TTS BASED VERY LOW BIT RATE SPEECH CODER

Ki-Seung Lee and Richard V. Cox

AT&T Labs-Research, SIPS Lab 180 Park Avenue, Florham Park, NJ 07932
email : [kseung, rvc]@research.att.com

ABSTRACT

This paper addresses a speech coder which uses a Text-To-Speech (TTS) synthesis system to achieve very low bit rates (sub 1kbps). The main issue of the work is the accurate coding of the pitch(f_0) and gain contours which are principle components of prosody. This is of paramount interest since the correct prosody will increase naturalness and an efficient coding scheme will provide high coding gain. Together with the phonetic transcription, the f_0 and gain contour constitute the parameters that are necessary for the TTS system to synthesize the speech signal. Piecewise linear approximation is used to code the f_0 parameter. A technique which minimizes bit rate while maintaining f_0 error below a given threshold are described. To obtain both high compression and smoothly changing gain contours, the variance of the signal is averaged over each half phoneme length is transmitted as gain information.

With single speaker stimuli, and a priori text transcription information, we obtained natural sounding speech at an average bit rate of about 300 bps.

I. INTRODUCTION

Contemporary speech coders such as CELP, MELP, MBE or WI provide good quality speech at bit rates as low as 2.4 kbps. However, for very low bit rates on the order of 100 bps, these coders are unable to produce high quality speech, due to the reduced number of bits available for accurate modeling of the signal. In an effort to overcome this limitation a new speech coder is proposed. This coder employs a new paradigm using TTS and is meant for applications where there are no delay or complexity limitations[1]-[6].

A segmental vocoder[1] is an example of such a coder. The large lengths of the segments (typically much larger than the frame length of a conventional speech coder), contributes to the high compression ratio of these coders. A phonetic vocoder[2] is also a segmental vocoder, where the individual segments are quantized using a phonetic inventory. This coder transmits the phonetic indices, pitch periods and gain (optional) that are the necessary parameters for the TTS synthesizer. Our work is focused on the implementation of a TTS based very low bit rates speech coder. More specifically, our aim is to develop a speech coder for KNOWN text, that is able to provide high quality speech at extremely low bit rate.

In TTS, synthesized speech can be produced by concatenating the waveforms of units selected from a large database. Prosody modification is often included as a post-processor of TTS system. This typically adjusts the time scale and/or pitch to adjust prosody. Thus, a TTS based coding scheme can be thought of as a speech coder that has a very large VQ codebook composed of raw speech signals with additional parameters for compensating prosodic difference between the synthesized and the original speech signal.

Our emphasis in this work is mainly on the coding of the f_0 parameter and gain, which would ideally makes it possible to further reduce the number of bits with an acceptable level of coding error. The principle of our coding scheme is to construct a piecewise linear approximation of f_0 which replaces the original f_0 contour by consecutive lines. We present methods of finding optimal sequences of f_0 points that provide good approximation to the original f_0 contour using a rate-distortion criterion. In this approach, we consider not only approximation error but also the required number of bits for representing an f_0 contour. Since both cannot be simultaneously minimized, our criterion will be to minimize the number of bits while the maximum approximation error is maintained below a given threshold.

We found that transmitting the averaged variance over a half phoneme length gave good results for bit rates and the quality of synthesized speech. In computing gain modification factors in TTS, adaptive interpolation is employed to prevent undesired gain jumps around phoneme boundaries.

To implement a full version of the TTS coder, we consider the remaining issues including compression of phonetic information. A Harmonic plus Noise Model (HNM)[7] based half phoneme synthesizer was used as the TTS system in our speech coder.

This paper is organized as follows. Section II gives an overview of our TTS coder. The method of coding phonetic transcription is introduced in Section III. The overall $f0$ coding scheme method is presented in Section IV. Gain coding and adjustment techniques are described in Section V. Simulation results and concluding remarks are presented in Sections VI and VII.

II. OVERVIEW OF THE TTS CODER

The functional block diagram of the proposed TTS coder is shown in Fig. 1. The input speech is fed through a prosody analysis stage that computes the f_0 and gain every 10ms. We used the f_0 estimation method adopted in the Waveform Interpolation coder[8]. The whole f_0 contour is split into a series of syllabic f_0 contours by a segmentation procedure. Each syllabic f_0 contour is then stylized, and the resulting small number of samples are quantized.

The phonetic analysis is the process of finding the corresponding sequences of phones and the time alignment between the transcription and the waveforms. This results in a sequence of phonemes and durations. Output from the phonetic analysis is quantized and transmitted with quantized f_0 and Gain information.

The decoding process is essentially the inverse sequence of the encoding process.

As shown in Figure 1, the phonetic analysis requires a text transcription. In this work, the text transcriptions are assumed to be available a priori.

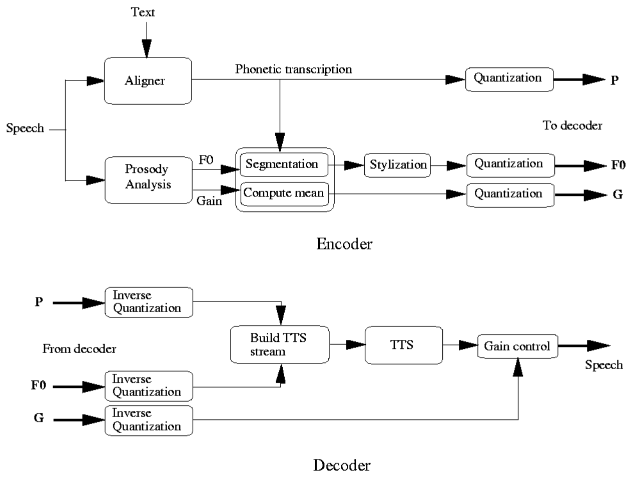

Figure 1: The block diagram of TTS coder.

III. CODING PHONEME INFORMATION

Lossless coding is applied to the phonetic transcription so that we can avoid intelligibility loss due to the coding error in the phoneme information. The phoneme information consists of phone identification and its duration. Therefore two quantizers are designed for encoding phoneme information.
The 50 basic phones of American English including silence and pause are used to represent phone identification. Thus 6 bits are used for quantizing phone id. By examining the duration distribution for typical phonemes shown in Fig. 2, one can conclude that different bit allocation for each group of phonemes increase coding efficiency. To this end, we classified all the phonemes into 3 classes according to the duration distribution and assigned different bits to each group.

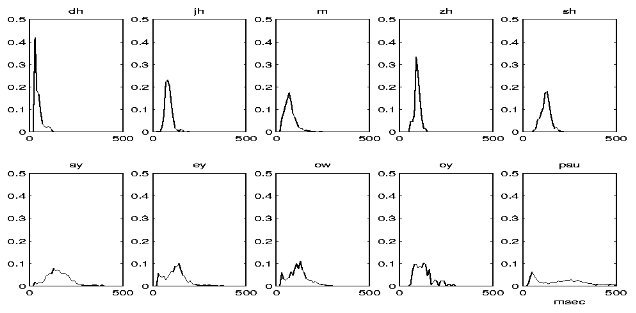

Figure 2: The duration distributions of typical phone. These results are obtained from a total of 485 sentences with 15044 phones

IV. CODING THE f_0 CONTOUR

In order to get a high compression ratio, our f_0 coding is performed contour-wise rather than frame-wise. Piecewise linear approximation (PLA) is used to implement the contour-wise f_0 coding. The use of PLA makes it possible to get high compression ratios because only a small number of sampled points need to be transmitted instead of all the individual samples.
In order to use PLA, the function being approximated has to possess some degree of smoothness. The f_0 contour reveals discontinuities around some syllable boundaries. This means that better results can be obtained by applying PLA to a syllabic f_0 contour rather than the whole f_0 contour. Consequently, segmentation is required prior to the use of PLA.
Gross representation of the f_0 contour by piecewise linear approximation causes larger coding error compared to frame-wise coding. This error depends on how to select f_0 samples as endpoints of approximation lines. Therefore, an optimizing PLA is formulated for finding the locations of f_0 points by minimizing error between the f_0 contour and approximated contour. In the following, we propose an optimal method which takes into account not only approximation error but also the number of bits required to code the parameters.
Let $\mathbf{P} = \{p_0, ..., p_{N_p-1}\}$ denote a set of f_0 points used to approximate contour, which is also an ordered set, with N_p, the total number of f_0 points in $\mathbf{P}$, and kth line starting at p_{k-1} and ending at p_k. Since $\mathbf{P}$ is an ordered set, the ordering rule and the set of points uniquely define the approximated contour.
Now, we define the constrained minimization problem

$$\text{Minimize } R(\mathbf{P}) \text{ subject to } D_{max}(\mathbf{P}) \leq d^*_{max} \qquad (1)$$

where d^*_{max} is a maximum allowable error, $R(\mathbf{P})$ is the total number of bits needed to encode the f_0 set $\mathbf{P}$ including values and positions, and $D_{max}(\mathbf{P})$ is the overall maximum absolute error defined by

$$D_{max}(\mathbf{P}) = \max_{k \in [1,\ldots,N_p-1]} d_{max}(p_{k-1}, p_k) \qquad (2)$$

where $d_{max}(p_{k-1}, p_k)$ is the maximum absolute error between the line p_{k-1} to p_k and the actual f_0 values. To find an easier way to solve the problem, we rewrite $R(\mathbf{P})$ in equation (1) as follows:

$$R(\mathbf{P}) = \sum_{k=0}^{N_p-1} \omega(p_{k-1}, p_k) \qquad (3)$$

where

$$\omega(p_{k-1}, p_k) = \begin{cases} \infty & \text{if } d_{max}(p_{k-1}, p_k) \geq d^*_{max} \\ r(p_{k-1}, p_k) & \text{otherwise} \end{cases} \qquad (4)$$

where $r(p_{k-1}, p_k)$ is the number of bits needed to encode line p_{k-1} to p_k.
Now, the problem can be formulated in the form of a directed graph, as shown in Fig. 3. The vertices of the graph correspond to the admissible f_0 points, and edges correspond to the possible segments of the approximation line. The edge have weights $\omega(p_{k-1}, p_k)$. The total number of bits $R(\mathbf{P})$ is proportional to the number of points N_p. Thus the problem can be considered as a problem of finding a shortest path. Note that the above definition of the weight function leads to a length of infinity for every path that includes a line segment, resulting in an approximation error larger than d^*_{max}.
A Dynamic Programming is employed to find the optimum path. It first finds the "*local minimal path*" for all f_0 points within a

syllabic contour, then the global minimum path is built by backtracking. The overall procedure for finding an optimal set of f_0 points $\mathbf{P}^* = \{p_0^*, ..., p_{N_p^*-1}^*\}$ is thus as follows

$$\begin{aligned} w(n) &= \arg\min_{1\le k<n} \{\alpha_{k,n}[R(p_k) + r(p_k, p_n)]\} \\ R(p_n) &= R(p_{w(n)}) + r(p_{w(n)}, p_n) \\ D_{max}(p_n) &= \max\{D_{max}(p_{w(n)}), d_{max}(p_{w(n)}, p_n)\} \end{aligned}$$

where $1 \le n \le N-1$ and N is the total number of f_0 samples within a syllabic contour, $R(p_k)$ is the accumulated number of bits up to p_k, similarly, $D_{max}(p_k)$ is the maximum error up to p_k, and $\alpha_{k,n}$ is given by

$$\alpha_{k,n} = \begin{cases} \infty & \text{if } \max\{D_{max}(p_k), d_{max}(p_k, p_n)\} > d_{max}^* \\ 1 & \text{otherwise} \end{cases}$$

The backtracking pointer, $w(n)$ holds an indication of which f_0 point is the starting point of the path with the minimum accumulated number of bits at p_n. The optimal sequence of f_0 points in reverse order is

$$p_N^*,\ p_{w(N)}^*,\ p_{w(w(N))}^*, \dots$$

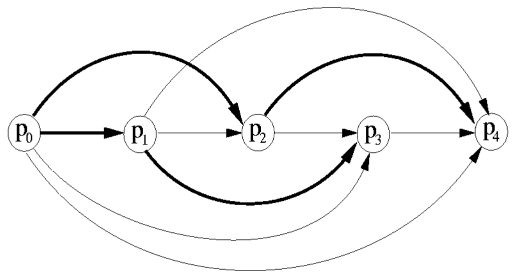

Figure 3: An example of directed graph for the linear approximation. The bold line means local minimal path.

V. GAIN CODING

In our coder, the gain is given by the mean of the variance of the individual frames associated with a given half phoneme. That is,

$$g_i = \frac{1}{N_i} \sum_{m=0}^{N_i-1} \sigma_m^2 \tag{5}$$

where g_i is the gain for i-th half phoneme and N_i is the total number of frames within i-th half phoneme. To get plausible gain information, silence-like frames are removed while computing gain. Instead of energy, variance is used in Eq. (5) in order to remove DC bias which disturbs proper gain adjustment. At the decoder, transmitted gain is used to compute the gain modification factor for the TTS algorithm. The gain modification factor for the i-th half phoneme is given by

$$c_i = \sqrt{\frac{g_i^{(t)}}{g_i^{(s)}}} \tag{6}$$

where $g_i^{(t)}$ and $g_i^{(s)}$ are gains of the i-th half phoneme, from original speech signal and synthesized speech signal, respectively.

To compute the gain modification factor for all frames, linear interpolation is used. The c_i in (6) is regarded as the gain modification factor for the center frame of the i-th half phoneme.
The limitation of this simple interpolation is that frames close to the phoneme boundary are influenced by the neighboring gain modification factor. For example, when a large energy, voiced phone with a small gain modification factor is followed by a small energy, unvoiced phone with a large gain modification factor, the first few frames in the voiced phone are over-amplified. To cope with this problem, interpolation should be adaptively performed according to the continuity of gain contour around the phone joining point. If a dip is found in the gain contour around the joining point, no interpolation is performed, a constant gain modification factor is used. Otherwise, an interpolated gain modification factor is used. Accordingly, the gain modification factor for m-frame is given by

if $\max_{M_i \le m \le M_{i+1}} |\log[\hat{g}^{(s)}(m)] - \log[g^{(s)}(m)]| > g_{thres}$,

$$c(m) = \begin{cases} c_i & M_i \le m \le M_i + \frac{D_i}{2} \\ c_{i+1} & M_{i+1} - \frac{D_{i+1}}{2} \le m \le M_{i+1} \end{cases} \tag{7}$$

otherwise,

$$c(m) = \frac{c_{i+1} - c_i}{M_{i+1} - M_i}(m - M_i) + c_i \tag{8}$$

where $g^{(s)}(n)$ and $\hat{g}^{(s)}(n)$ represent gain and approximated gain from the synthesized speech signal, respectively, M_i and D_i denote center position and duration of the i-th half phoneme, respectively. The approximated gain $\hat{g}^{(s)}(n)$ is constructed by the interpolation of averaged variance, $g_i^{(s)}$ in (6). From the above equation, it can be understood that detecting a dip around the phone joining point is achieved by comparing maximum difference between gain contour and the approximated one, with a given threshold g_{thres}.
The overall bit allocation for prosody information is shown in Table 1.

Table 1: Bit allocation of the proposed TTS coder

parameters	allocated bits
f_0 (value)	8
f_0 (duration)	5
gain	5

VI. EXPERIMENTAL RESULTS

In this section, experimental results of the proposed coder are presented. The speech corpus used in our experiments consists of 15 utterances from a TTS talker[1]. In Fig. 4, we compare the original f_0 contour versus approximated contours for d_{max}^*=5 Hz and d_{max}^*=10 Hz, respectively. A more coarse representation of a given f_0 contour is found at the higher d_{max}^* value. Practically, setting a maximum allowable error d_{max}^* to 5∼6 Hz results perceptually good approximation for a female voice's f_0 contour. We also encoded 160 syllabic f_0 contours from 15 sentences and averaged the resulting bit rates. The experiments were performed for various $d_{max}^* = 1,\ 2,\ ...,\ 11$ and the results are shown in Fig 5.

[1]The speech material used in experiments was not used to construct the database in the TTS system.

The interesting point to note is that the resulting curve comes up with general rate-distortion characteristics even thought there is no explicit relationship between bit rate and d^*_{max}.
The average bit rate for each parameter is summarized in Table 2. We find that the bit rate for f_0 is much lower than that of conventional coders.
The TTS used in our experiment is based on the Festival speech synthesis system[10]. The number of units in the TTS database is about 60K. The quality of reconstructed speech from TTS is reasonable in both intelligibility and naturalness. The synthesized speech signals sounds close to original speech with the prosody information sufficiently preserved.
TTS systems suffer from unnatural quality for several reasons, including the lack of natural prosody. Other reasons for unnatural quality include discontinuites around unit joining points and the annoying quality of unvoiced regions. So, further naturalness can be obtained by sophisticated unit concatenation and improved synthesis schemes for unvoiced speech.

Table 2: Average values of bit rate for each parameter (d^*_{max} = 6Hz)

parameters	bit/s
phone	108.4
f_0	120.5
gain	99.5
Total	328.4

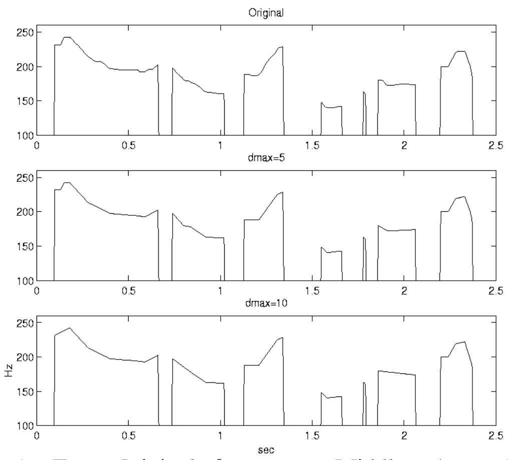

Figure 4: Top: Original f_0 contour, Middle: Approximated (d^*_{max}=5Hz), Bottom: Approximated (d^*_{max}=10Hz)

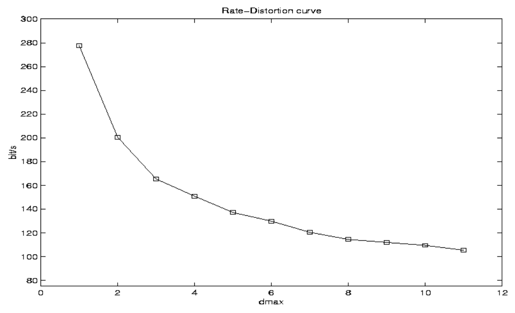

Figure 5: Rate distortion curve

VII. CONCLUSION

A 300 bit/s TTS based speech coder is proposed. The coder transmits phonetic transcription, f_0 contour and gain contour as the coded parameters. The reconstructed speech is obtained through an HNM based TTS system.
Phonetic transcription including phoneme identifications and durations is encoded by a lossless quantizer. Significant bit-reduction is accomplished by the use of contour-wise f_0 coding schemes. The proposed coding scheme minimizes bit rate given a maximum permissible error. For the gain parameter, average variance over a half-phoneme length is transmitted. Experiments with the proposed schemes showed that prosodic information is sufficiently preserved while f_0 and gain are transmitted with a high coding gain.
Future work will be focused towards constructing an automatic unit generation and multiple speaker capability.

VIII. REFERENCES

[1] J. Černocký, G. Baudoin and G. Chollet, "Segmental vocoder-going beyond the phonetic approach," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 2, pp. 605-608, 1998.

[2] C. M. Ribeiro and I. M. Trancoso, "Phonetic vocoding with speaker adaptation," in *Proc. EUROSPEECH-97*, vol. 3, pp. 1291-1294, 1997.

[3] G. Benbassat and X. Delon, "Low bit rate speech coding by concatenation of sound units and prosody coding," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, vol. 1, pp. 1.2.1-1.2.4, 1984.

[4] P. Vepyek and A. B. Bradley, "Consideration of processing strategies for very-low-rate compression of wide band speech signal with known text transcription," in *Proc. EUROSPEECH-97*, vol. 3, pp. 1279-1282, 1997.

[5] M Ismail and K Ponting, "Between recognition and synthesis-300 bits/second speech coding," in *Proc. EUROSPEECH-97*, vol. 1, pp. 441-444, 1997.

[6] H. C. Chen, C. Y. Chen, K. M. Tsou and O. T.-C. Chen, "A 0.75 Kbps speech codec using recognition and synthesis schemes," in *IEEE Workshop on Speech Coding for Telecommunications Proceedings*, pp. 27-29, 1997.

[7] Y. Stylianou, T. Dutoit and J. Schroeter, "Diphone concatenation using a Harmonic plus Noise Model of speech," in *Proc. EUROSPEECH-97*.

[8] W. B. Kleijn and J. Haagen, "Waveform interpolation for coding and synthesis," in *Speech Coding and Synthesis* (W. Kleijn and K. Paliwal, eds.), ch. 4, pp. 175-207, Elsevier, 1995.

[9] A. K. Katsaggelos, L. P. Kondi, F. W. Meier, J. Ostermann and G. M. Schuster, "MPEG-4 and rate-distortion-based shape-coding techniques," *Proceedings of the IEEE*, vol. 86, No. 6, pp. 1126-1154, June, 1998.

[10] A. J. Hunt and A. W. Black, "Concatenative speech synthesis using units selected from a large speech database," *Draft paper.*