ENHANCED SCALABLE TO LOSSLESS AUDIO CODING SCHEME

Haiyan Shu, Haibin Huang, Te Li and Susanto Rahardja

Signal Processing Department
Institute for Infocomm Research
Agency for Science, Technology & Research
Singapore 138632

ABSTRACT
Scalable to lossless (SLS) audio coding is a state-of-art audio coding technique that has been adopted as MPEG scalable audio coding tool. To realize bit-plane refinement, this technique employs bit-plane arithmetic coding for lossless entropy coding, and Laplacian distribution is used to model the input data to realize high compression efficiency. In this paper, bit-plane probability is analyzed when generalized Gaussian distribution is used to model the input data. Based on the result of bit-plane probability for generalized Gaussian distribution, a low cost bit-plane arithmetic coding method is presented. This scheme is implemented in the SLS audio coding platform. With the same computational complexity, the proposed algorithm presents higher compression efficiency than SLS.

Index Terms— Scalable to lossless audio coding, arithmetic coding, bit-plane probability, generalized Gaussian distribution.

1. INTRODUCTION
Lossless coding of data is a very challenging problem for many applications. Additionally, frequent delays in transmissions require progressive coding in order to provide the end users with useful successive refinements of the final data. Embedded progressive coding refers to the encoding of data into a bitstream that can be parsed efficiently to obtain lower rate and lower quality descriptions of the data. It enables the granularity quality improvement for different environment constraints and applications.

Bit-plane arithmetic coding is an efficient solution to reach embedded progressive coding. It encodes the data into multiple bit-plane layers. Each bit-plane may refine the final quality with different contributions. When all the bit-planes are correctly received, lossless quality is achieved. These bit-planes in the generated bitstream can also be arbitrarily truncated when certain constraint is applied. This fulfills the requirements of embedded coding. Bit-plane arithmetic coding is widely used for data compression [1, 2].

In bit-plane arithmetic coding process, it is necessary to get the actual bit-plane probability (BPP) for efficient compression. In [3], scalable to lossless (SLS) audio coding is presented. This technique utilizes the bit-plane arithmetic coding scheme to realize the scalability of the bitstream. An bit-plane probability estimation method, named bit-plane Golomb code (BPGC), was introduced in SLS. Based on this technology, SLS improves the coding efficiency of scalable audio significantly. It is for this reason, this audio coding scheme was adopted as MPEG-4 scalable audio coding tool in 2006 [4]. In BPGC, the distribution of the input data is assumed to be Laplacian distribution. It presents a simple but efficient estimation.

However, not all data follows Laplacian distribution. When the input data is not Laplacian distributed, the coding efficiency will be degraded. In [5], approximated bit-plane probability for generalized Gaussian distribution has been proposed. This approximation gives an analytic solution to estimate the bit-plane probability for generalized Gaussian distribution. However, when applying for real application, it still requires additional overhead bits. In this paper, we propose to apply this approximated solution for scalable to lossless audio coding on the SLS platform. The proposed solution can further improve the coding efficiency of SLS without additional overhead information transmission. In addition, the proposed scheme can keep the computational complexity as low as SLS.

This paper is organized as follows. The scalable to lossless audio coding and BPGC is first reviewed in Section 2. Based on the result from the approximated BPP of generalized Gaussian distribution, enhanced bit-plane arithmetic coding for SLS is proposed in Section 3. Experimental results are presented in Section 4, and the conclusion is drawn in Section 5.

2. SCALABLE TO LOSSLESS AUDIO CODING
MPEG-4 SLS audio coding [4] was released as a standard audio coding tool in June 2006. It allows the fully scaling up to lossless with a wide range of intermediate bitrate representa-
In SLS, the input integer PCM data is first transformed into the frequency domain using integer modified discrete cosine transform (intMDCT) [6]. The output intMDCT coefficients are then passed to the AAC encoder to generate the core layer AAC bitstream, and the coding residue is then coded using BPCC to generate the scalable enhancement layer bitstream.

In this process, bit-plane arithmetic coding is adopted to code each bit-plane symbol with its corresponding probability based on the calculation from BPCC. In the following, we first review the BPCC estimation scheme in BPCC.

Considering an intMDCT coefficient $x$, it can be represented by a set of binary symbols $b_j \in \{0, 1\}$ in the following binary format

$$x = (2s - 1) \sum_{j=0}^{M} b_j \cdot 2^j$$  \hspace{1cm} (1)

where $s$ is the sign symbol and $M$ is the maximum bit-plane number.

Since the probability of sign symbol $P_s(s=0) = P_s(s=1) = 0.5$, only the non-negative part of $x$ is discussed in the following. In [3], BPCC assumes that the input intMDCT coefficients follow zero mean Laplacian distribution:

$$f(x; \sigma) = \sqrt{\frac{1}{2\sigma^2}} e^{-\frac{x}{\sqrt{2\sigma^2}}}$$  \hspace{1cm} (2)

where $\sigma$ is the standard deviation of $x$. Then the bit-plane symbol probability of $b_j$ taking value 1 is given by

$$P_j = \begin{cases} \frac{1}{1 + 2^{2^{L-1} - j}}, & j \geq L - \frac{1}{2} \\ \frac{1}{2}, & j < L - \frac{1}{2} \end{cases}$$  \hspace{1cm} (3)

Here, $L$ is the code parameter. It is used to index the family of BPCC. The bit-plane symbols are then coded with an arithmetic coder using the probability assignment given by (3). For the sign symbols, they are simply coded with probability assignment $1/2$.

In [3], the code parameter $L$ is defined as

$$L = \min \left \{ L' \in \mathbb{Z} | 2^{L'+1} \geq \sum_{i} |x| / N \right \}$$  \hspace{1cm} (4)

where $\sum |x|$ is the summation of the input data, $N$ is the number of input data, and $\mathbb{Z}$ is the set of integer.

### 3. ENHANCED SLS AUDIO CODING WITH GENERALIZED GAUSSIAN ASSUMPTION

The bit-plane Golomb coding has been successfully implemented in SLS. It assumes that the input data are Laplacian distributed. However, when the input data are not Laplacian distributed, bit-plane probability calculated by BPCC may not be optimum. Generalized Gaussian distribution is a most commonly used description, and has been widely observed in many practical applications. This inspires us to investigate the coding efficiency when generalized Gaussian distribution is assumed for the input data to further improve the coding efficiency of SLS.

In [5], an approximated analytic solution for the bit-plane probability of generalized Gaussian distribution was proposed. In this section, we propose a low cost scheme that utilizes the result from [5] to enhance the performance of bit-plane arithmetic coding for MPEG-4 SLS.

#### 3.1. Bit-plane probability of generalized Gaussian distribution

Assuming the intMDCT coefficient $x$ has a zero mean generalized Gaussian distribution, the probability density function of $y = |x|$ is given by

$$g(y; \sigma, \gamma) = \frac{\gamma \cdot \eta(\gamma, \sigma)}{\Gamma(\gamma)} \cdot e^{-\frac{\gamma \cdot \eta(\gamma, \sigma)}{\sigma^2}} y^{\gamma - 1}, \quad y \geq 0.$$  \hspace{1cm} (5)

where $\sigma > 0$ is the standard deviation, $\gamma > 0$ is the shape parameter, and

$$\eta(\gamma, \sigma) = \sqrt{\frac{2}{\pi}} \frac{\Gamma(\frac{\gamma + 1}{2})}{\Gamma(\frac{\gamma}{2}) \Gamma(\frac{1}{2})}$$  \hspace{1cm} (6)

$$\Gamma(z) = \int_0^\infty t^{z-1} e^{-t} dt.$$  \hspace{1cm} (7)

In [5], the approximated solution of bit-plane probability $P_j$ for generalized Gaussian distribution was presented as

$$P_j(\sigma, \gamma) = \frac{1}{1 + e^{\sqrt{\gamma \eta(\gamma, \sigma) \cdot 2^{j - L}}}}.$$  \hspace{1cm} (8)

This is an analytic description. It is verified in [5] that this solution presents bit-plane probability very close to the actual bit-plane probability.

#### 3.2. Practical bit-plane probability representation

Referring to (8), we can estimate the BPP of the input signal with generalized Gaussian distribution. However, to get an accurate estimation, we need to calculate the parameter $\sigma$ which requires high computation. In addition, many bits is required to transmit this non-integer parameter. To reduce the overhead information to be transmitted and lower the computational complexity, a practical solution is designed to improve the coding efficiency without introducing additional computation.

For generalized Gaussian distribution, we define the code parameter $L$ as

$$L = \left \lceil \log_2 \frac{1}{\eta(\gamma, \sigma)} \right \rceil.$$  \hspace{1cm} (9)
and $\alpha$ as

$$\alpha = \log_2 \frac{1}{\eta(\gamma, \sigma)} - \log_2 \frac{1}{\eta(\gamma, \sigma)} = \log_2 \frac{1}{\eta(\gamma, \sigma)}. \quad (10)$$

where the value of $\alpha$ varies between 0 and 1 (i.e. $\alpha \in [0, 1]$). So, we have following relationship:

$$\eta(\gamma, \sigma) = 2^{-L-\alpha}. \quad (11)$$

Then the BPP of generalized Gaussian distribution presented in (8) can be written as

$$P_j(\sigma, \gamma) = \frac{1}{1 + e^{\gamma(\frac{1}{2} - \frac{1}{\sigma})\sqrt{2}}}. \quad (12)$$

According to the probability density function of $y$ defined in (5), the expectation of $y$ is derived as

$$E\{y\} = \int_0^\infty y \cdot g(y; \sigma, \gamma) \, dy = \int_0^\infty \frac{y \cdot \eta(\gamma, \sigma)}{\Gamma(\frac{1}{2})} \cdot e^{-\frac{1}{\eta(\gamma, \sigma)} \cdot y^2} \, dy$$

$$= \frac{\gamma \cdot \eta(\gamma, \sigma)}{\Gamma(\frac{1}{2})} \cdot \int_0^\infty \frac{\omega^{\frac{1}{\sigma}} \cdot e^{-\omega}}{\eta^2(\gamma, \sigma) \cdot \gamma} \cdot \omega^{\frac{1}{\sigma}-1} \, d\omega$$

$$= \frac{\gamma^{-1}(\gamma, \sigma)}{\Gamma(\frac{1}{\sigma})} \cdot \frac{\Gamma(\frac{1}{2})}{\Gamma(\frac{1}{\gamma})}. \quad (13)$$

According to 11, the following result is derived:

$$L = \min \left\{ L' \in \mathbb{Z} | 2^{L'+1} \geq E\{y\} \cdot \frac{\Gamma(\frac{1}{2})}{\Gamma(\frac{1}{\gamma})} \right\}. \quad (14)$$

In this equation, $E\{y\}$ is expectation of the input $y$. $\sum^{\infty}_{n=0} N$ is its unbiased estimator, where $N$ is the number of samples. In practical calculation, $\sum^{\infty}_{n=0} N$ is used to replace $E\{y\}$. Code parameter $L$ can be determined with little computational cost.

However, additional computation will be introduced in the estimation of code parameter $L$ when comparing with SLS. To solve this problem, we rewrite (12) as

$$P_j(\sigma, \gamma) = \frac{1}{1 + e^{\gamma(\frac{1}{2} - \frac{1}{\sigma})\sqrt{2}} \cdot 2^{-L-\alpha}}. \quad (15)$$

In such a way, the code parameter $L$ in (15) is defined the same way as that in (4).

### Table 1. Comparison of compression ratio of different SLS codecs in non-core mode.

<table>
<thead>
<tr>
<th>Lossless coders</th>
<th>SLS-BPGC</th>
<th>Proposed scheme</th>
<th>improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set 1 (15 sequences)</td>
<td>2.2308</td>
<td>2.2342</td>
<td>0.15%</td>
</tr>
<tr>
<td>Set 2 (15 sequences)</td>
<td>1.5804</td>
<td>1.5821</td>
<td>0.1%</td>
</tr>
<tr>
<td>Set 3 (15 sequences)</td>
<td>2.1277</td>
<td>2.1297</td>
<td>0.1%</td>
</tr>
<tr>
<td>Set 4 (7 sequences)</td>
<td>2.5511</td>
<td>2.5539</td>
<td>0.11%</td>
</tr>
</tbody>
</table>

### 3.3. Analysis of complexity

In MPEG-4 SLS, the bit-plane probability is pre-calculated. Only parameter $L$ is calculated online and transmitted to the decoder. The proposed solution calculates the parameter $L$ in the same way as SLS. In addition, the BPP table is pre-calculated by (15) and stored the same way as MPEG-4 SLS. This process requires no additional storage and introduces no additional computation.

### 4. EXPERIMENTAL RESULTS

In this section, the performance of the proposed enhanced SLS audio coding scheme will be evaluated.

First, we examine the performance of BPP for generalized Gaussian distribution in the format given in (15). The probability calculated from (15) is compared with the actual BPP value based on some parameter sets $(\gamma, \sigma)$. In this simulation, two different shape parameters $(\gamma)$ are chosen. Their values are 0.4 and 1.3, respectively. For each $\gamma$, three different standard deviations $(\sigma = 20, 80, 640)$ are selected. The comparison results of two (approximated and actual) BPP values based on the different parameter sets $(\gamma, \sigma)$ are illustrated in Fig. 1. It can be observed that, the approximated values are quite close to the actual one, especially when the BPP value falls into $[0.05, 0.4]$. This shows that the format given in (15) can be used as an approximation of the actual BPP value of generalized Gaussian distribution.

Next, we apply our proposed scheme in the SLS platform to verify its performance. In the experiment, the MPEG test sequences are coded. The test sequences consist of 15 different types of music with different sampling frequency and resolution. According to the sampling frequency and resolution, they are grouped into four sets: 48kHz/24bit, 48kHz/16bit, 96kHz/24bit, and 192kHz/24bit, respectively. Each waveform lasts 30 seconds. The performance of the proposed scheme is compared with the SLS Bit-Plane Golomb Coding mode (SLS-BPGC) [4].

To be compatible with the SLS coding platform, the parameter $\gamma$ is fixed for the proposed algorithm. There is no need to estimate the value of $\gamma$. Table 1 summarizes the experimental results for the tested coders in the non-core mode where $\gamma = 1.04$ for the proposed algorithm, and Table 2 shows the results when there is an AAC core coder running at 128kbps and $\gamma = 1.34$ for the proposed algorithm. Compared
Fig. 1. Comparison between the actual bit-plane probability and the approximated bit-plane probability based on reference bit-plane ($L$) with different standard deviation ($\sigma$) (a) $\gamma = 0.4$; (b) $\gamma = 1.3$.

Table 2. Comparison of different SLS codecs in 128kbps AAC core mode.

<table>
<thead>
<tr>
<th>Lossless coders</th>
<th>SLS-BPGC</th>
<th>Proposed scheme</th>
<th>improvement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set 1 (15 sequences)</td>
<td>2.1291</td>
<td>2.1333</td>
<td>0.20%</td>
</tr>
<tr>
<td>Set 2 (15 sequences)</td>
<td>1.5498</td>
<td>1.5514</td>
<td>0.1%</td>
</tr>
<tr>
<td>Set 3 (15 sequences)</td>
<td>2.1004</td>
<td>2.1025</td>
<td>0.1%</td>
</tr>
<tr>
<td>Set 4 (7 sequences)</td>
<td>2.5268</td>
<td>2.5305</td>
<td>0.14%</td>
</tr>
</tbody>
</table>

6. REFERENCES


