Multimode Tree Coding of Speech with Backward Pitch Prediction and Perceptual Pre- and Post-weighting

Ying-Yi Li and Jerry D. Gibson

Department of Electrical and Computer Engineering, University of California, Santa Barbara, USA
Email: yingyi_li@umail.ucsb.edu, gibson@ece.ucsb.edu

Abstract—A low delay and low complexity Multimode Tree Coder with backward pitch predictor is proposed. For the Multimode Tree Coding, the speech is classified into five different modes, and each mode is coded at a suitable bit-rate using a tree coder with perceptual pre- and post-weighting filters. In order to improve the speech quality without increasing the delay, a backward pitch predictor is added into the Voiced mode. The results show that the pitch predictor does improve the PESQ-MOS. In addition, the PESQ-MOS of the Multimode Tree Coder is equivalent to the PESQ-MOS of AMR-NB at 12.2 kbps and G.728 at 16 kbps while the computational complexity is lower than AMR-NB and the delay is lower than AMR-NB.

I. INTRODUCTION

A low delay, low complexity, and low bit-rate speech coder would be attractive for Voice over IP (VoIP) [1] and Voice over Wireless LAN (VoWLAN) [2] applications. G.727 [3] is an ITU-T standard embedded Adaptive Differential Pulse Code Modulation (ADPCM) narrowband speech coder with low delay and low complexity. However, it offers high speech quality only at higher bit-rates. Adaptive Multi-Rate Narrowband (AMR-NB) is a narrowband speech coder that achieves high speech quality at lower bit-rates. However, the computational complexity and delay of AMR-NB are high. Therefore, we developed the Multimode Tree coder which achieves high speech quality with low computational complexity and delay. Compared with G.727 coder, the average bit-rate of the Multimode Tree coder is lower. Compared with AMR-NB, the computational complexity and delay of the Multimode Tree coder is lower.

The Multimode Tree coder (MMT) is a tree coder combined with multimode coding. Multimode coding is based on phonetic classification of speech. The speech is classified into different modes and each mode is coded with a suitable bit-rate. Tree coding is a delayed encoding procedure where speech samples are coded effectively based on the best long-term fit to the input waveform [1], [2]. By delayed coding, the possible reconstruction sample paths are evaluated for the set of input samples, and the best path is chosen based on suitable distortion measures which define the fit of the reconstructed samples to the input samples. A perceptual weighting filter is employed for distortion calculations. In order to reduce the computational complexity of the distortion calculation in the tree search, pre-weighting and post-weighting filters are introduced in the Multimode Tree coder. In addition, a backward pitch predictor is applied to the code generator of our tree coder. By using the backward pitch predictor, the speech quality is improved without increasing the bit-rate and delay.

The paper is organized as follows. Section II describes the details of the Multimode Tree coder with backward pitch prediction and perceptual pre- and post-weighting. The experimental results and comparison with standardized speech codecs are shown in Section III. Finally, the conclusions are presented in Section IV.

II. MULTIMODE TREE CODING FOR SPEECH WITH PERCEPTUAL WEIGHTING AND BACKWARD PITCH PREDICTOR

The block diagram of the Multimode Tree coder is shown in Fig. 1. The input speech frame is classified into five phonetic modes: Voiced (V); Onset (ON); Unvoiced (UV); Hangover (H); and, Silence (S), based on the mode decision. If the speech frame is classified as Voiced (V) or Onset (ON), the pre-weighting filter is used. The code generator in the tree coder consists of an inverse quantizer, $Q^{-1}$, a long-term pitch
The weighting filter is shown in (1). The criteria help in choosing the path with minimum cumulative distortion, we use the M-L Tree Search as the tree search algorithm. In order to improve the speech quality with low bit-rate, a pitch predictor is applied to the code generator of our tree coder. In addition, since we need a low delay code generator, a backward pitch predictor [5], [6] is employed. The pitch predictor \( P_l(z) \) is a 3-tap backward pitch predictor, which is defined as:

\[
P_l(z) = \sum_{i=-1}^{d+1} \beta_i z^{-(d+i)},
\]

where \( \beta_i \)'s are pitch coefficients and \( d \) is the pitch period.

1) Backward Pitch Estimation: The pitch period \( d_k \) at time instant \( k \) is estimated from the previous output of the pitch predictor, \( P(z) \), and a short-term adaptive predictor, \( P_s(z) \). The inverse quantizer \( Q^{-1} \) is controlled by the mode decision output. For example, when the output of mode decision is Voiced (V) or Onset (ON), the bit-rate of the quantizer is higher than the bit-rate of Unvoiced (UV) or Hangover (H). The pitch predictor is used only when the speech frame is classified as Voiced (V). The minimum cumulative distortion path is selected as the best long-term fit along \( L-1 \) delayed samples to the input waveform and the first node of the path is encoded. The mode decision output and the symbol of each path are sent to the code generator. The decoder of the Multimode Tree coder is similar to the code generator in the tree coder. The code generator is constructed based on G.727 with a backward pitch predictor. In Fig. 3 (b), the minimum cumulative distortion path, \( I_3 \rightarrow I_{15} \), is marked as search paths. Therefore, the symbol \( I_3 \) is released and encoded.

2) Pitch Predictor: The pitch predictor is a 3-tap backward pitch predictor, which is defined as:

\[
P_l(z) = \sum_{i=-1}^{d+1} \beta_i z^{-(d+i)},
\]

where \( \beta_i \)'s are pitch coefficients and \( d \) is the pitch period.

1) Backward Pitch Estimation: The pitch period \( d_k \) at time instant \( k \) is estimated from the previous output of the pitch predictor, \( P(z) \), and a short-term adaptive predictor, \( P_s(z) \). The inverse quantizer \( Q^{-1} \) is controlled by the mode decision output. For example, when the output of mode decision is Voiced (V) or Onset (ON), the bit-rate of the quantizer is higher than the bit-rate of Unvoiced (UV) or Hangover (H). The pitch predictor is used only when the speech frame is classified as Voiced (V). The minimum cumulative distortion path is selected as the best long-term fit along \( L-1 \) delayed samples to the input waveform and the first node of the path is encoded. The mode decision output and the symbol of each path are sent to the code generator. The decoder of the Multimode Tree coder is similar to the code generator in the tree coder. The code generator is constructed based on G.727 with a backward pitch predictor. In Fig. 3 (b), the minimum cumulative distortion path, \( I_3 \rightarrow I_{15} \), is marked as search paths. Therefore, the symbol \( I_3 \) is released and encoded.

B. Code Generator

The code generator of our tree coder, as shown in Fig. 4, includes an inverse quantizer \( Q^{-1} \), a backward pitch predictor \( P_l(z) \), and a short-term adaptive predictor \( P_s(z) \). Since G.727 [4] has a low delay and low complexity predictor, it is used as the short-term predictor in the code generator. Therefore, our code generator is constructed based on G.727 with a backward pitch predictor.

In order to improve the speech quality with low bit-rate, a pitch predictor is applied to the code generator of our tree coder. In addition, since we need a low delay code generator, a backward pitch predictor [5], [6] is employed. The pitch predictor \( P_l(z) \) is a 3-tap backward pitch predictor, which is defined as:

\[
P_l(z) = \sum_{i=-1}^{d+1} \beta_i z^{-(d+i)},
\]

where \( \beta_i \)'s are pitch coefficients and \( d \) is the pitch period.
After initializing the pitch period, it is recursively updated. The estimate of the autocorrelation function at lags of \( m = d_k + 1, d_k, \) and \( d_k - 1 \) is used for pitch tracking. The estimated autocorrelations \( \hat{\rho}(m) \) are obtained from the following recursion:

\[
\hat{\sigma}_e^2(k) = \delta \hat{\sigma}_e^2(k - 1) + (1 - \delta)(e'_s(k))^2,
\]

\[
\hat{\rho}(m) = \delta \hat{\rho}^{(k - 1)}(m) + (1 - \delta)\frac{e'_s(k) e'_s(k - m)}{\hat{\sigma}_e^2(k)},
\]

where \( \hat{\sigma}_e^2 \) is the estimated variance of \( e'_s \) and \( \delta = 0.95 \).

After updating the estimated autocorrelation function, the pitch period \( d_k \) increases one when the \( \hat{\rho}^{(k)}(d_k + 1) \) is the maximum of \( \hat{\rho}^{(k)}(d_k + j), j = -1, 0, 1, \) and \( \hat{\rho}^{(k)}(d_k + 1) \) is greater than \( \hat{\rho}_{\text{min}} = 0.2 \) to avoid tracking unvoiced speech. The pitch period \( d_k \) decreases one when the \( \hat{\rho}^{(k)}(d_k - 1) \) is the maximum of \( \hat{\rho}^{(k)}(d_k + j), j = -1, 0, 1, \) and \( \hat{\rho}^{(k)}(d_k - 1) \) is also greater than \( \hat{\rho}_{\text{min}} = 0.2 \). Otherwise, the pitch period is not modified.

If the pitch period is modified, then the values of the estimate of the autocorrelation function are shifted in the same way that the pitch coefficients are shifted. The new autocorrelation value, either \( \hat{\rho}^{(k)}(d_k - 1) \) or \( \hat{\rho}^{(k)}(d_k + 1) \), is computed to be a constant fraction of \( \hat{\rho}^{(k)}(d_k) \), typically 0.3.

2) Backward Pitch Coefficients Calculation: After getting the pitch period \( d \), the pitch coefficients can be calculated as follows. The initial pitch coefficients of each block are calculated using the Wiener-Hopf equation, shown in (6), where \( \mu = 0.001 \). If the initialized pitch coefficients are unstable, the stabilization procedure [7] needs to be applied to the initialized pitch coefficients. When pitch coefficients of each block are initialized, other pitch coefficients are recursively adapted by using the equation:

\[
\beta_i(k) = \lambda \beta_i(k - 1) + \alpha \frac{\sigma_u}{\sigma_{\text{ac}}(k)} u(k) e'_s(k - d_k - i), i = -1, 0, +1,
\]

where \( \lambda = 1 - 2^{-7} \) and \( \alpha = 2^{-7} \), \( u(k) \) is the output of inverse quantizer, \( \sigma_u^2 \) is the estimate of the variance of \( u(k) \), and \( \sigma_{\text{ac}}^2 \) is the estimate of the variance of \( \sigma_{\text{ac}}^2 \).

\[
\beta_{-1} = \begin{bmatrix}
(1 + \mu)R_{ee}(0) & R_{ee}(1) & R_{ee}(2) \\
R_{ee}(1) & (1 + \mu)R_{ee}(0) & R_{ee}(1) \\
R_{ee}(2) & R_{ee}(1) & (1 + \mu)R_{ee}(0)
\end{bmatrix}^{-1}
\begin{bmatrix}
R_{ee}(d - 1) \\
R_{ee}(d) \\
R_{ee}(d + 1)
\end{bmatrix}
\]

C. Perceptual Pre-weighting and Post-weighting

As mentioned in Section II-A, the computational complexity with the perceptual weighting filter inside the loop as in Fig. 2 is high. Assume that the computational complexity of \( W(z) \) is \( C \) operations, and \( B = 2^n \) is the number of children of a tree node such that if \( B = 4 \) for the \( n = 2 \) bits/sample tree, then the complexity of releasing one symbol is \( M \cdot B \cdot L \cdot C \) operations. Schuller, Yu, Huang, and Edler [8] have employed adaptive pre-filtering and post-filtering in lossless audio coding. They showed that lossless audio coding with pre- and post-filtering maintains high quality. In addition, Shetty and Gibson [3] employed perceptual pre-weighting and post-weighting in a G.726 ADPCM codec [9] and a modified AMR-NB CELP codec. They showed that the performance of lossy coding with pre- and post-weighting also improves. As shown in Fig. 5, the computational complexity of our Multimode Tree Coder is reduced to \( 2C \) operations for releasing one symbol by using pre-weighting and post-weighting filters.

The objective of pre- and post-weighting is to match the frequency response of the perceptual error weighting filter generated with 5th order LPC coefficients in (1) with the frequency response of the filter generated with G.727 ADPCM predictor coefficients. As a result, the post-weighting filter generated with G.727 ADPCM pole-zero coefficients is

\[
H_{\text{post}}(z) = \frac{1 + \sum_{i=1}^{6} m_i z^{-i}}{(1 + \sum_{i=1}^{6} m_i b_i z^{-i})(1 - \sum_{i=1}^{3} a_i z^{-i})},
\]

where \( a_i \)’s are pole coefficients, \( b_i \)’s are zero coefficients, \( m_1 = 0.2, m_2 = 1.0, \) and \( m_3 = 0.85 \) in both pre- and post-weighting filters in our experiments.

III. PERFORMANCE OF MULTIMODE TREE CODER FOR NARROWBAND SPEECH

In this section, the effects of perceptual pre- and post-weighting and backward pitch prediction are discussed. The
results show that the perceptual pre- and post-weighting and backward pitch predictor does improve the perceptual evaluation of speech quality (PESQ) of the Multimode Tree coder (MMT). Moreover, the average bit-rate, algorithmic delay, and computational complexity of the Multimode Tree coder are also analyzed. By comparing the PESQ, average bit-rate, algorithmic delay, and computational complexity of the MMT with standardized speech codecs, the low complexity and low delay characteristics of the MMT are shown.

A. Simulation Settings

The bit-rate of the Multimode Tree coder is controlled by the mode decision output. The mode decision output, the frame header, is coded with 2 bits. Since the frame length for narrowband Multimode Tree coder is 5 msec, the bit-rate of the header is 0.4 kbps.

In the tree coder, \( M = 4 \) and \( L = 10 \) for M-L Tree Search algorithm.

In order to lower the average bit-rate, the Comfort Noise Generator (CNG) motivated by the CNG of AMR-NB [10] is used for Silence (S) mode. In the CNG, the pole-zero predictor coefficients from the short-term predictor are averaged between each transmission frame and encoded every 15 frames. The absolute magnitude of each frame is averaged and transmitted every 8th and 15th frames. The bit-rate for Silence (S) mode is 0.72 kbps.

The test sequences for narrowband speech are chosen from ITU-T coded-speech database [11]. The sampling rate of each sequence is either a male or female (M/F) English clean sequence.

The performance of the narrowband speech codec is evaluated by perceptual evaluation of speech quality (PESQ), which is an objective method for narrowband speech quality assessment and is standardized by ITU-T P.862 [12].

B. Comparison with MMT, MMT with Weighting, and MMT with Weighting and Pitch

In order to investigate the influence of perceptual pre- and post-weighting and backward pitch predictor on the Multimode Tree coder, we compare the PESQ of the MMT, the MMT with perceptual pre- and post-weighting (MMT-W), and the MMT with perceptual pre- and post-weighting and backward pitch prediction (MMT-WP). The results of the MMT for narrowband sequences using 3 core bits/sample for Voiced and Onset and 2 core bits/sample for Unvoiced and Hangover are shown in Table I. Compared with the MMT, the average PESQ of the MMT-WP increases from 3.656 to 3.811. The improvement caused by perceptual pre- and post-weighting and backward pitch predictor on PESQ is about 0.155. In addition, for the sequence “we were away,” the PESQ of the MMT-W increases from 3.445 to 3.739, and the PESQ of the MMT-WP increases from 3.445 to 4.012. Since the sequence “we were away” is a fully voiced sentence, it shows that the perceptual pre- and post-weighting and backward pitch predictor improve a lot on the performance of the voiced segment.

The average bit-rate of each narrowband sequence is calculated based on the mode decision results. The bit-rate of header is 0.4 kbps, Silence is 0.72 kbps, Unvoiced and Hangover are 16 kbps, and Voiced and Onset are 24 kbps. Hence, the average bit-rate of each narrowband sequence using 3 core bits/sample on Voiced and Onset is shown in Table I. It shows that with multimode coding—Comfort Noise Generator for Silence and 2 bits/sample for Unvoiced and Hangover—decreases the average bit-rate of the MMT. However, the performance of the MMT-WP is still between fair and good, PESQ of the MMT-WP is between 3.6–4.1.

C. Estimated Computational Complexity of MMT

Based on the ITU-T Basic Operators from [13], the estimated computational complexity of each process in the Multimode Tree coder is shown in Table II. The computational complexity of G.727 is 1.25 MIPS [14], [15]. Reference [16] mentioned that “for state-of-the-art DSPs, such as the TI C55, the number of WMOPS and MIPS is similar.” Thus, we assume that the computational complexity of G.727 is 1.25 WMOPS as well.

Based on the mode decision, the estimated computational complexities of MMT, MMT-W, and MMT-WP are shown in Table III. When the sequence is 100% voiced, the computational complexity for the MMT presents the worst case. Based on the estimated computational complexity in Table II, the

---

**TABLE I**

PESQ of the MMT for Narrowband Sequences Using 3 Bits/Sample for Voiced and Onset

<table>
<thead>
<tr>
<th>Sequence</th>
<th>MMT</th>
<th>MMT-W</th>
<th>MMT-WP</th>
<th>Average bit-rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>lathe</td>
<td>3.747</td>
<td>3.847</td>
<td>3.938</td>
<td>14.55</td>
</tr>
<tr>
<td>we were away</td>
<td>3.445</td>
<td>3.739</td>
<td>4.012</td>
<td>24.12</td>
</tr>
<tr>
<td>af1s01</td>
<td>3.748</td>
<td>3.812</td>
<td>3.838</td>
<td>9.06</td>
</tr>
<tr>
<td>af1s02</td>
<td>3.719</td>
<td>3.721</td>
<td>3.784</td>
<td>9.72</td>
</tr>
<tr>
<td>af1s03</td>
<td>3.697</td>
<td>3.762</td>
<td>3.768</td>
<td>11.73</td>
</tr>
<tr>
<td>am1s01</td>
<td>3.582</td>
<td>3.582</td>
<td>3.638</td>
<td>7.90</td>
</tr>
<tr>
<td>am1s02</td>
<td>3.734</td>
<td>3.853</td>
<td>3.825</td>
<td>8.35</td>
</tr>
<tr>
<td>am1s03</td>
<td>3.577</td>
<td>3.659</td>
<td>3.681</td>
<td>9.06</td>
</tr>
<tr>
<td>Average</td>
<td>3.656</td>
<td>3.747</td>
<td>3.811</td>
<td>11.81</td>
</tr>
</tbody>
</table>

**TABLE II**

Estimated Computational Complexity (WMOPS) of MMT

<table>
<thead>
<tr>
<th>Process</th>
<th>Computational Complexity (WMOPS)</th>
<th>Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mode Decision in Encoder</td>
<td>0.0178 V, UV, ON, H, S</td>
<td></td>
</tr>
<tr>
<td>Pre-weighting filter in Encoder</td>
<td>0.208 V, ON</td>
<td></td>
</tr>
<tr>
<td>Tree Coder in Encoder</td>
<td>1.385 V, UV, ON, H</td>
<td></td>
</tr>
<tr>
<td>Pitch Predictor in Encoder</td>
<td>1.1919 V</td>
<td></td>
</tr>
<tr>
<td>Silence Encoding</td>
<td>0.0008 S</td>
<td></td>
</tr>
<tr>
<td>G.727 Decoder</td>
<td>0.625 V, UV, ON, H</td>
<td></td>
</tr>
<tr>
<td>Pitch Predictor in Encoder</td>
<td>1.1919 V</td>
<td></td>
</tr>
<tr>
<td>Post-weighting filter in Decoder</td>
<td>0.208 V, ON</td>
<td></td>
</tr>
<tr>
<td>Silence Decoding</td>
<td>0.07056 S</td>
<td></td>
</tr>
</tbody>
</table>

**TABLE III**

Estimated Computational Complexity (WMOPS) for Narrowband Sequences

<table>
<thead>
<tr>
<th>Sequence</th>
<th>MMT</th>
<th>MMT-W</th>
<th>MMT-WP</th>
</tr>
</thead>
<tbody>
<tr>
<td>lathe</td>
<td>1.31</td>
<td>1.31</td>
<td>2.63</td>
</tr>
<tr>
<td>we were away</td>
<td>2.03</td>
<td>2.43</td>
<td>4.73</td>
</tr>
<tr>
<td>af1s01</td>
<td>0.86</td>
<td>0.96</td>
<td>1.49</td>
</tr>
<tr>
<td>af1s02</td>
<td>0.93</td>
<td>1.03</td>
<td>1.61</td>
</tr>
<tr>
<td>af1s03</td>
<td>1.15</td>
<td>1.27</td>
<td>1.91</td>
</tr>
<tr>
<td>am1s01</td>
<td>0.74</td>
<td>0.82</td>
<td>1.31</td>
</tr>
<tr>
<td>am1s02</td>
<td>0.81</td>
<td>0.89</td>
<td>1.34</td>
</tr>
<tr>
<td>am1s03</td>
<td>0.89</td>
<td>0.97</td>
<td>1.45</td>
</tr>
<tr>
<td>Average</td>
<td>1.09</td>
<td>1.24</td>
<td>2.06</td>
</tr>
</tbody>
</table>
worst-case computational complexity for the MMT includes mode decision, the tree coder in the encoder, and the G.727 decoder. Therefore, the worst-case computational complexity for the MMT is 2.03 WMOPS. For the MMT-W, pre-weighting in the encoder and post-weighting in the decoder are added. Therefore, the worst-case computational complexity for the MMT-W is 2.44 WMOPS. When the backward pitch predictor is added to the encoder and the decoder, the worst-case computational complexity for the MMT-WP is 4.83 WMOPS. Since “we were away” is a fully voiced sentence, this computational complexity presents the worst case. Since the computational complexity for silence is 0.08916 WMOPS, and the probability of Silence for the sequences af1s01, af1s02, af1s03, am1s01, am1s02, and am1s03 is about 58%, the computational complexities of these sequences are less than half of the worst-case computational complexity.

D. Comparison with AMR-NB, G.727, and G.728

Table IV shows the PESQ, average bit-rate, algorithmic delay, computational complexity of standardized codecs for voiced speech. Compared with narrowband standardized codecs, the worst-case complexity of the Multimode Tree coder is one third of AMR-NB and one eighth of G.728.

<table>
<thead>
<tr>
<th>Attribute</th>
<th>MMT V:3, UV:2</th>
<th>MMT-W V:3, UV:2</th>
<th>MMT-WP V:3, UV:2</th>
<th>AMR-NB 12.2 kbps</th>
<th>G.727 24 kbps</th>
<th>G.728 16 kbps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bit-rate (kbps)</td>
<td>7.90–24.40</td>
<td>5.97–12.2</td>
<td>24</td>
<td>16</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Delay (msec)</td>
<td>6.125</td>
<td>25</td>
<td>0.125</td>
<td>&lt; 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Complexity (WMOPS)</td>
<td>0.74–2.03</td>
<td>0.82–2.44</td>
<td>1.31–4.83</td>
<td>11.9–16.7</td>
<td>1.25</td>
<td>35–40</td>
</tr>
</tbody>
</table>

In this paper, we have developed a low delay and low complexity Multimode Tree coder with perceptual pre- and post-weighting and backward pitch prediction for narrow-band speech. The results show that perceptual pre- and post-weighting filters and backward pitch prediction does improve the speech quality without increasing the bit-rate and delay for voiced speech. Compared with narrowband standardized speech codecs, the worst-case complexity of the Multimode Tree coder is one third of AMR-NB and one eighth of G.728, and the delay of the Multimode Tree coder is a quarter of AMR-NB.

REFERENCES