ESCAPED-HUFFMAN AND ADAPTIVE RECURSIVE RICE CODING FOR LOSSLESS COMPRESSION OF THE MAPPED DOMAIN LINEAR PREDICTION RESIDUAL

Noboru Harada, Yutaka Kamamoto, Takehiro Moriya

1NTT Communication Science Laboratories, Nippon Telegraph and Telephone Corporation, Japan

ABSTRACT

ITU-T Recommendation G.711.0 has just been established. It defines a lossless and stateless compression for G.711 packet payloads (for both A-law and μ-law). This paper introduces some coding technologies proposed and applied to the G.711.0 codec, such as Plus-Minus zero mapping for the mapped domain linear predictive coding and escaped-Huffman coding combined with adaptive recursive Rice coding for lossless compression of the prediction residual. Performance test results for those coding tools are shown in comparison with the results for the conventional technology. The performance is measured based on the figure of merit (FoM), which is a function of the trade-off between compression performance and computational complexity. The proposed tools improve the compression performance by 0.16% in total while keeping the computational complexity of encoder/decoder pair low (about 1.0 WMOPS in average and 1.667 WMOPS in the worst-case).


1. INTRODUCTION

The ITU Rec. G.711 [1] is widely used for narrowband telephony applications, including PSTN/GSTN and packet-based network applications such as VoIP, and has been used for many decades because of its proven voice quality, ubiquity, and utility. ITU has just established a lossless coding technology for G.711 encoded payloads. This new standard is ITU-T Rec. G.711.0 [2].

The G.711.0 codec may be used as a traditional codec and its use negotiated (end-to-end) by the end terminals (IP phones, conference bridge endpoints, etc.). Additionally, owing to its lossless and stateless design, G.711.0 may also be used as a lossless compression mechanism on any intermediate link (e.g., service provider VoIP backbone links at voice gateways) where G.711 is used by the end systems. G.711.0 employed in these transcoding applications provides bandwidth savings without any degradation of audio quality relative to G.711 since it is a lossless algorithm. For these gateway applications, low computational complexity is desired. The figure of merit (FoM), defined in the G.711.0 Terms of Reference (ToR) [3], was used to assess the tradeoff between complexity and signal compression during the design phase and in the G.711.0 selection process [4].

G.711.0 is a lossless compression algorithm that operates on 40, 80, 160, 240, and 320 samples per 8-kHz sampled G.711 input frame. The bit rate is variable and the size of the (compressed) output frame depends on the input signal characteristics. The minimum size of an encoded frame is one byte. The maximum size of an encoded frame is the input frame size plus one byte.

Following coding tools are included in G.711.0: An uncompressed coding tool, constant coding tools (Constant Plus零 zero coding, Constant Minus zero coding, Constant non-zero coding), a Plus-Minus (PM) zero Rice coding tool, a binary coding tool, a pulse mode coding tool, a value-location coding tool, a fractional-bit coding tool, a min-max level coding tool, a direct linear predictive coding tool, and a mapped domain linear predictive (MDLP) coding tool.

The MDLP coding is a kind of LP coding but especially designed for G.711 A-law and μ-law input (A similar scheme had been also proposed by F. Ghido, et. al. [5]).

This paper introduces some new coding schemes proposed and applied to the G.711.0 codec, especially related to the MDLP coding tool. PM zero mapping is used to calculate the prediction residual and Escaped-Huffman (E-Huffman) coding combined with adaptive recursive Rice coding is used as an entropy coding scheme for the prediction residual. Test results are also examined in terms of the compression performance/computational complexity trade-off based on the FoM.

Section 2 overviews the mapped domain linear prediction. Sections 3 and 4 introduce PM zero mapping and E-Huffman coding combined with adaptive recursive Rice coding, respectively. Section 5 shows the test results. Finally, Section 6 concludes the paper.

2. MAPPED DOMAIN LINEAR PREDICTION

Figure 1 shows a block diagram of the MDLP encoding tool used in the G.711.0 encoder. The MDLP coding tool takes a sequence of N G.711 A-law, \( y(n) \), or G.711 μ-law, \( y(n) \), symbols. First, these N G.711 symbols are converted into \( x_{\text{pcm}}(n) \), \( 0 \leq n < N \) in the uniform (linear) PCM domain and a short-term prediction is carried out in that domain using LP analysis. The prediction residual signal, however, is obtained in the range of \([-255,255]\) since the predicted value is subtracted from the target value \( x_{\text{pcm}}(n) \) in the 8-bit logarithmic domain (denoted as the \( \text{int8} \) domain in this paper). PARCOR coefficients are used to represent and signal the LPC parameter.

Linear prediction is applied as follows:

\[
\hat{x}_{\text{int8}}(n) = f_{\text{PCM}} \rightarrow \text{int8} \left( \sum_{i=1}^{P} a_i \cdot f_{\text{int8}} \rightarrow \text{PCM} \left( x_{\text{int8}}(n-i) \right) \right) \tag{1}
\]

where \( a_i \) is the \( i \)-th LPC coefficient of \( P \)-th order prediction, \( x_{\text{int8}}(n-i) \) and \( \hat{x}_{\text{int8}}(n) \) are the previous sample value and the predicted sample value in the \( \text{int8} \) domain, and \( f_{\text{PCM}} \rightarrow \text{int8} \) and
Input signal (A-law/μ-law) → Frame buffer → Prefix code for Frame length and coding tool → Encoding tool selection → Mapped domain LP coding → Output G.711.0 encoded frame

**Tools proposed in this paper**

Fig. 1. Block diagram of the mapped domain linear prediction tool in the G.711.0 encoder.

\[ f_{\text{int8-PCM}} \] are the mapping function from uniform PCM to int8 and the inverse mapping function. Prediction residual is calculated in the int8 domain as

\[ r(n) = x_{\text{int8}}(n) - \hat{x}_{\text{int8}}(n), \quad 0 \leq n < N \quad (2) \]

### 3. PM ZERO MAPPING AND RESIDUAL CALCULATION

Because of the definition of the μ-law [1], there are two zeros (plus zero 0+ and minus zero 0- ) in the minimum quantization interval of the μ-law signal. In the MDLP coding, the value 0- cannot be predicted because the value can’t be represented in the uniform (linear) PCM domain. However, in order not to lose the value 0-, the value 0- has to be kept as the value -1 in the int8 domain (See the mapped values for \( f_{\text{int8}}(n) = 0x7F \) in Table 1).

Here, we have proposed the PM zero mapping for the residual calculation in μ-law case, as

\[ r(n) = f_{\text{int8-PCM}}(x_{\text{int8}}(n)) - f_{\text{int8-PCM}}(\hat{x}_{\text{int8}}(n)) \quad (3) \]

where \( f_{\text{int8-PCM}} \) is one of the mapping functions shown in Table 1, which maps the values for the int8 domain depending on the observed existence of 0- and 0+ samples.

First, the numbers of samples of which the value is 0+ and the value is 0- are counted in the input frame. Then, depending on the observation, one of the non-linear mappings shown in Table 1 is applied to the target value \( x_{\text{int8}}(n) \) and to the predicted value \( \hat{x}_{\text{int8}}(n) \) before the residual calculation. For instance, if neither 0- nor 0+ is observed in the frame, all input sample values \( x_{\text{int8}}(n) \) and predicted sample values \( \hat{x}_{\text{int8}}(n) \) of +1, 0 and -1 are mapped to 0, and other negative values are increased by 1 before the residual calculation of Eq. (3). The selected mapping (existence of 0- and 0+) for the frame is signaled by the corresponding codes shown in Table 1.

With this mapping, the magnitude of the prediction residual can be reduced by 1 (or 2) when one of (or both of) the zeros is (are) not found in the input frame and the signs of the target value and the predicted value are different. The code is sent only for μ-law input.

### 4. PREDICTION RESIDUAL CODING

In the G.711.0, E-Huffman coding combined with adaptive recursive Rice coding is newly proposed as an entropy coding scheme for the prediction residual signal \( r(n) \), \( 0 \leq n < N \), and applied to frames larger than 40 samples. For 40-sample frames, Rice coding is used. First, the residual \( r(n) \) is decomposed into a quotient \( k(n) \) and a remainder \( j(k) \) based on a separation parameter \( S \), as described later in Sections 4.1 and 4.2. The separation parameter \( S \) (Rice parameter for Rice coding) is calculated as

\[ S = \left[ \log_2 \left( 2 \ln(2) \cdot \bar{F} \right) + \lambda \right] \quad (4) \]

where \( \bar{F} \) is averaged absolute amplitude of \( r(n) \) calculated in the range \( \min(2, P) \leq n < N \), \( \lambda = 0.5 \) for Rice coding (for 40-sample frames) and \( \lambda = 0.3 \) for E-Huffman coding. The value of \( S \) is Huffman encoded in 1- to 6-bits and transmitted to the decoder. When sub-frame separation is applied, a difference of \( S \) from the previous sub-frame is encoded for the second and latter sub-frames.

---

**Table 1. PM zero mapping functions for μ-law values.**

<table>
<thead>
<tr>
<th>( f_{\mu}(n) )</th>
<th>( x_{\text{int8}}(n) )</th>
<th>( 0^+ ) and 0- exist</th>
<th>( 0^+ ) or 0- exist</th>
<th>( 0^- ) exist</th>
<th>( 0^- ) exists</th>
</tr>
</thead>
<tbody>
<tr>
<td>0x080</td>
<td>+127</td>
<td>+127</td>
<td>+126</td>
<td>+127</td>
<td>+127</td>
</tr>
<tr>
<td>0x081</td>
<td>+126</td>
<td>+126</td>
<td>+125</td>
<td>+126</td>
<td>+126</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>0x0FF</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0x7F</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>0x01</td>
<td>-127</td>
<td>-127</td>
<td>-126</td>
<td>-127</td>
<td>-127</td>
</tr>
<tr>
<td>0x00</td>
<td>-128</td>
<td>-128</td>
<td>-127</td>
<td>-127</td>
<td>-127</td>
</tr>
<tr>
<td>Code</td>
<td>0</td>
<td>100</td>
<td>101</td>
<td>110</td>
<td></td>
</tr>
</tbody>
</table>

---

4647
4.1. Rice coding (Conventional coding scheme)

Rice code (also known as Golomb-Rice code) of the residual value \( r(n) \) is calculated as follows when a Rice parameter \( S \) is given:

If \( S = 0 \), after \( k(n) \) Os, one 1 is presented.

\[
k(n) = \begin{cases} 2r(n) & \text{if } r(n) \geq 0 \\ -2r(n) - 1 & \text{if } r(n) < 0 \\ \end{cases} \quad \ldots (5).
\]

For cases \( S > 0 \), after \( k(n) \) Os, one 1 appears. Then remainder \( j(n) \) follows in \( S \)-bit representation.

\[
k(n) = \begin{cases} 2^{-(S-1)}r(n) & \text{if } r(n) \geq 0 \\ 2^{-(S-1)}[-r(n) - 1] & \text{if } r(n) < 0 \\ \end{cases} \quad \ldots (6).
\]

\[
j(n) = \begin{cases} r(n) \& 2^{S-1} - 1 & 2^{S-1} - 1 & \text{if } r(n) \geq 0 \\ -(r(n) - 1) \& 2^{S-1} - 1 & \text{if } r(n) < 0 \\ \end{cases} \quad \ldots (7)
\]

where \( \& \) denotes an AND bit-operator.

4.2. E-Huffman coding with adaptive recursive Rice coding

The distribution of the residual signal \( r(n) \), however, sometimes does not follow the expected model of Rice coding. For cases \( S = 0 \), Rice coding with \( 0 \) is equivalent to Rice code when \( 0 \) \& \( r(n) \) is calculated as follows when a Rice parameter \( S \) is given.

\[
S = \begin{cases} 0 & \text{if } r(n) \geq 0 \\ 0 & \text{if } r(n) < 0 \\ \end{cases} \quad \ldots (8).
\]

Table 2 shows the E-Huffman code table entries. The best one of the four E-Huffman tables is selected for every frame on the basis of the estimated encoded bitstream size of the frame. The selected E-Huffman table for the frame is signaled by an index code. Note that the number of entries is limited to \( \text{maxCode} \), which is either 6 or 7. Quotient values \( k(n) < \text{maxCode} \) can be encoded by one of the Huffman codes listed in E-Huffman tables I to IV. For the quotient values \( k(n) \geq \text{maxCode} \), an extra code \( E(k(n) - \text{maxCode}) \) is produced after the escape code, e.g., '00001'. The coding scheme used for the extra code E is adaptively switched depending on the separation parameter \( S \) (Therefore, the coding scheme is switched depending on averaged amplitude and individual values of the residual signal). Unary code is used for the extra coding if \( S = 0 \). Rice coding with a Rice parameter of 1 is used for the extra coding if \( S > 0 \). Note that E-Huffman table IV is equivalent to Rice code when \( S = 0 \) or \( k(n) < \text{maxCode} \).

5. EVALUATION OF THE PROPOSED SCHEMES

The proposed coding algorithms and conventional algorithms were implemented in ANSI-C using software tool library STL2005 v2.2 [6]. The coding performance of the codec in various conditions were evaluated in terms of the computational complexity/compression performance trade-off based on the FoM.

5.1. FoM

The performance of the codec should be measured in terms of the computational complexity/compression performance tradeoff. The trade-off is assessed by using the following FoM:

\[
\text{FoM} = R - w \times (\max(C, C_{\text{obj}})) \quad \ldots (8)
\]

Table 2. Escaped Huffman code tables.

<table>
<thead>
<tr>
<th>( k(n) )</th>
<th>I</th>
<th>II</th>
<th>III</th>
<th>IV</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>01</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>10</td>
<td>01</td>
<td>01</td>
<td>01</td>
</tr>
<tr>
<td>2</td>
<td>11</td>
<td>0001</td>
<td>001</td>
<td>001</td>
</tr>
<tr>
<td>3</td>
<td>001</td>
<td>0010</td>
<td>0001</td>
<td>0001</td>
</tr>
<tr>
<td>4</td>
<td>0001</td>
<td>000001</td>
<td>00010</td>
<td>00001</td>
</tr>
<tr>
<td>5</td>
<td>00001</td>
<td>000000</td>
<td>000000</td>
<td>000000</td>
</tr>
<tr>
<td>6</td>
<td>000000</td>
<td>000001</td>
<td>000001</td>
<td>000001+E</td>
</tr>
<tr>
<td>7</td>
<td>000001+E</td>
<td>0011+E</td>
<td>00011+E</td>
<td>–</td>
</tr>
</tbody>
</table>

Note: E denotes an extra code for \( k(n) - \text{maxCode} \)

Table 3. Test corpus.

| Speech duration | 523 seconds |
| Languages | Mandarin, Chinese, English, American, Japanese, Polish, and American Spanish |
| Speakers | Four sentence pairs spoken by four different speakers (two males and two female) |
| Test categories | (a1): Clean speech with input levels -16, -26, and -36 dBv; voice activity factor (VAF) of 45 % +/- 1 %; both A-/\( \mu \)-law. |
| | (a2): Noisy speech with input level -26 dBv; VAF of 45 % +/- 1 %; both A-/\( \mu \)-law; SNRs of 15, 20 and 25 dB. Noise conditions: cafeteria, street, office noise, interfering talker, background music |
| | (a3): Tandem cases with G.711.1 R1, EFR+DTX, G.729, and G.726 for clean speech/noisy speech conditions [same as above (a1) and (a2)] (for EFR, car noise is added) |
| Total duration | 137 hours |

where \( C \) is averaged computational complexity for encoder/decoder pair measured using basic operators defined in ITU-T Recommendation averaged over all input frame sizes and conditions in weighted million operations (WMOPS), and \( R \) is compression ratio defined as

\[
R: \text{Compression ratio [%]} = \left(1 - \frac{\text{compressed size}}{\text{original size}}\right) \times 100 \ldots (9).
\]

The weighting factor for the complexity penalty was set to \( w = 2.0 \) based on the discussion for ToR [3] among the experts in ITU-T. The minimum penalty factor was clipped on the basis of the objective complexity value \( C_{\text{obj}} = 1.0 \).

5.2. Test corpus

The test corpus used for the evaluation test is shown in Table 3. The test corpus is generated from P.501 speech corpus [6]. The duration of an input speech signal was 523 seconds, which includes all languages listed and resulted in over 137 hours of processed data for all categories [4] of test conditions in Table 1. A \( \mu \)-law corpus recorded from an in-service network operated in Japan was provided by NTT (1.4 GB) [8] and was also used.

5.3. Performance of the PM zero mapping tool

Tables 4 shows the compression performance of the conventional MDLP coding scheme (similar to [7]) and the MDLP with the PM
zero mapping scheme proposed for the G711.0 standard. Test corpora described in Section 5.2 were used for the test. It is shown that the proposed PM zero mapping (PMZ) performs better than the conventional scheme (No PMZ) for all frame lengths except 40.

Based on the results in Table 4, PM zero mapping was applied only for μ-law signals in 80-sample frames and larger (marked with "∗" and under lined) in the actual G.711.0 specification. Table 5 shows the complexity of encoder and decoder and the FoM for the G.711.0 coder with and without PM zero mapping for μ-law input signals. The FoM score is increased by 0.19 and the compression ratio is improved by 0.2%, while averaged complexity is increased by 0.06 WMOPS.

5.4. Performance of the adaptive recursive Rice coding and E-Huffman coding tools

Table 6 shows the compression performance, complexity, and FoM for conventional Rice coding (Rice), adaptive recursive Rice coding (RR), E-Huffman without adaptive recursive Rice coding (EH) and E-Huffman combined with adaptive recursive Rice coding (EH+RR). Though the (EH+RR) is the scheme used in the G.711.0, conditions applied only RR and EH were also tested in order to show the individual contribution of EH and RR. The RR was implemented by limiting the (EH+RR) to use only the E-Huffman table IV with no index code transmission. The EH was implemented by limiting the (EH+RR) to use a unary code for the extra coding E for all S values. Note that all additional tools are applied only for the 80-sample frames and larger because the additional index code required for the E-Huffman coding made the FoM score worse for 40-sample frames. All results are averaged over all input categories and all frame lengths for both a-law and μ-law with the test corpora described in Section 5.2.

It is shown that the compression performance of the RR is 0.03% better than that of Rice but that the FoM score became worse because the computational complexity is increased by 0.015 WMOPS. Both the compression performance and FoM score are improved by EH. When EH is combined with adaptive RR, compression performance is further improved because EH reduces the code size for quotient values \(k(n) < \text{maxCode}\) and RR reduces the code size for larger quotient values, \(k(n) \geq \text{maxCode}\). The combined scheme improves both the compression performance and FoM score. Note that the average complexity of EH+RR is increased only 0.001 compare to that of EH because RR can be implemented on top of EH at almost no cost.

6. CONCLUSION

Coding schemes proposed and applied to the G.711.0 were described. PM zero mapping is proposed for the prediction residual calculation and E-Huffman coding combined with adaptive recursive Rice coding is proposed for the prediction residual compression. It is shown that the PM zero mapping improves the compression performance by 0.2% and improves the FoM score by 0.19 for μ-law input. The E-Huffman coding combined with adaptive recursive Rice coding improves the compression by 0.16% while the complexity was increased by 0.046 WMOPS averaged for all test conditions, compared to the conventional Rice coding scheme. The worst-case complexity was increased by 0.079 WMOPS. Therefore, the FoM score was improved by 0.072.

Average computational complexity is 1.071 WMOPS for the encoder/decoder pair and the worst-case complexity is 1.667 WMOPS in total.

<table>
<thead>
<tr>
<th>Frame length</th>
<th>No PMZ</th>
<th>PMZ</th>
</tr>
</thead>
<tbody>
<tr>
<td>μ-law 40</td>
<td>46.10%</td>
<td>45.97%</td>
</tr>
<tr>
<td>80</td>
<td>49.59%</td>
<td>*49.75%</td>
</tr>
<tr>
<td>160</td>
<td>51.96%</td>
<td>*52.24%</td>
</tr>
<tr>
<td>240</td>
<td>52.56%</td>
<td>*52.84%</td>
</tr>
<tr>
<td>320</td>
<td>52.84%</td>
<td>*53.13%</td>
</tr>
</tbody>
</table>

Table 4. Compression performance of the conventional and with PM zero mapping (PMZ) for μ-law signal [%].

<table>
<thead>
<tr>
<th>Input data size [Mbytes]</th>
<th>No PMZ</th>
<th>PMZ N&gt;40</th>
</tr>
</thead>
<tbody>
<tr>
<td>3,354</td>
<td>3,354</td>
<td></td>
</tr>
<tr>
<td>1,657</td>
<td>1,650</td>
<td></td>
</tr>
<tr>
<td>Comp. ratio [%]</td>
<td>50.61%</td>
<td>50.81%</td>
</tr>
<tr>
<td>Ave. WMOPS (total enc.+dec.)</td>
<td>1.125</td>
<td>1.131</td>
</tr>
<tr>
<td>Worst-case WMOPS (enc.+dec.)</td>
<td>1.655</td>
<td>1.667</td>
</tr>
<tr>
<td>Worst-case WMOPS (enc.)</td>
<td>1.067</td>
<td>1.078</td>
</tr>
<tr>
<td>Worst-case WMOPS (dec.)</td>
<td>0.556</td>
<td>0.589</td>
</tr>
<tr>
<td>FoM score</td>
<td>48.3584</td>
<td>48.5493</td>
</tr>
</tbody>
</table>

Table 5. Test results with/without PM zero mapping (PMZ) for μ-law input signals.

<table>
<thead>
<tr>
<th>Rice</th>
<th>RR N&gt;40</th>
<th>EH N&gt;40</th>
<th>EH+RR N&gt;40</th>
</tr>
</thead>
<tbody>
<tr>
<td>Comp. ratio [%]</td>
<td>53.13%</td>
<td>53.16%</td>
<td>53.25%</td>
</tr>
<tr>
<td>Ave. complexity [WMOPS]</td>
<td>1.025</td>
<td>1.040</td>
<td>1.070</td>
</tr>
<tr>
<td>Worst complexity [WMOPS]</td>
<td>1.588</td>
<td>1.616</td>
<td>1.663</td>
</tr>
<tr>
<td>FoM score</td>
<td>51.080</td>
<td>51.079</td>
<td>51.106</td>
</tr>
</tbody>
</table>

Table 6. Performance results for the adaptive recursive Rice coding (RR) and E-Huffman (EH) coding.

7. ACKNOWLEDGEMENTS

The authors thank all G.711.0 contributors, especially Dr. Yusuke Hiwasaki, Mr. Csaba Kós, Mr. Lei Miao, Mr. Lorin Netsch, Mr. Fengyan Qi, Mr. Michael A. Ramalho, Mr. Jacek Stachurski, and Mr. Hervé Taddei (alphabetical order).

8. REFERENCES