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 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 G.728 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 low. Compared with AMR-NB, the computational complexity and delay of the Multimode Tree coder is low.

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.

This research has been supported by NSF Grant Nos. CCF-0728646 and CCF-0917230.

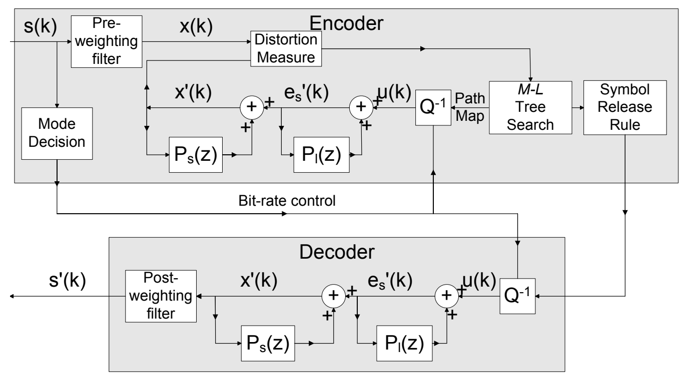

Fig. 1. Block diagram of the Multimode Tree coder with backward pitch predictor and perceptual pre- and post-weighting filters

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

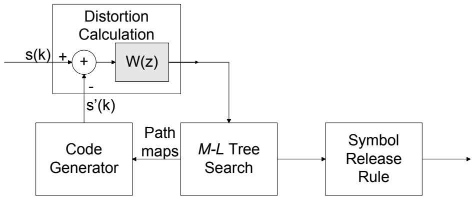

Fig. 2. Block diagram of the tree coder with perceptual weighting filter

predictor, $P_l(z)$, 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 sample are transmitted. The decoder of the Multimode Tree coder is similar to the code generator in the tree coder. The pitch predictor is used for Voiced (V) frames, and the post-weighting filter is applied when the speech frame is Voiced (V) or Onset (ON).

A. Tree Coding

Fig. 2 shows the block diagram of a tree coder. A tree coder consists of a code generator, a tree search algorithm, distortion calculations, and a path map symbol release rule. The tree search algorithm, in combination with the code generator and appropriate distortion measure, chooses the best candidate path to encode the current input sample. The symbol release rule decides the symbols on the best path to encode.

The distortion between the candidate output $s'(k)$ and the input sample $s(k)$ is computed by filtering the error between them along the depth-L path through the perceptual error weighting filter shown in (1). The criteria help in choosing the path where the noise is masked by the speech spectrum. The weighting filter is

$$W(z) = \frac{1 - \sum_{i=1}^{N} a_i z^{-i}}{1 - \sum_{i=1}^{N} \mu^i a_i z^{-i}}, \tag{1}$$

where the value of μ is 0.86, a_i's are the LPC coefficients calculated from the current speech frame, and the value of N is 5. The distortion values are stored along each searched path map. The path resulting in minimum cumulative distortion is encoded using a symbol release rule. However, the distortion calculation along each path obtained by filtering the error along depth-L path through the perceptual error weighting filter in (1) is computationally expensive. Therefore, perceptual pre-weighting and post-weighting filters [3] are used. The details of perceptual pre- and post-weighting filters are described in Section II-C.

In order to reduce the computational complexity of searching the path with minimum cumulative distortion, we use the *M-L* Tree Search as the tree search algorithm. The searching is on a 2^n-ary tree with depth L, where n is the bit-rate of each sample. The M paths with minimum cumulative distortion are chosen and extended along their children. The minimum cumulative distortion path among the M depth-L paths is chosen and the symbol corresponding to the first node in the minimum cumulative distortion path is encoded. For example, there are 4^L paths of a tree generated with a 2 bits/sample coder as shown in Fig. 3. Assume L is 3 and M is 8, the eight minimum cumulative distortion paths, $I_1 \rightarrow I_6$, $I_1 \rightarrow I_7$, $I_2 \rightarrow I_9$, $I_2 \rightarrow I_{11}$, $I_2 \rightarrow I_{12}$, $I_3 \rightarrow I_{15}$, $I_4 \rightarrow I_{17}$, and $I_4 \rightarrow I_{18}$, with their children are marked as search paths in Fig. 3 (a). Based on *M-L* Tree Search, we only need to maintain M minimum cumulative distortion paths instead of 4^L paths, which reduces the computational complexity for the tree search. In Fig. 3 (b), the minimum cumulative distortion path, $I_3 \rightarrow I_{15}$, is marked. Therefore, the symbol I_3 is released and encoded.

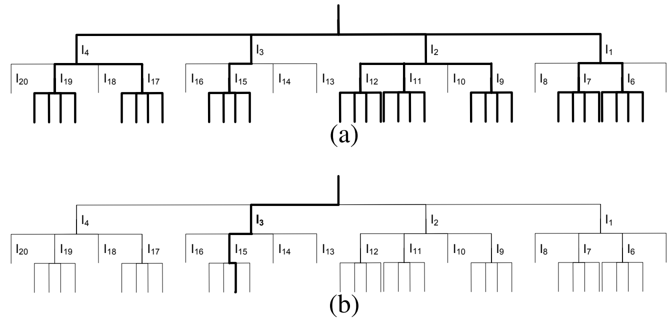

Fig. 3. An example of 2 bits/sample tree (a) Search paths of *M-L* Tree Search for $L = 3$ and $M = 8$ (b) Symbol release rule

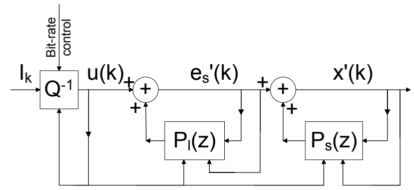

Fig. 4. Block diagram of the code generator with backward pitch predictor

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:

$$P_l(z) = \sum_{i=-1}^{i=+1} \beta_i z^{-(d+i)}, \tag{2}$$

where β_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, $e'_s(j)$, $j < k$. The pitch period is initialized every T samples and recursively updated sample-by-sample. The pitch period is initialized by calculating the autocorrelation function R_{ee} and finding the maximum of R_{ee} between $d_{\min} = 20$ and $d_{\max} = 125$. The autocorrelation function R_{ee} is calculated from $e'_s(j)$, $j = k - T, k - T + 1, ..., k - 1$. T is 240 in our experiments for narrowband speech. In addition, when the pitch period increases or decreases by one and the shifted coefficients are not stable, as determined by the stability test [7], the pitch period and pitch coefficients are initialized again by calculating the autocorrelation function R_{ee} and finding the maximum of R_{ee}. The pitch period d_k is initialized by:

$$d_k = \underset{d_{\min} \leq m \leq d_{\max}}{\arg\max} \; R_{ee}(m). \tag{3}$$

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:

$$\hat{\sigma}_e^2(k) = \delta\hat{\sigma}_e^2(k-1) + (1-\delta)(e'_s(k))^2, \tag{4}$$

$$\hat{\rho}^{(k)}(m) = \delta\hat{\rho}^{(k-1)}(m) + (1-\delta)\frac{e'_s(k)e'_s(k-m)}{\hat{\sigma}_e^2(k)}, \tag{5}$$

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 $\hat{\rho}_{\min} = 0.2$ 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 $\hat{\rho}_{\min} = 0.2$. 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:

$$\beta_i(k) = \lambda\beta_i(k-1) + \frac{\alpha}{\sigma_u(k)\sigma_e(k)}u(k)e'_s(k-d_k-i), i = -1, 0, +1, \tag{7}$$

where $\lambda = 1 - 2^{-7}$ and $\alpha = 2^{-7}$, $u(k)$ is the output of inverse quantizer, σ_u^2 is the estimate of the variance of $u(k)$, and σ_e^2 is the estimate of the variance of the output of pitch predictor $e'_s(k)$. The estimated variances are calculated using (4). If the updated pitch coefficients are not stable, then no update is performed on the pitch coefficients. When the pitch period is either incremented or decremented by one, the pitch coefficients are shifted by one in the appropriate direction. The new coefficient, either β_{-1} or β_{+1} is computed as a constant fraction of β_0, typically 0.67. If the updated coefficients are unstable, the pitch period and the pitch coefficients are re-initialized by (3) and (6), respectively.

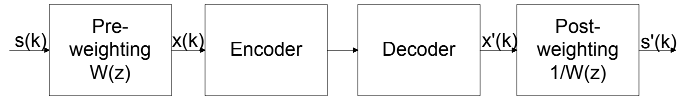

Fig. 5. Pre-weighting and post-weighting filters for speech codec

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

$$H_{post}(z) = \frac{1 + \sum_{i=1}^{6} m_2^i b_i z^{-i}}{(1 + \sum_{i=1}^{6} m_3^i b_i z^{-i})(1 - \sum_{i=1}^{2} m_1^i a_i z^{-i})}, \tag{8}$$

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

$$\begin{pmatrix} \beta_{-1} \\ \beta_0 \\ \beta_{+1} \end{pmatrix} = \begin{pmatrix} (1+\mu)R_{ee}(0) & R_{ee}(1) & R_{ee}(2) \\ R_{ee}(1) & (1+\mu)R_{ee}(0) & R_{ee}(1) \\ R_{ee}(2) & R_{ee}(1) & (1+\mu)R_{ee}(0) \end{pmatrix}^{-1} \begin{pmatrix} R_{ee}(d-1) \\ R_{ee}(d) \\ R_{ee}(d+1) \end{pmatrix} \tag{6}$$

TABLE I
PESQ OF THE MMT FOR NARROWBAND SEQUENCES USING 3 BITS/SAMPLE FOR VOICED AND ONSET

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

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 is 4 and L is 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 8 kHz, and 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.

TABLE II
ESTIMATED COMPUTATIONAL COMPLEXITY (WMOPS) OF MMT

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

TABLE III
ESTIMATED COMPUTATIONAL COMPLEXITY (WMOPS) FOR NARROWBAND SEQUENCES

Sequence	MMT	MMT-W	MMT-WP
lathe	1.31	1.51	2.63
we were away	2.03	2.43	4.73
af1s01	0.86	0.96	1.49
af1s02	0.93	1.03	1.61
af1s03	1.15	1.27	1.91
am1s01	0.74	0.82	1.31
am1s02	0.81	0.89	1.34
am1s03	0.89	0.97	1.45
Average	1.09	1.24	2.06

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 IV
COMPARISON OF MMT, AMR-NB, G.727, AND G.728

Attribute	MMT V:3, UV:2	MMT-W V:3, UV:2	MMT-WP V:3, UV: 2	AMR-NB 12.2 kbps	G.727 24 kbps	G.728 16 kbps
PESQ	3.445–3.748	3.582–3.853	3.638–4.012	3.602–4.136	3.243–3.814	3.825–4.127
Bit-rate (kbps)	7.90–24.40			5.97–12.2	24	16
Delay (msec)	6.125			25	0.125	< 2
Complexity (WMOPS)	0.74–2.03	0.82–2.44	1.31–4.83	11.9–16.7	1.25	35–40

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 the MMT, MMT-W, MMT-WP, AMR-NB at 12.2 kbps, G.727 at 24 kbps, and G.728 at 16 kbps. The PESQ and the average bit-rate in Table IV represent a range of eight test sequences. The algorithmic delay and computational complexity of standardized codecs are summarized in [17]. Even though the average bit-rate of the MMT is 0.25–1.5 bits/sample higher than that of AMR-NB, the algorithmic delay and computational complexity of the MMT are much lower than those of AMR-NB. The algorithmic delay of the MMT is 6.125 msec while that of AMR-NB is 25 msec. The algorithmic delay of the MMT is about a quarter of AMR-NB. The worst-case computational complexity of the MMT-WP is 4.83 WMOPS, while the worst-case computational complexity of AMR-NB is 16.7 WMOPS, which shows that the MMT reduces computational complexity about 70%. Compared to G.728, the PESQ and average bit-rate of the MMT-WP are comparable with those of G.728. Even though the algorithmic delay of the MMT is larger than that of G.728, the worst-case computational complexity of the MMT is only one eighth of G.728.

IV. CONCLUSIONS

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 narrowband 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

[1] A. Goris and J. D. Gibson, "Incremental tree coding of speech," *IEEE Trans. on Information Theory*, vol. 27, no. 4, pp. 511 – 516, July 1981.

[2] H. C. Woo and J. D. Gibson, "Low delay tree coding of speech at 8 kbit/s," *IEEE Trans. on Speech and Audio Processing*, vol. 2, no. 3, pp. 361 –370, July 1994.

[3] N. Shetty and J. D. Gibson, "Perceptual Pre-weighting and Post-inverse weighting for Speech Coding," *the 41st Annual Asilomar Conference on Signals, Systems, and Computers*, Nov. 4-7, 2007.

[4] ITU-T Recommendation G.727, "5-, 4-, 3- and 2-bit/sample embedded Adaptive Differential Pulse Code Modulation (ADPCM)," Dec. 1990.

[5] R. Pettigrew and V. Cuperman, "Backward pitch prediction for low-delay speech coding," in *Proc. GLOBECOM*, Nov. 1989.

[6] V. Cuperman and R. Pettigrew, "Robust low-complexity backward adaptive pitch predictor for low-delay speech coding," *IEE Proceedings-I*, vol. 138, no. 4, pp. 338 – 344, Aug. 1991.

[7] R. Ramachandran and P. Kabal, "Stability and performance analysis of pitch filters in speech coders," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. 35, no. 7, pp. 937 – 946, Jul. 1987.

[8] G. Schuller, B. Yu, D. Huang, and B. Edler, "Perceptual Audio Coding using Adaptive Pre- and Post-Filters and Lossless Compression," *IEEE Trans. on Speech and Audio Processing*, vol. 10, no. 6, pp. 379 –390, Sept. 2002.

[9] ITU-T Recommendation G.726, "40, 32, 24, 16 kbit/s Adaptive Differential Pulse Code Modulation (ADPCM) ," Dec. 1990.

[10] 3GPP, "Mandatory Speech Codec speech processing functions; Adaptive Multi-Rate (AMR) speech codec; Comfort noise aspects," 3rd Generation Partnership Project (3GPP), TS 26.092, Dec. 2009.

[11] ITU-T Recommendation P. Supplement 23, "ITU-T coded-speech database," Feb. 1998.

[12] ITU-T Recommendation P.862, "Perceptual Evaluation of Speech Quality (PESQ), an objective method for end-to-end Speech Quality Assessment of Narrow-band telephone networks and Speech Codecs," Feb. 2001.

[13] ITU-T Recommendation G.191 STL-2009 Manual, "ITU-T Software Tool Library 2009 User's manual," Nov. 2009.

[14] J. D. Gibson, "Speech Coding Methods, Standards, and Applications," *IEEE Circuits and Systems Magazine*, vol. 5, no. 4, pp. 30 – 49, Fouth Quarter 2005.

[15] ——, "Speech Coding for Wireless Communications," in *Mobile Communications Handbook*. CRC Press, 2012.

[16] L. Hanzo, F. C. Somerville, and J. Woodard, *Voice and Audio Compression for Wireless Communications*. John Wiley & Sons, Inc. New York, NY, USA, 2007.

[17] ITU-T SG16 Q7/16, "Media Coding Summary Database," Jul. 2010. [Online]. Available: http://www.itu.int/ITU-T/studygroups/com16/mcsd/mscd.html