different application scenarios. For example, an AAC bit-stream can be parsed from the AAZ lossless bit-stream, which can be decoded by a conventional MPEG-4 AAC decoder to generate a perceptually optimized reproduction of the original audio signal. On the other hand, a lossless reconstruction is obtained if both the base layer and the lossless enhance layer bit-streams are decoded. In other application scenarios, the AAZ provides the flexibility to truncate its bit-stream to lower rates at the encoder/decoder or in the communication channel for any rate/fidelity/complexity constraints that may be arisen in practical systems.

2. STRUCTURE
Fig. 1 illustrates the structure of the AAZ codec, which comprises of two distinguished layers, namely, a perceptual core layer, in particular an MPEG-4 AAC audio coder [2], and a Lossless Enhancement (LLE) layer.

In the encoder, the input audio frames are transformed losslessly by using the integer MDCT (IntMDCT) [3] to generate the IntMDCT coefficients \( c(k) \), where \( k = 1, \ldots, 1024 \). In order to generate the core layer bit-stream, \( c(k) \) are scaled to approximate the output of an MDCT filterbank used in the MPEG-AAC encoder. This scaled IntMDCT coefficients are then quantized and coded with perceptually weighted so that the noise introduced is best masked by the masking threshold of the human auditory system, in the core AAC encoder. The resulted core layer bit-stream thus constitutes the minimum rate/quality unit of the final lossless bit-stream.

For optimal coding efficiency, an error-mapping procedure is employed in the LLE layer to remove the information that has been coded in the core layer from the IntMDCT coefficients. This error mapping procedure can be described as follows:

\[ e(k) = c(k) - \text{thr}(k). \]

where \( \text{thr}(k) \) is the lower (closer to zero) quantization threshold for \( c(k) \) in the AAC core layer quantizer. Clearly, for \( c(k) \) that has already been significant in the core layer (\( \text{thr}(k) \neq 0 \)), the sign of \( e(k) \) is determined...
and hence only its amplitude is coded. In addition, it is well known that for most audio signals, $c(k)$ can be approximated by Laplacian random variables with the probability density function (pdf):

$$ f(c(k)) = e^{k(c(k))} \frac{1}{\sqrt{2\sigma^2}}. $$

From the “memoryless” property of a Laplacian pdf we have,

$$ f(k) = \frac{1}{\sqrt{2\sigma^2}} e^{-|k|/\sigma}. $$

where the distribution parameter $\sigma$ is determined by the variance of $c(k)$ and the quantization step size. If the core layer quantizer is uniform, the amplitude of $c(k)$ is independent to the base layer quantization threshold $thr(k)$, and Geometrically distributed. In the decoder, the IntMDCT coefficients can be reconstructed as:

$$ c(k) = e(k) + thr(k). $$

### 3. Perceptually embedded coding of IntMDCT residual using BPGC

In order to achieve the bit-rate scalability, the IntMDCT residual signal $e(k)$ is then bit-plane coded using Bit-Plane Golomb Code (BPGC) [4] to generate the embedded LLE bit-stream. First, $e(k)$ is converted into sign and bit-plane symbols as:

$$ e(k) = \begin{cases} k \in \{0, 1\} & \text{if the input data, respectively.} \\
0 & \text{else for (L = -1;(N >> (-L )) >= A;L --);} \\
1 & \text{else for (L = 0;(N << (L + 1)) < A;L ++);} 
\end{cases} $$

$\frac{1}{\sqrt{2\sigma^2}} e^{-|k|/\sigma}.$

where $\theta$ is determined by the variance of $c(k)$ and the quantization step size. If the core layer quantizer is uniform, the amplitude of $c(k)$ is independent to the base layer quantization threshold $thr(k)$, and Geometrically distributed. In the decoder, the IntMDCT coefficients can be reconstructed as:

$$ c(k) = e(k) + thr(k). $$

### 3. Perceptually embedded coding of IntMDCT residual using BPGC

In order to achieve the bit-rate scalability, the IntMDCT residual signal $e(k)$ is then bit-plane coded using Bit-Plane Golomb Code (BPGC) [4] to generate the embedded LLE bit-stream. First, $e(k)$ is converted into sign and bit-plane symbols as:

$$ e(k) = \begin{cases} k \in \{0, 1\} & \text{if the input data, respectively.} \\
0 & \text{else for (L = -1;(N >> (-L )) >= A;L --);} \\
1 & \text{else for (L = 0;(N << (L + 1)) < A;L ++);} 
\end{cases} $$

$\frac{1}{\sqrt{2\sigma^2}} e^{-|k|/\sigma}.$

where $\theta$ is determined by the variance of $c(k)$ and the quantization step size. If the core layer quantizer is uniform, the amplitude of $c(k)$ is independent to the base layer quantization threshold $thr(k)$, and Geometrically distributed. In the decoder, the IntMDCT coefficients can be reconstructed as:

$$ c(k) = e(k) + thr(k). $$

### Importance of the code family $C$ lies in the fact that there exists a one-to-one relationship in terms of compression performance between code family $C$ of $L > 0$ and the Golomb code [5], which is the optimal prefix code for Geometric distributed integers [6]. It can be shown that if a positive integer number is losslessly coded using $G^L$ (hereby the bit-plane scanning is performed from $\infty$ to 0) with $L > 0$, the resulted code length would be identical to that of the Golomb code of order $m = 2^L$ [4].

Clearly, the code family $C$ partitions the support of the Geometric distribution parameter $\theta$ into disjoint regions and each region has a correspondent optimal BPGC from $C$. Here the optimality is measured in terms of the expected code length of BPGC. This partition rule is given in [5] as follows

$$ \phi^{2^{-1}} \leq \theta < \phi^{2^{-1}}. $$

where $\phi = \frac{1}{\sqrt{5}} - 1$ (inverse of the golden ratio). A simplified decision rule is also given in the following C-program,

```c
if (N <= A) for (L = 0;((N << (-L )) >= A;L ++));
else for (L = -1;((N >> (-L )) >= A;L --));
```

where $N$ and $A$ are the length and the absolute sum of the input data, respectively.

In AAZ, for each scalar-factor bank an optimal $G^L$ is assigned to code the IntMDCT residual signal $e(k)$. The optimal $L$ is also transmitted as side information for correct reconstruction at the decoder. For $e(k)$ that has not been significant in the base layer coder its sign symbol $s(k)$ is also coded after first non-zero bit-plane symbol $b(k,j)$ is coded.
In order to preserve the spectral shape of the core layer noise for best perceptual quality, AAZ adopt a perceptually embedded coding procedure, which is illustrated in Fig. 2. It can be seen that BPGC starts from the largest bit-planes for all the scalar-factor bands, and progressively moves to lower bit-planes after finishing coding the current bit-planes for all the scalar-factor bands. Consequently, during this process the energy of the quantization noise of each scalar-factor band is gradually reduced with the same amount. As a result, only the level of the quantization noise is reduced and its spectral shape is preserved. This technique is particularly useful to maintain the perceptual optimality of the reconstructed audio signal when the LLE layer bit-stream is truncated to lower bit-rates.

The AAZ lossless audio codec also provides a non-core mode by simply disabling the AAC lossy core encoder, which is useful for applications where the embedded lossy AAC bit-stream is not required. This mode significantly reduces the complexity of AAZ, and provides a slightly better overall compression performance. However, if the resultant bit-stream is truncated, the reconstructed audio signal is optimal in terms of Mean Square Error (MSE) instead of perceptual quality.

4. PERFORMANCE

The compression performance of AAZ is tested using audio sequences (48 kHz, 16 bits/sample) that have been used in the evaluation process for MPEG-4 lossless audio [7]. The core layer codec is an MPEG-4 AAC codec. The bit-rates for the core encoder are 64 kbps/ch. For comparison, the latest version of Monkey’s audio (ver. 3.97) [8], a state of the art lossless audio codec, is used as the benchmark codec. The comparison results are given in Table 1. Clearly, despite the ample scalabilities and functionalities provided by AAZ, it achieves an average compression performance only 4% inferior to the state-of-art non-scalable lossless audio coding scheme. In addition, better lossless compression performance can be achieved by disabling the AAC core if it is not needed.

In order to evaluate the lossy performance of AAZ, we further compared the Noise to Masking Ratio (NMR) [9] of the reconstructed audio signal where the LLE layer bit-stream is truncated to lower bit-rates. Here instead of comparing the average NMR performance, we measure the maximum NMR (MaxNMR) from all scalar-factor banks for each frame, which reflects the most perceptible distortion for a given time instance. Fig. 3 gives the MaxNMR as a function of time for the first 100 audio frames of testing item Violin. Evidently, the LLE layer improves the perceptual quality of the core AAC layer, with higher bit-rates in the LLE layer bit-stream always resulting in smaller MaxNMR for all the audio frames.

5. CONCLUSION

In this paper we propose a scalable perceptual and lossless audio codec - Advanced Audio Zip (AAZ). AAZ provides excellent lossless compression performance, while retaining full bit-rate scalability. In addition, backward compatibility is provided in that an MPEG-4 AAC compliant bit-stream is embedded in its lossless bit-stream. With these characteristics, it is envisaged that AAZ has widespread applications in the current and future audio storage and streaming applications.

REFERENCE
[7] ISO/IEC JTC 1/SC 29/WG 11N5208, Final Call for Proposals on MPEG-4 Lossless Audio Coding, Shanghai

<table>
<thead>
<tr>
<th>Item</th>
<th>Monkey Audio</th>
<th>AAZ</th>
<th>AAZ (no-core)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avermario</td>
<td>0.37</td>
<td>0.41</td>
<td>0.40</td>
</tr>
<tr>
<td>Clarinet</td>
<td>0.46</td>
<td>0.49</td>
<td>0.48</td>
</tr>
<tr>
<td>Cymbal</td>
<td>0.30</td>
<td>0.38</td>
<td>0.35</td>
</tr>
<tr>
<td>Etude</td>
<td>0.40</td>
<td>0.44</td>
<td>0.42</td>
</tr>
<tr>
<td>Flute</td>
<td>0.38</td>
<td>0.42</td>
<td>0.41</td>
</tr>
<tr>
<td>Haffner</td>
<td>0.53</td>
<td>0.57</td>
<td>0.55</td>
</tr>
<tr>
<td>Violin</td>
<td>0.47</td>
<td>0.50</td>
<td>0.48</td>
</tr>
<tr>
<td>Overall</td>
<td>0.42</td>
<td>0.46</td>
<td>0.44</td>
</tr>
<tr>
<td>Difference to Monkey</td>
<td>0.04</td>
<td>0.02</td>
<td></td>
</tr>
</tbody>
</table>

Table 1 Lossless performance of AAZ
Fig. 2 Perceptually embedded coding in AAZ

Fig. 3 MaxNMR as a function of audio frames (Symphony, 48 kHz, 16 bits/sample)