

# MULTIMODE TREE CODING OF SPEECH WITH PERCEPTUAL PRE-WEIGHTING AND POST-WEIGHTING

Pravin Ramadas, Ying-Yi Li, and Jerry D. Gibson

Department of Electrical and Computer Engineering, University of California, Santa Barbara, USA

Email: pravin\_ramadas@umail.ucsb.edu, yingyi\_li@umail.ucsb.edu, gibson@ece.ucsb.edu

## ABSTRACT

A low delay and low complexity speech coder based on Multimode Tree Coding is proposed. In our Multimode Tree Coder, a simple mode classification method along with frame energy are used to classify the input speech frames into five different modes. Each mode is coded at a suitable bit-rate using a Tree coder with computationally efficient perceptual error pre-weighting and post-weighting filters, and a G.727 Code Generator. At an average bit-rate about 12 kbps for 50% Voice Activity sequences, the variable rate Multimode Tree Coder produces speech quality equivalent to the G.727 ADPCM coder at 32 kbps, resulting in nearly 63% bit-rate saving. Apart from bit-rate savings, a low delay of 6.125 ms, and low complexity are achieved by our Multimode Tree Coder, making it an interesting alternative to the AMR-NB Codec.

## KEY WORDS

Multimode Tree Coder,  $M-L$  Tree Search, Pre-weighting and Post-weighting, Speech Coding

## 1 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 are high. Therefore, we developed a Multimode Tree Coder which achieves high speech quality with low computational complexity and delay. Compared to the G.727 coder, the average bit-rate of our Multimode Tree Coder is low, and it fills an important gap between CELP-based codecs and G.727.

The proposed coder is based on Multimode classification and Tree 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 [4, 5]. By delayed coding, the possible reconstruction sample paths are evalu-

ated for the set of input samples, and the best path is chosen based on suitable distortion measure which defines the fit of the reconstructed samples to the input samples. In order to reduce the computational complexity of the distortion calculation in the Tree Search, we introduce pre-weighting and post-weighting filters in our Multimode Tree Coder.

The input speech signal is classified into five phonetic modes. Each mode is coded suitably using a Tree coder with computationally efficient perceptual error pre-weighting and post-weighting filters. The Tree coder uses an  $M-L$  Tree Search and a G.727 Code Generator. G.727 is a low complexity ADPCM coder. In addition, the  $M-L$  Tree Search limits the number of search paths in the Tree coder. Most importantly, pre- and post-weighting filters reduce the computational complexity of the distortion measure calculation. In our Tree coder, the mode decision is made every 40 samples, so the delay of the Multimode Tree Coder is only 5 ms. Hence, our Tree coder achieves low complexity and low delay. The resulting variable rate coder achieves speech quality equivalent to 32 kbps G.727 at an average bit-rate of about 12 kbps for 50% Voice Activity Sequences.

The details of Multimode Tree Coding are described in Section 2. Section 3 describes the pre- and post-weighting operations. The results of our proposed coder are shown in Section 4. Finally, conclusions are in Section 5.

## 2 Multimode Tree Coder

Tree coding is a multi-path search procedure to encode each speech sample based on the best long term fit to the input waveform. An ADPCM coder encodes each input sample at time instant  $k$ , using the data at time  $j \leq k$ . Tree coders improve the approach by delaying the encoding decision  $L$  samples, such that the input samples at time  $j \leq k + L$  are used to encode the sample at time instant  $k$ . By this delayed decision, a Tree coder can search along different possible encoding sequences, called paths, before encoding the current sample. The fit of the reconstructed samples to the input samples and the consequent path selection is defined by a suitable error measure.

A Tree coder consists of a Code Generator, a Tree Search algorithm, a distortion measure and a path map symbol release rule. The Tree Search algorithm, in com-

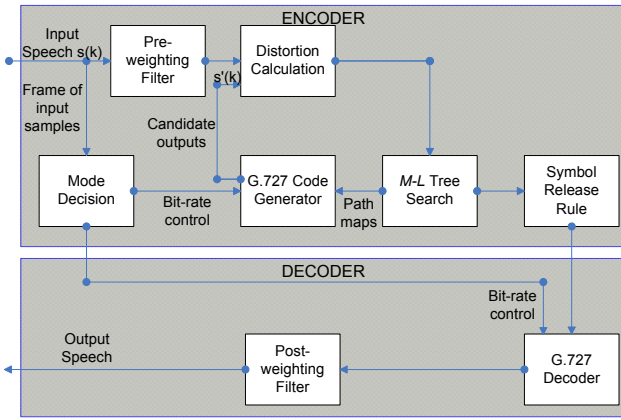


Figure 1. Multimode Tree Coder with pre-weighting and post-weighting

bination 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.

In a Multimode Tree Coder, the rate of coding each input sample is controlled based on the mode decision of the current frame of input samples. The block diagram of the Multimode Tree Coder is shown in Figure 1. The mode decision is made at the beginning. Based on the result of the mode decision, the Code Generator codes the sample at a suitable bit-rate. Then the distortion between candidate outputs and input samples is calculated via the Tree Search. Finally, the symbol relative to the minimum distortion is released. At the mode boundaries, the Tree coder uses the samples of the next mode into the Tree while a sample in the current mode is still being coded. Since the current symbol is encoded considering the cumulative distortion measure due to the samples of the next mode also, which are already in the Tree path pipeline, the transition at the mode boundaries is smooth. The details of each block in Figure 1 is discussed in the following sections.

## 2.1 Mode Decision

The mode decision is a low delay, low complexity method based on the ADPCM coder state parameters, step-size scale factor ( $y$ ), and long-term average magnitude of weighted quantization level ( $d_{ml}$ ), along with frame energy ( $f_e$ ). The input to the mode decision is a speech frame of 40 samples and the output is the classification of the frame among one of these five modes: Voiced (V), Onset (ON), Unvoiced (UV), Hangover (H), and Silence (S). The input frame of speech is first classified into Voice/Silence by Voice Activity Detection (VAD) before further classification into five modes. In the following, the term "Voice" is used to denote all non-silence speech and the term "Voiced" is used to denote the Voiced (V) mode of speech.

The step-size factor ( $y$ ) is updated for the adaptive quantizer in the G.727 ADPCM coder. The long-term average magnitude of weighted quantization level ( $d_{ml}$ ) is calculated for the adaptation speed of speech in the coder. The frame energy ( $f_e$ ) is widely used in VAD algorithms such

as G.729 VAD. It is computed for each frame  $i$  as follows,

$$f_e(i) = \sum_{k=i}^{i+39} s^2(k), \quad (1)$$

where  $s$  is the input speech signal.

The step-size scale factor ( $y$ ), long-term average magnitude of weighted quantization level ( $d_{ml}$ ), and frame energy ( $f_e$ ) are high during Voice sequence and low during Silence sequence. Therefore, the three parameters are compared against threshold values computed for each parameter to make a Voice Activity Detection. If  $y$  or  $d_{ml}$  is greater than its threshold ( $y_{VAD}$  or  $d_{VAD}$ ), and  $f_e$  is greater than its threshold ( $f_{VAD}$ ), then the frame is marked as Voice. If  $y$  and  $d_{ml}$  are smaller than their thresholds ( $y_{VAD}$  and  $d_{VAD}$ ), and the status continues for at least 15 frames, then the frame is marked as Silence.

After Voice Activity Detection, each frame should be further classified into Voiced (V), Onset (ON), Unvoiced (UV), or Hangover (H). For the frame marked as Voice, it will be further classified into Unvoiced (UV), Onset (ON) or Voiced (V). If  $y$ ,  $d_{ml}$ , and  $f_e$  are smaller than its threshold ( $y_{VOICE}$ ,  $d_{VOICE}$ , and  $f_{VOICE}$ ), the frame is classified as Unvoiced (UV). Otherwise, it is classified as Voiced (V). The first Voiced frame following Unvoiced or Silence frame is marked as Onset (ON). For the frame marked as Silence, it will be further classified into Silence (S), or Hangover (H). The first ten Silence frames following Voice are marked as Hangover (H). Otherwise, it is marked as Silence (S).

## 2.2 Code Generator

We use an G.727 ADPCM coder [3] as the Code Generator in our Multimode Tree Coder. It is an embedded ADPCM coder, providing coding rates of 5, 4, 3, and 2 bits/sample. In our Multimode Tree Coder, the Voiced (V) and Onset (ON) modes are coded at 24 kbps, and the Unvoiced (UV) and Hangover (H) modes are coded at 16 kbps. G.727 ADPCM coder uses 2-poles and 6-zeros for adaptive prediction.

## 2.3 M-L Tree Search

All the possible paths along the depth  $L$  search are coded by the Code Generator and stored in a tree. For instance, there are  $4^L$  possible paths for the 16 kbps ADPCM Code Generator. An example of a Tree generated with a 16 kbps ADPCM Code Generator for depth  $L = 2$  is shown in Figure 2. The optimal path through the tree to encode the current sample is one of these  $4^L$  paths. Since the computational complexity is high for exhaustive searching, the  $M$ - $L$  Tree Search algorithm is used. The  $M$ - $L$  Tree Search algorithm limits the number of paths extended in the Tree Search to only the  $M$  most likely paths instead of all the  $4^L$  possible paths. The  $M$  paths with minimum cumulative distortion are chosen and extended along their siblings, and each extended path has a path map along which the reconstructed values will be generated by the Code Generator.

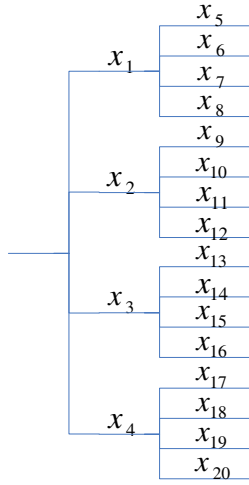


Figure 2. Tree generated by a 16 kbps ADPCM Code Generator for depth  $L = 2$

## 2.4 Distortion Calculation with Pre-weighting

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 Eq. (2). The criteria helps in choosing the path where the noise is masked by the speech spectrum. The weighting filter is

$$W(z) = \frac{1 - \sum_{i=1}^N a_i z^{-i}}{1 - \sum_{i=1}^N \mu^i a_i z^{-i}}, \quad (2)$$

where the value of  $\mu$  is 0.86,  $a_i$ 's are the short term predictor coefficients calculated from the current speech frame. The value of  $N$  is 5 in our Coder. 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 Tree path obtained by filtering the error along depth- $L$  path through the perceptual error weighting filter in Eq. (2) is computationally expensive. Assume the computational complexity of error weighting filter is  $C$  operations, then the computational complexity of releasing one output symbol is  $M \cdot B \cdot L \cdot C$  operations, where  $B$  is the number of siblings along which the  $M$  chosen paths are extended. The perceptual error weighting filter  $W(z)$  can be rearranged to reduce the computational complexity. The distortion measure using perceptual error weighting filter  $W(z)$  is

$$W(z)[s(z) - s'(z)], \quad (3)$$

which is equivalent to

$$W(z)s(z) - W(z)s'(z). \quad (4)$$

When the input sample  $s(z)$  is pre-filtered with  $W(z)$ , the reconstructed sample at the Code Generator is close to  $W(z)s'(z)$  [6]. Therefore, filtering the error along the Tree paths and then computing the minimum mean

square (MMSE) values of the filtered error is similar to pre-weighting the input and computing just the MMSE along the Tree paths. In this case, the computational complexity reduces to  $C$  operations to release a symbol. Because the reconstructed output at the decoder corresponds to the input  $W(z)s(z)$  instead of  $s(k)$ , the decoder output needs to be post-filtered by  $\frac{1}{W(z)}$  to get the desired output. The computational complexity of the weighting filter at both the encoder and decoder is  $2C$  operations. Therefore, the distortion measure computed by pre-weighting and post-filtering achieves low computational complexity. The pre-filtering is used only in Onset (ON) and Voiced (V) modes. The details of designing the perceptual pre-weighting and post-weighting filters is described in Section 3.

## 2.5 Symbol Release Rule

A symbol release rule defines how many symbols of the path are released after selecting the optimal path. In our Multimode Tree Coder, we use the single symbol release rule. The symbol corresponding to the first node in the minimum error path is encoded. After the current sample is encoded,  $M$  minimum error paths for encoding the next symbol are chosen and the iteration is repeated until all the samples are encoded.

## 2.6 Silence Coding and Frame Generator

During Silence (S) frames, the pole-zero predictor coefficients from the G.727 32 kbps coder are averaged between each transmission frame and encoded differentially every 15th frame. As mentioned before, the G.727 ADPCM coder uses 2-pole and 6-zero coefficients. The frame energy  $f_e$  is differentially encoded in dB every 8th and 15th frame. At the decoder, the intermediate silence frames use the recently received predictor and energy information. Pole coefficients are transmitted with 7 bits each while zero coefficients are transmitted with 5 bits each. Frame energy is transmitted with 5 bits. The total transmitted bits for every 15 frames is 54 bits. Hence, the bit-rate of Silence coding is 0.72 kbps.

Two header bits, 00 – Unvoiced (UV), 01 – Voiced (V), 10 – Onset (ON), and 11 – Silence (S), are used to identify each mode. The header bits are transmitted for every frame. Silence parameters are transmitted only every 8th and 15th frame. Other Silence frames carry the header information only.

## 2.7 Decoder with Post-weighting

Each encoded frame is decoded by the G.727 decoder at the rate decided by the header of each frame. Since the pre-weighting operation is performed only during Onset (ON) and Voiced (V) frames, post-weighting is also performed only during Onset (ON) and Voiced (V) frames. The details of the post-weighting filter design is explained in Section 3.

## 3 Pre-weighting and Post-weighting

In order to reduce the complexity of computing the short term prediction coefficients  $a_i$ 's in Eq. (2), the pole and

zero predictor coefficients corresponding to the last node of the optimal path in the Tree are used to form the pre-weighting filter for the incoming sample. The pre-weighting filter is designed based on the frequency response of the predictor, and the post-weighting filter is designed as the inverse of the pre-weighting filter.

The design of the pre-weighting filter  $W(z)$  and post-weighting filter  $\frac{1}{W(z)}$  is to mask the reconstruction error at the output by the input spectrum. Let  $S(z)$  be the input speech,  $X(z)$  be the pre-weighted speech,  $X'(z)$  be the pre-weighted speech output, and  $S'(z)$  be the output speech after post-weighting. Then the relation of  $S(z)$  and  $X(z)$  is

$$S(z)W(z) = X(z), \quad (5)$$

and the relation of  $S'(z)$  and  $X'(z)$  is

$$X'(z)\frac{1}{W(z)} = S'(z). \quad (6)$$

Let  $E(z)$  denote the coding error for the pre-weighted speech. From Eq. (5) and Eq. (6), the coding error  $E(z)$  will be

$$\begin{aligned} E(z) &= X(z) - X'(z) \\ &= W(z)[S(z) - S'(z)]. \end{aligned} \quad (7)$$

$W(z)$  is used to shape the reconstruction error [6].

The objective is to match the frequency response of perceptual error weighting filter generated with 5th order LPC coefficients in Eq. (2) with the frequency response of the filter generated with ADPCM predictor coefficients. Since the pre-weighting and post-weighting filter are the inverse of each other, we only compare the post-weighting filter configurations. The post-weighting filter of 5th order LPC coefficients is

$$\frac{1}{W(z)} = \frac{1 - \sum_{i=1}^5 (0.86)^i a_i z^{-i}}{1 - \sum_{i=1}^5 a_i z^{-i}} \quad (8)$$

while the post-weighting filter generated with ADPCM pole-zero coefficients is

$$H_{post}(z) = \frac{1 + \sum_{i=1}^6 m_2^i b_i z^{-i}}{(1 + \sum_{i=1}^6 m_3^i b_i z^{-i})(1 - \sum_{i=1}^2 m_1^i a_i z^{-i})} \quad (9)$$

The comparison of different post-weighting filter configurations is shown in Figure 3. It shows that the spectrum of larger  $m_3$  has better tilt compensation while  $m_1$  and  $m_2$  are fixed. Also, the spectrum of larger  $m_1$  has wider low-pass filtering while  $m_2$  and  $m_3$  are fixed. Hence, a large value of  $m_3$  and a small value of  $m_1$  is used in our filter implementation. After analyzing the filter response due to several combinations of  $m_1$ ,  $m_2$ , and  $m_3$ , we decided the values as  $m_1 = 0.2$ ,  $m_2 = 1.0$ , and  $m_3 = 0.85$  in both pre- and post-weighting filters.

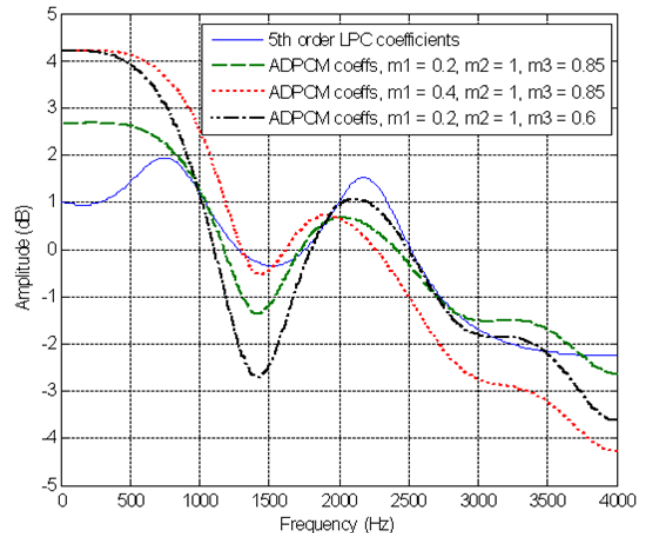


Figure 3. Comparison of different post-weighting filter configurations

## 4 Experiment Results

In our experiments, the depth of tree,  $L$ , is 10. Hence, an additional 9 samples are buffered for Tree coder look-ahead. Apart from the 40 sample frame, the 9 look-ahead samples belonging to the next frame also need to be coded at the rate of the current frame since it becomes a part of the Tree coder while encoding the last sample of the current frame. This is because the mode decision of the 9 samples belonging to the next frame is not available yet. Moreover, the value of  $M$  influences the Tree Search algorithm and complexity. The  $M$  value is 4 in our current codec. The chosen value of  $M$  helps in maintaining low encoder complexity, at the same time achieving good coding efficiency. The Voiced (V) and Onset (ON) modes are coded at 24 kbps while Unvoiced (UV) and Hangover (H) modes are coded at 16 kbps.

### 4.1 Results

The test sequences [7] for the AMR-NB coder provided on the ITU-T website are used for testing the Multimode Tree Coder for clean sequences. PESQ [8] is used for evaluating the quality of the Narrowband coder. We compare the result of Multimode Tree Coder with a G.727 Coder Generator with the AMR-NB coder at 12.2 kbps, and the G.727 ADPCM coder at 32kbps. The results are shown in Table 1.

The evaluation of the performance for noisy sequences uses the clean versions of the noisy sequence as the reference and the decoded noisy sequence as the degraded sequence. Since the clean version of the noisy sequences are not available for the AMR-NB test sequences, we use a different set of sequences for noisy sequences. The result of noisy sequences is shown in Table 2.

The details of the test sequences are listed below:

- Clean Sequences – Language, Gender of the speaker
  1. T04 – Spanish, Female
  2. T05 – Spanish, Male
  3. T06 – English, Female
  4. T07 – English, Female
  5. T08 – English, Female
  6. T12 – English, Male
  7. T13 – English, Male
- Noisy Sequences – Language, Gender of speaker, SNR
  1. L1 – English, Female, 10 dB Train noise
  2. L2 – English, Female, 15 dB Train noise
  3. L3 – English, Female, 20 dB Train noise
  4. W1 – English, Female, 10 dB Airport noise
  5. W2 – English, Female, 15 dB Airport noise
  6. W3 – English, Female, 20 dB Airport noise
  7. F1 – English, Male, 10 dB Car noise
  8. F2 – English, Male, 15 dB Car noise
  9. F3 – English, Male, 20 dB Car noise

Table 1. Comparison of performance of Multimode Tree Coder, AMR-NB coder, and G.727 ADPCM coder for clean sequences

Sequence (PESQ)	MM Tree Coder	AMR-NB 12.2 kbps	G.727 32 kbps
T04	3.844	3.772	3.771
T05	3.958	4.029	3.901
T06	3.820	3.875	4.002
T07	3.912	3.732	4.007
T08	4.001	4.064	4.087
T12	3.848	3.898	3.819
T13	3.924	4.104	3.852
average	3.901	3.925	3.920

From Table 1 and Table 2, we see that the speech quality of the Multimode Tree Coder is equivalent to AMR-NB coder at 12.2 kbps and G.727 coder at 32 kbps for both clean and noisy sequences. However, the complexity and the delay of the Multimode Tree Coder are lower than those of AMR-NB. In addition, the average bit-rate of the Multimode Tree Coder is lower than that of the G.727 coder. The analysis of delay, computational complexity, and average bit-rate for 50% Voice Activity sequences is described in Section 4.2

#### 4.2 Analysis of Delay, Complexity, and Bit-rate

The delay in the encoder is due to the 40 samples required

Table 2. Comparison of performance of Multimode Tree Coder, AMR-NB coder, and G.727 ADPCM coder for noisy sequences

Sequence (PESQ)	MM Tree Coder	AMR-NB 12.2 kbps	G.727 32 kbps
L1	2.719	2.812	2.677
L2	3.036	3.150	2.989
L3	3.265	3.387	3.290
W1	2.520	2.566	2.488
W2	2.720	2.794	2.717
W3	3.018	3.008	2.930
F1	2.447	2.487	2.318
F2	2.707	2.743	2.587
F3	2.973	2.987	2.852
average	2.823	2.882	2.761

for the mode decision of a frame and 9 samples required for the Tree coder look-ahead. This results in a total delay of 6.125 ms.

For the analysis of complexity and average bit-rate, we assume that the speech sequences are 50% Voice Activity sequences. In addition, we also assume that 40% of the frames are in Voiced (V) or Onset(ON) and 10% of the frames are in Unvoiced (UN) or Hangover (H). Since Silence coding is 0.72 kbps, Voiced (V) and Onset (ON) frames are coded at 24 kbps, and Unvoiced (UV) and Hangover (H) frames are coded at 16 kbps, the average bit rate of our Multimode Tree Coder for 50% Voice Activity sequences is 11.96 kbps.

The complexity due to each function in the Multimode Tree Coder is given in Table 3. In a Tree coder, each path is ADPCM coded and the extension to each sibling is additionally generating reconstruction values for different quantization levels, which involve calling the set of functions associated with sample reconstruction for each sibling. Since we maintain a small  $M$  value of 4, we restrict the complexity due to the Tree coder.

Pre- and post-filtering optimize the perceptual error weighting for computational complexity as explained in Section 2.4. Since the mode decision procedure uses the ADPCM parameters itself, apart from only frame energy values, their complexity is very small. Moreover, Silence (S) is encoded by differential predictors and frame energy as explained in Section 2.6. Therefore, the complexity due to silence encoding procedure is also low.

At the decoder, apart from the regular ADPCM decoder complexity, the additional complexity is due to post-filtering, and synthesis filtering and frame energy scaling. Therefore, the complexity of the codec is very small, 3.161 wMOPS (50% Voice Activity) and 5.472 wMOPS (100% Voice Activity, Voiced and Unvoiced are 80% and 20% respectively). ITU-T standard on weights of basic operators [9] has been used to measure the complexities. The memory requirements of the speech codec is expected to be very small compared to CELP coders such as AMR-NB.

The Multimode Tree Coder with a G.727 Code Gen-

Table 3. Computational complexity in wMOPS due to each function in the Multimode Tree Coder and decoder (for 50% Voice Activity Sequences. Voiced and Unvoiced are 40% and 10% respectively. )

Function	Complexity (wMOPS)	Probability	Complexity
Mode Detection	0.68	1	0.68
MM Tree Coder	3.65	0.5 (V+UV)	1.825
Silence Encoding	0.01	0.5 (S)	0.005
Pre-filter	0.32	0.4 (V)	0.128
Decoder	0.63	0.5 (V+UV)	0.315
Silence Decoding	0.16	0.5 (S)	0.08
Post-filter	0.32	0.4 (V)	0.128
Codec Complexity			3.161

erator achieves quality comparable to the 32 kbps G.727 ADPCM coder at only 11.96 kbps in average rate for 50% Voice Activity sequences, resulting in nearly 63% bit-rate reduction. In addition, the complexity is only 3.161 wMOPS and the delay is low at 6.125 ms. Comparisons of the Multimode Tree Coder with AMR-NB and G.727 coders for 50% Voice Activity sequences for important speech coder attributes is shown in Table 4. Although the average bit-rate of the Multimode Tree Coder is higher than AMR-NB, a popular CELP coder, it has significant advantages in terms of delay and complexity as seen in Table 4. When compared to the G.727 coder, the Multimode Tree Coder has higher delay and complexity but has significantly lower bit-rate.

Table 4. Comparison of the speech coder attributes of the Multimode Tree Coder, AMR-NB coder, and G.727 ADPCM coder for 50% Voice Activity

Speech Coder Attributes	MM Tree Coder	AMR-NB 12.2 kbps	G.727 32 kbps
Avg Bit-rate (kbps)	11.96	<7	32
Delay (ms)	6.125	20	0.125
Complexity (wMOPS)	3.161	16.75 [10]	1.25 [11]

## 5 Conclusion

The proposed Multimode Tree Coder combines the principle of Multimode classification with Tree coding. Multimode coding reduces the average bit-rate of the coder. Even though Tree Coding is a delayed coding, frame delay and look-ahead delay are still low. Since G.727 ADPCM is

a low complexity coder, the Tree coder with a G.727 Code Generator has low computational complexity. Moreover, the *M-L* Tree Search reduces the computational complexity of Tree Search; the perceptual weighting achieved by using pre-weighting and post-weighting is also computationally efficient.

From the experiments, the Multimode Tree Coder with a G.727 Code Generator achieves speech quality similar to 32 kbps G.727 an average bit-rate of 11.96 kbps for 50% Voice Activity sequences, which is much lower than that of G.727. In addition, the delay and the computational complexity are lower than AMR-NB at 12.2 kbps. Thus, our Multimode Tree Coder has low delay, low complexity, and a low relatively average bit-rate. These features make it attractive for VoIP and VoWLANs.

## Acknowledgements

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

## References

- [1] D. Minoli and E. Minoli, *Delivering Voice over IP networks* (New York: John Wiley and Sons Inc., 1998).
- [2] J. Geier and J. T. Geier, *Implementing High Performance IEEE 802.11 Networks* (Indianapolis: Sams, 2001).
- [3] ITU-T Recommendation G.727, 5-, 4-, 3- and 2-bit/sample embedded Adaptive Differential Pulse Code Modulation (ADPCM), 1990.
- [4] A. Goris and J. D. Gibson, Incremental tree coding of speech, *IEEE Trans. on Information Theory*, 27(4), 1981, 511-516.
- [5] H. C. Woo and J. D. Gibson, Low delay tree coding of speech at 8kbit/s, *IEEE Trans. on Speech and Audio Processing*, 2(3), 1994, 361-370.
- [6] 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*, 2007.
- [7] 3GPP, AMR speech Codec; Test sequences, 3rd Generation Partnership Project (3GPP), TS 26.074, 2007. Available: <http://www.3gpp.org/ftp/Specs/html-info/26074.htm>
- [8] 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, 2001.
- [9] ITU-T G.191 STL-2005 Manual, ITU-T Software Tool Library 2005 User's Manual, 2005. Available: <http://www.itu.int/rec/T-REC-G.191/en>
- [10] 3GPP TSG-S4, AMR Wideband Permanent project document WB-4: Design Constraints, v.1.0, 2000.
- [11] J. D. Gibson, Speech coding methods, standards, and applications, *IEEE Circuits and Systems Magazine*, 5(4), 2005, 30-49.