

# Phonetically Switched Tree coding of speech with a G.727 code Generator

Pravin Ramadas and Jerry D. Gibson

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

Email: {pravin\_ramadas, gibson}@ece.ucsb.edu

## Abstract

Several improvements done to Phonetically-Switched G.726 based ADPCM coder<sup>[1]</sup> (PS-ADPCM) reported previously have resulted in enhanced speech quality, bit-rate scalability, and good tandem performance while reducing the delay, and bit-rate. In this improved coder, a simple phonetic classification method classifies speech into different modes. Each mode is coded at appropriate bit-rate using Tree based G.727 ADPCM coder (PS-Tree coder) with perceptual error weighting distortion measure resulting in significantly improved speech quality at a low average bit-rate. This improved PS-tree coder results in speech quality comparable to G.727 ADPCM at 32 kbps but at an average bit-rate about 16 kbps while encoding a typical telephone conversation. A simple Comfort Noise Generation procedure is used to improve the perceptual quality during silence mode.

## I. Introduction

G.727 is an embedded Adaptive Differential Pulse Code Modulation (ADPCM) ITU-T standard, widely used in digital telephony applications. It provides the bit-rate scalability option through its embedded Quantizer structure. Hence we use G.727 instead of G.726 used in our previous work to have bit-rate scalability.

Phonetic classification of speech has been used effectively for low bit-rate coding of speech<sup>[7,8]</sup>. This method helps in identifying different phonetic classes in speech and coding them appropriately with sufficiently enough bits in order to reduce the overall bit-rate. Our previous work<sup>[1]</sup> used phonetic classification of speech and coding each phonetic class appropriately using G.726 based Adaptive Differential Pulse Code Modulation (PS-ADPCM) coder. This resulted in speech quality comparable to G.726 ADPCM at 24 kbps but at an average bit-rate less than 16 kbps while encoding a typical telephone conversation.

In the previous work, we had (a) used a complex phonetic segmentation algorithm which also introduced a delay of 40 ms in the encoder (b) used single path search based ADPCM encoder, allowing degradation in speech quality due to instantaneous spurious samples (c) encoded frames of same mode contiguously with unique state parameters, requiring separate set of state parameters to be stored for each mode, introducing memory overhead. This method also required smoothing along the frame boundaries of different modes leading to quality loss (d) silence frames were removed with zeros, affecting the quality of noisy speech sequences.

The major modifications in the current PS-Tree coder to overcome the above list of problems are summarized as follows, (a) a simple phonetic segmentation procedure based on reconstruction error energy is used and the delay is reduced to 11.25 ms (b) a multi-path search procedure - Tree coding, is used which encodes speech samples based on the best long term fit thereby improving the speech quality (c) continuous encoding of the speech sequence is done instead of encoding each mode separately. This could be done because tree coder has the virtue of adapting itself to fast growing sample values across boundaries of different modes eliminating the need to code each mode uniquely thereby eliminating the need to preserve state parameters for different modes (d) a simple comfort noise generation procedure is used to improve the perceptual quality during silence. These improvements result in speech quality equivalent to 32 kbps G.727 at an average bit-rate about 16 kbps.

This paper is organized as follows. Section II describes the phonetic classification procedure. Section III describes the mode based tree coding procedure. Section IV presents the comfort noise generation method. Section V presents the speech quality and bit-rate improvements achieved by this coder in comparison with PS-ADPCM coder and G.727 coder at 32 kbps. Section VI presents the tandem performance of PS-Tree coder.

## II. Phonetic classification procedure

Our previous work<sup>[1]</sup> used phonetic information in speech to achieve low bit-rate ADPCM speech coding. It used a robust Voice

This research has been supported by the California Micro Program, Applied Signal Technology, Cisco, Qualcomm Inc, and Sony-Ericsson, and by NSF Grant Nos. CCF-0429884, CNS-0435527, and CCF-0728646.

Activity Detection method, AMR-VAD1, in combination with a phonetic segmentation method. After speech was classified into silence/voice activity region by the VAD, seven feature values were extracted from the voice activity region and a weighted linear combination of the values was used to further classify the region into voiced/unvoiced. The set of features are: zero-crossing rate, low-band speech energy, first reflection coefficient, preemphasized energy ratio, second reflection coefficient, forward pitch prediction gain and backward pitch prediction gain. In addition to the complexity of computing the feature values, the method introduced an encoder delay of 40 ms.

In the current encoder, Phonetic classification of voice activity region is simplified as follows: Speech samples are grouped into frames of 90 samples each and coded with G.727 ADPCM at 16 kbps. The error between the original and reconstructed speech is calculated for the frame. The error energy is computed for the frame and compared with the threshold obtained by the weighted combination of the error energy in the previous speech frames and average Voiced/Unvoiced error energy, to make a Voiced/Unvoiced decision as shown in fig 1. Note that there could be misalignment between AMR-VAD1 which makes VAD decision on 160 speech samples and the phonetic classification method which takes 90 sample frames. In such cases, if the previous VAD decision is Voice, the same decision is made for the current frame by the phonetic classification method. Otherwise if the previous VAD decision is silence, the current frame energy is compared with the frame energy of the previous silence frames for Voice/Silence decision. A Voice decision is made if the frame energy is higher than the threshold obtained from previous Silence frame energies. The complete phonetic classification procedure as explained above is shown in fig 1.

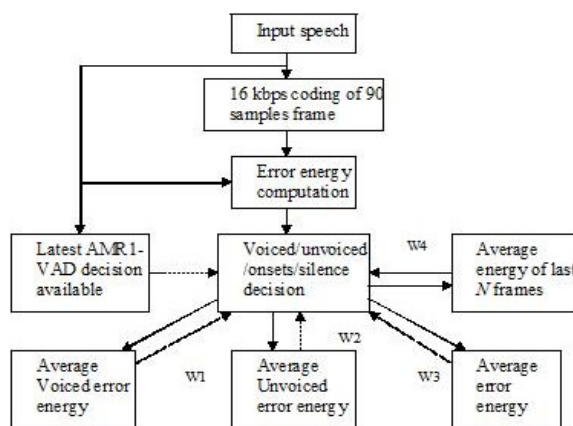


Fig 1. Phonetic Classification procedure for silence, Voiced, Unvoiced mode classification. W's are the weights given to each parameter to form the decision threshold.

This phonetic classification method classifies higher reconstruction error regions as Voiced, which are coded at high bit-rate and lower reconstruction error regions as Unvoiced, which are coded at low bit-rate. This method ensures that different regions in speech are coded at sufficient bit-rate and the over all reconstruction error is reduced. The Voiced/Unvoiced decision made, closely approximates the phonetic classification done in our previous work<sup>[1]</sup>. Fig 2 shows the phonetic classification result for a sample speech sequence. A mis-classified Voiced region is also indicated in the figure, which does not affect the speech quality since the region is still sufficiently coded.

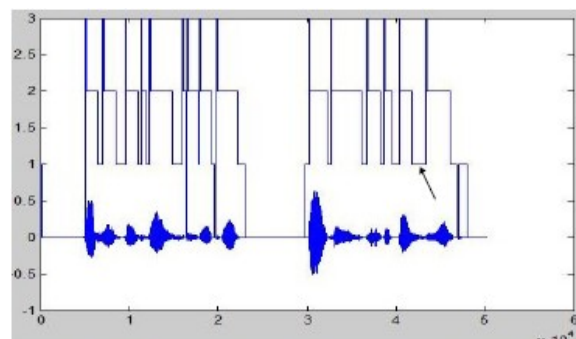


Fig 2. Phonetic Classification result for the sequence 'Acid burns holes in old cloth'. Segments enclosed with value 3 are onsets, 2 are voiced, 1 are unvoiced and 0 are silence. The region marked by the arrow is Voiced classified as Unvoiced but speech quality is not affected by this misclassification.

Framing method consists of the usage of a header bit to identify the mode of the current frame for decoding as explained in our previous work<sup>[1]</sup>. The first two Voiced frames following unvoiced or silence frames are marked as onsets. The first five silence frames are marked as hangover frames for smooth transition from Voice to Silence and for estimation of parameters required to reconstruct the background noise. This phonetic classification method reduces computational complexity and encoder delay to 11.25 ms.

### III. Mode based G.727 Tree coding

Tree coding is a multi-path search procedure to encode each speech sample based on best long term fit to the input waveform. Tree coder encodes each input sample at time instant  $k$ , using only the data at times  $j \leq k$ . Tree coders improve on this approach by delaying the encoding decision by few more samples, say  $L$ , such that the input samples at time instants  $j \leq k + L$  are used to encode the sample at time instant  $k$ . By this, delayed decision tree coder searches for all (or most likely) among the  $2^L$  possible paths to come up with best fit for the

current sample. The fit and the consequent path selection is defined by suitable error metric.

Design of tree coder consists of selecting a code generator, a tree search algorithm, a distortion measure and a path map symbol release rule as shown in fig 3 [2]. Tree search, in combination with code generator and appropriate distortion calculation method, chooses the best candidate path to encode the current input sample  $s(k)$ . The Symbol release rule decides on the symbol(s) to be encoded in order to reconstruct the sample at the decoder.

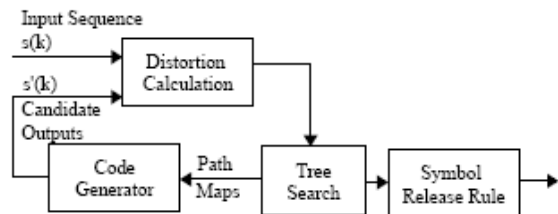


Fig 3. Block diagram of Tree coder

In our coder the following methods are used for each tree coder block: G.727 ADPCM encoder is used as the code generator. ML-tree search algorithm is used to reduce the computational complexity by limiting the multi-path search to  $M$  most likely paths rather than all  $2^L$  possible paths.  $M=10$  and  $L=10$  are used in this encoder. Perceptual weighted error is used as the distortion measure. This criteria helps in choosing the path where the noise is masked by the speech spectrum. Distortion measure is obtained by filtering the prediction errors along depth- $L$  path through the Perceptual error weighting filter as shown [2]:

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

where the value of  $\mu$  is 0.86.  $a_i$ 's are the predictor coefficients calculated in the speech frame. Value of  $N$  is 5. Single symbol release rule is used as path map symbol release rule.

The phonetically-switched tree-based G.727 ADPCM coder (PS-Tree coder) is shown in fig 4. Voiced and Onset frames are coded at 32 kbps while Unvoiced and hangover frames are coded at 16 kbps, with this PS-Tree coder. Since the tree coder adapts itself to effectively look into the raise and falls in sample values across different mode boundaries, frames of similar mode need not be coded separately eliminating the need to store the mode specific state parameters and also eliminating the need for any median smoothing at the mode boundaries as done in our previous work [1], thereby improving the speech

quality. Silence frames are encoded using Comfort Noise parameters which are explained in detail in section IV. The frames are packed with appropriate header bits to identify the mode information at the decoder, in the bit packing step.

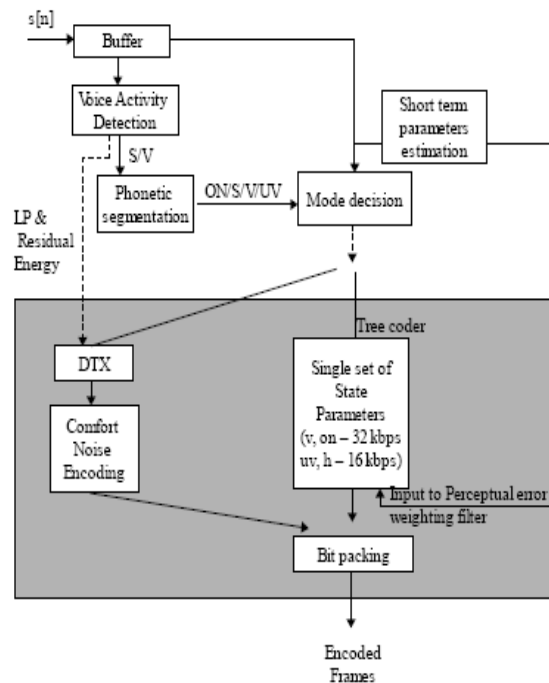


Fig 4. Phonetically-Switched tree-based G.727 ADPCM coder

The mode based G.727 decoder is shown in fig 5. Based on the mode information decoded from the header bit of the frame, G.727 ADPCM decoder functions at appropriate bit-rate to decode Voiced, Onset and Unvoiced frames. Silence frames are reconstructed using Comfort Noise Generation procedure.

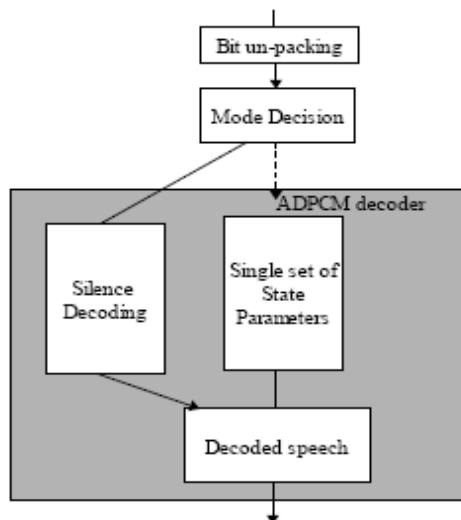


Fig 5. Mode based G.727 decoder

## IV. Comfort Noise Generation

In PS-ADPCM coder the silence mode samples are removed at the encoder and inserted with zeros at the decoder. This is perceptually unpleasant due to sudden drop in energy level. To improve the perceptual quality, a simple comfort noise generation procedure is used.

This CNG procedure is based on G.711 Appendix II<sup>[3]</sup>, G.729B CNG<sup>[4]</sup> and AMR-CNG<sup>[5]</sup>. Unlike G.729B<sup>[4]</sup>, our method does not use code vectors to increase the whiteness of the excitation hence it does not have to store any codebook, and LP coefficients are scalar quantized rather than LSP representation for computational simplicity.

Silence frames are represented by Linear Prediction (LP) and Residual Energy parameters in Silence Descriptor (SID) frames. Unlike voiced speech which needs large number of LP coefficients to represent vocal tract resonance, glottal pulse and radiation load for accurate modeling, fewer LP coefficients can sufficiently model noise and hence 8 coefficients are used in this CNG method. A Discontinuous transmission (DTX) scheme evaluates the background and decides when a SID frame has to be transmitted. DTX algorithm used is similar to G.729B CNG.

At the encoder shown in fig 4, DTX algorithm measures spectral distortion and frame energy similar to G.729B CNG<sup>[4]</sup>, except that the spectral distortion is now calculated on for 8 LP coefficients instead of 10 and the same Itakura distance threshold is used. When the DTX indicates a significant change, an updated SID frame is transmitted and a minimum spacing of two frames is imposed between consecutive SID frames to save bandwidth during non-stationary noise.

In a SID frame, the autocorrelation values of previous six frames are averaged, and LP coefficients are calculated. These are converted into PARCORs<sup>[3]</sup>, if any of the PARCOR exceeds absolute value of 1.0 then the LP filter is unstable and previous SID frame LP values are used. Each LP coefficient is represented by a 6 bit value. The residual energy is scalar Quantized using a 5-bit nonuniform quantizer<sup>[4]</sup>. Hence, 6 bits each for 8 PARCORs and 5 bits for Residual energy are allocated in a SID frame. When the DTX does not indicate a significant change, a Silence Not Updated (SNU) frame is transmitted instead of SID frame. During SNU frame the previously received SID information is used. During silence, SNU and SID frames are distinguished at the decoder using an additional header bit.

At the decoder shown in fig 5, to avoid abrupt level transition of noisy frames, smoothing procedure for excitation gain<sup>[4]</sup> is used. To increase the

excitation whiteness, a white Gaussian excitation is prestored. The excitation is scaled according to the residual energy and filtered through the LP synthesis filter obtained back from the PARCORs. The LP synthesis filter coefficients are interpolated with previous LP synthesis filter coefficients to minimize the effect of spectral distortion due to quantization of PARCORs.

## V. Results

In a typical telephone conversation silence is present nearly 50% of the time and is encoded at <1 kbps, Unvoiced and Voiced are present at 25% of the time each which are encoded at 16 kbps and 32 kbps respectively. This ideally achieves a bit-rate of about 12 kbps. However, assuming higher Voiced decisions than Unvoiced, an average bit-rate of about 16 kbps can be achieved. List of sentences used in the experiments are:

1. "Acid burns holes in old cloth. Fairy tales are fun to read"
2. "Oak is strong and also gives shade"
3. "A lathe is a big tool"
4. "Wipe the grease off your dirty face"

The first two sequences represent male voice and the second two represent female voice. The sequences used are clean without any back ground noise in table 1 and with back ground noise in table 2.

Table 1 compares the PESQ-MOS values of G.727 at 32 kbps with PS-ADPCM coder and PS-Tree coder for clean speech sequences. The PS-Tree coder performs significantly better than PS-ADPCM coder and its speech quality is comparable to 32 kbps G.727 coder. Clearly there is almost 50% savings in bit-rate by using PS-Tree coder at an encoder delay of 11.25 ms.

Filename	G.727 32 kbps	G.727 PS- Tree Coder	G.727 PS- ADPCM
Fairytales	3.936	3.933	3.481
Oak	3.941	4.196	3.832
Lathe	3.993	3.887	3.390
Wipeface	3.956	3.906	3.565

Table 1 Comparison of PESQ-MOS values of G.727 at 32 kbps with PS-ADPCM coder and PS-Tree coder for clean speech sequences.

Table 2 compares the PESQ-MOS values of G.727 at 32 kbps with PS-Tree coder for noisy speech sequences. Again G.727 PS-Tree Coder performs similar to G.727 at 32 kbps.

Filename	G.727 24 kbps	G.727 32 kbps	G.727 PS- Tree Coder
Fairytales	3.031	3.225	3.219
Oak	3.583	3.746	3.856
Lathe	3.177	3.501	3.473
Wipeface	3.139	3.323	3.332

Table 2 Comparison of PESQ-MOS values of G.727 at 24 kbps, 32 kbps with PS-Tree coder for noisy speech sequences.

## VI. Tandem Performance

As a result of growing heterogeneous networking environment with each network likely to use different speech codecs, it is important to make sure the end-to-end speech quality is not affected significantly due to the asynchronous tandem operation of different speech codecs. The degradation is particularly due to transcoding at network interfaces and source coding distortion accumulation due to repeated coding<sup>[6]</sup>. To ensure that the improved PS-tree coder maintains acceptable tandeming with certain commonly used narrow-band speech codecs such as AMR-NB (at 12.2 kbps) and G.729 (at 8 kbps), tandem experiments are performed and the results are compared with the tandem performance of G.727 at 32 kbps. All inputs are clean speech.

Table 3 represents the PESQ-MOS results of the tandem performance of G.727 (at 32 kbps) with AMR and G.729. Table 4 represents the PESQ-MOS results of the tandem performance of PS-Tree coder (at avg. bit-rate 16 kbps) with AMR and G.729. The first row of tables represent the order of tandeming.

File	G727- AMR	G727- G729	G727- G727	AMR- G727	G729- G727
Fairy	4.103	3.792	3.726	3.782	3.642
oak	3.812	3.581	3.885	3.785	3.599
lathe	3.976	3.456	3.789	3.766	3.538
wipe	4.052	3.809	3.768	3.886	3.720

Table 3 Tandem performance of G.727 at 32 kbps

File	PS- AMR	PS- G729	PS- PS	AMR- PS	G729- PS
Fairy	3.981	3.755	3.932	3.673	3.576
oak	4.056	3.781	4.096	3.798	3.601
lathe	3.754	3.564	3.887	3.786	3.498
wipe	3.935	3.713	3.800	3.843	3.650

Table 4 Tandem performance of PS-Tree coder

By comparing tables 3 and 4, it can be seen that the PS-tree coder tandem performance is very close

to the performance of G.727 at 32 kbps and results in good speech quality. Hence PS-Tree coder has a good tandem performance with commonly used narrow-band coders.

## VI. Conclusion

The speech quality is improved significantly in PS-Tree coder from our previous work<sup>[1]</sup> featuring PS-ADPCM coder. The coder achieves speech quality equivalent to 32 kbps G.727 ADPCM coder at an average bit-rate of about 16 kbps, resulting in almost 50% saving in bit-rate. We have also introduced a simple method to phonetically classify speech which reduces computational complexity and encoder delay to 11.25 ms compared to 40 ms delay in our previous work. This fully-backward adaptive speech coder now has the option of bit-rate scalability and also has a good tandem performance with other popular narrow-band speech coders. A simple comfort noise generation method is also explained in this paper.

## VII. References

- [1] Pravin Ramadas and Jerry D. Gibson, "A phonetically switched ADPCM speech coder", Asilomar Conference on Signals, Systems and Computers, October 26 - October 29, 2008
- [2] J.D. Gibson and W.-W. Chang, "Fractional rate multitree speech coding", IEEE Transactions on communications, vol. 39, No. 6, June 1991
- [3] ITU-T G.711 Appendix II: A Comfort noise payload definition for ITU-T G.711 use in packet-based multimedia communication systems
- [4] ITU-T Recommendation G.729, Coding of speech at 8 kbits/s using conjugate-structure algebraic-code-excited linear prediction (CS-ACELP)
- [5] 3GPP TS 26.092 v7.0.0 (2007-06), "Adaptive Multi-Rate (AMR) speech codec; Comfort noise aspects".
- [6] Jerry D. Gibson and Bo Wei, "Tandem Voice Communications: Digital Cellular, VoIP, and Voice over Wi-Fi", IEEE Communications Society, Globecom 2004
- [7] S. Wang and A. Gersho, "Phonetically-based vector excitation coding of speech at 3.6 kbit/s," Proc. IEEE Intern. Conference on Acoustics, Speech, and Signal Processing, Glasgow, May 1989.
- [8] Shihua Wang and A. Gersho, "Improved Phonetically- Segmented Vector Excitation Coding at 3.4 Kb/s," Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, San Francisco, vol. 1, pp. 349-352, March 1992.
- [9] N. S. Jayant and S. A. Christensen, "Tree-Encoding of speech using the (M, L)- Algorithm and Adaptive Quantization", IEEE Transactions on Communications, Vol. COM-26, NO. 9, September 1978.