ULTRA LOW BIT RATE SPEECH CODING USING AN ERGODIC HIDDEN MARKOV MODEL

Matthew E. Lee, Adriane Swalm Durey, Elliot Moore, and Mark Clements

Georgia Institute of Technology
Center for Signal and Image Processing
School of Electrical and Computer Engineering
Atlanta, GA 30332-0250 USA

ABSTRACT

This paper presents the framework for an ultra low bit rate speech vocoder. The system is based on a recognition-synthesis paradigm in which a single ergodic hidden Markov model (EHMM) is used to capture the statistical characterizations of speech in a flexible manner capable of limiting the effects of recognition errors. Because predetermined speech units are not used, this system has the advantage of not requiring a transcription for the training data set. By incorporating a mixed excitation scheme based on an improved MELP formulation into the EHMM, additional gains in quality and speaker characterization are achieved at no cost to the bit rate.

1. INTRODUCTION

Methods for speech coding have been shown to be capable of achieving bit rates under 3000 bps while maintaining acceptable levels of intelligibility and quality. These methods typically rely on acoustic modelling techniques to parameterize the characteristics of the vocal tract, glottal source, and prosodic features. However, in order to compress transmission bit rates to under 800 bps, it is necessary to adopt a strategy based on speech recognition and synthesis methods. Several recognition/synthesis vocoders have been proposed [1, 2] which employ techniques that recognize and segment speech signals so that only a sequence of speech units (i.e., monophones, triphones) are transmitted along with prosodic information. This class of methods are often termed *phonetic vocoders*. The decoding process in these vocoders is typically performed using speech synthesis algorithms to reconstruct the output speech from the sequence of transmitted unit indices. One disadvantage of phonetic vocoders lies in their requirement of a fully transcribed training corpus. In many cases, databases such as these must be manually produced, especially when designing speaker-specific or language-specific systems.

This work presents a framework for a low bit rate speech coder based on recognition-synthesis techniques in which segmental units such as phonemes are not predetermined but instead are automatically identified by an ergodic hidden Markov model (EHMM). Statistical characterizations that incorporate transitions as well as steady-state characteristics can be extracted using a large, ergodic HMM. This type of model has the advantage of not requiring labelled transcriptions of any data set used for training, as well as possessing greater robustness to individual errors in recognition. At the decoding stage, synthesis is performed by deriving spectral parameters from the HMM and the transmitted state sequence. An increased level of naturalness is obtained by integrating an improved mixed excitation procedure based on the Mixed Excitation Linear Prediction (MELP) Military Standard for speech coding [3]. By incorporating the bandpass voicing strengths into the feature vector of the HMM, a mixed excitation can be realized without increasing the bit rate.

In the following section, a brief overview of ergodic hidden Markov models is presented. Section 3 provides a detailed description of the training, coding, and decoding procedures. A discussion concerning the performance and achievable bit rates of the coding procedure is given in Section 4, and conclusions are discussed in Section 5.

2. ERGODIC HIDDEN MARKOV MODELS

In typical phoneme recognition applications, a left-right HMM is used in which individual monophones or triphones are modelled with a single HMM. Left-to-right HMMs permit transitions between states to occur only from left to right as time progresses. Often, with left-right models, additional constraints are placed on the state transitions. These include allowable skipping of states. Figure 1(a) illustrates an example of a left-to-right HMM with no skips allowed. In this example, states 1 and 5 are the beginning and ending states. States 2, 3, and 4 permit transitions either back to the current state or to the next sequential state. Phoneme recognition with this configuration typically involves using a Viterbi algorithm for identifying the most likely sequence of underlying models. While this class of models has been shown to be somewhat effective in phoneme recognition tasks, even small error rates can affect intelligibility measures in coding applications. There are a number of problems inherent in phonetic recognition systems including the wide range of allophones possible for a given phoneme and the effects of coarticulation.

In the present work, a single ergodic HMM is used to model the speech processes. An EHMM is characterized by the existence of possible transitions between any two states. Figure 1(b) provides an example of this with a 4-state EHMM. For an effective speech coder, the size of the EHMM must be considerably larger. Previous work [4–6] has shown that at least 64 states are required to represent all the acoustic variations present in fluent North American English. A single EHMM of this size can then be trained to model all of the sounds in such speech as well as

This work was sponsored by the Defense Advanced Research Projects Agency under Contract N00024-02-C-6339. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

the time dependent transitions from one sound to another. Furthermore, due to the flexibility of the model, as well as the absence of a language-dependent phonetic dictionary, the EHMM is capable of providing a model for any language, given sufficient training data.

One important aspect of using an ergodic configuration for coding purposes concerns the presence of recognition errors. Because the EHMM does not rely on set acoustic units such as phonemes, it is possible for recognition errors to be limited to single state-sized segments. Additionally, state errors are typically between states that are acoustically similar, which can minimize their effect on intelligibility.

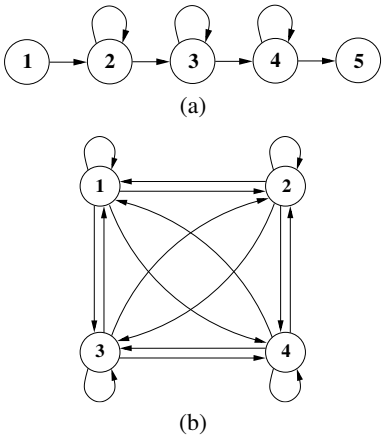

Fig. 1. Illustration of two configurations of HMMs. (a) 5-state left-right model. (b) 4-state ergodic model.

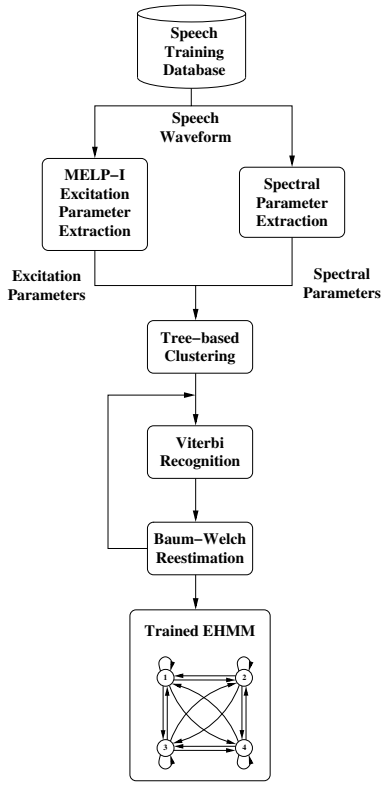

Fig. 2. EHMM training procedure.

3. VOCODER FRAMEWORK

3.1. Training

One of the most attractive features of the EHMM vocoder is the ability to train the model from a set of unlabelled training data. This characteristic enables this type of coder to be trained on a new speaker in a rapid and automatic fashion. A block diagram of the training procedure is shown in Figure 2.

The first step in the training process is to extract the spectral and excitation parameters from the training data. These parameters are combined on a frame-by-frame basis to form the feature vectors which are fed into the EHMM training algorithm. In our implementation, the feature vectors consist of two separate streams. Mel-cepstral coefficients (24th order) as well as their delta and delta-delta coefficients serve as the spectral parameters. The excitation parameters compose the second stream and consist of four bandpass voicing strengths along with their delta and delta-delta values. The spectral and excitation parameters are calculated from training data using the procedure described in the following section. The training data is sampled at 8 kHz and windowed with a 25 ms Blackman window. Each state in the EHMM is modelled by a single Gaussian distribution.

The EHMM training stage begins with an initialization procedure followed by an iterative reestimation process which is used to generate the final model probabilities. The initialization procedure uses a tree-based clustering algorithm for determining an initial set of Gaussian distribution functions which best represents the statistics of the training data.

Once the clustering algorithm converges on a set of representative distributions, an iterative procedure for reestimating the model parameters is implemented. This procedure alternates the Viterbi algorithm for estimating the sequence of model states that best represents each sentence in the training database with the Baum-Welch algorithm for reestimating the model distributions. After every iteration besides the final one, the transition probability matrix is replaced with a randomized transition matrix in order to give a starting point for the model training that is irregular. This irregular matrix helps the model states to be more easily distinguished from each other in the early stages of the model training procedure. The random transition matrix is generated by selecting a mean value of 0.8 for the diagonal elements and then randomly generating the remaining elements such that the rows sum to unity. The mean probability of 0.8 was chosen for the initial self-loop probabilities in order to generate a model that is more likely to stay in one state for a reasonably long period of time before transitioning to a different state.

3.2. Coding

The coding procedure for the vocoder, illustrated in Figure 3, processes an 8 kHz speech waveform and transmits a frame-by-frame sequence of states and a pitch estimate. Although the current state is transmitted every frame, large throughput savings can be

achieved by taking advantage of the two-state entropy of the system as dictated by the transition probability matrix obtained in the training procedure. Section 4 provides further details concerning the entropy calculations and achievable bit rates.

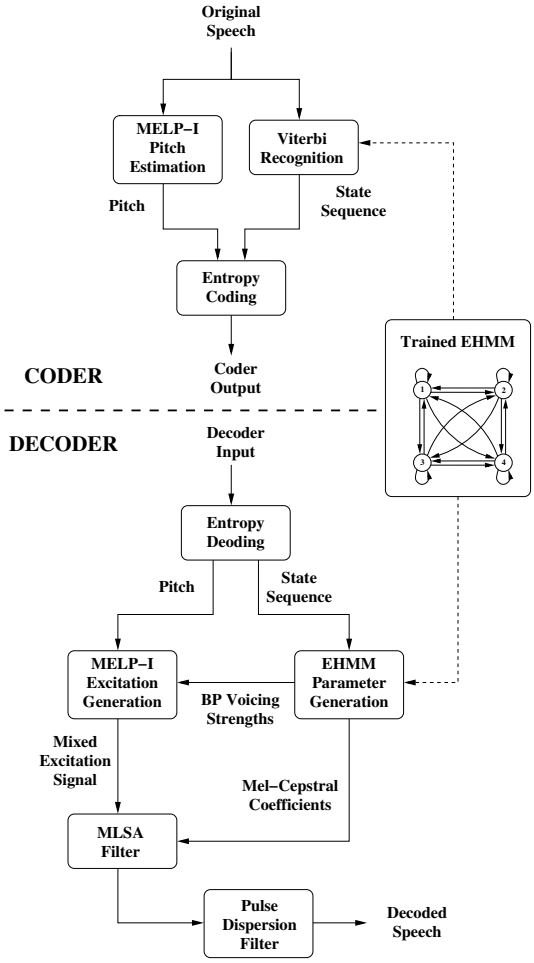

Fig. 3. Coding and decoding procedures.

The coding procedure contains two separate processes for encoding separate parameter streams. A Viterbi recognition algorithm is used to determine the underlying model state sequence with the highest probability given the trained EHMM and the observed features. This recognition algorithm is similar to the one employed by the training procedure during the iterative reestimation process.

The second parameter stream consists of a fractional pitch estimate extracted from the signal using a single stage, subframe-based method implemented as part of the improved MELP (MELP-I) coder [3, 7]. Each analysis frame is broken into subframes, and the optimal integer pitch track that maximizes the normalized correlation coefficient in each subframe and across the full frame is selected. The pitch track is then refined into a single fractional pitch estimate for the full frame. Pitch is estimated from both the input speech signal and its residual, which is obtained by applying the inverse linear prediction filter to the speech. To reduce errors, an extensive pitch doubling logic is applied to this pair of estimates to produce the final fractional pitch estimate for the frame. Overall voicing for the signal is determined by the associated correlation for that pitch. Mixed excitation voicing for the four upper bands is estimated using the normalized correlation coefficients of the bandpass filtered speech signal and its envelope at the estimated pitch. The standard MELP voicing bands are used; these are: 0–0.5 kHz, 0.5–1 kHz, 1–2 kHz, 2–3 kHz, and 3–4 kHz. These bandpass voicing strengths comprise the second stream used in training the EHMM.

3.3. Decoding

As shown in Figure 3, the decoder operates by first separating the received data into pitch and state sequence streams. A parameter generation technique is then used to extract the mel-cepstral coefficients and bandpass voicing strengths from the state sequence using the trained EHMM. The parameter generation algorithm identifies the mel-cepstral coefficients and bandpass voicing strengths that maximize the probability of the observation vector (including delta and delta-delta features), given the state sequence and trained EHMM. A fast algorithm for this maximization is derived in [8, 9]. By taking into account the dynamic features of the parameters, it is ensured that the decoded features are not simply the mean vectors of each state in the model.

The decoded mel-cepstral features are then used to construct a MLSA (Mel Log Spectrum Approximation) filter [10]. The MLSA filter is a stable IIR filter that provides an accurate approximation of the spectrum of a speech signal.

The mixed excitation signal is generated using several stages of the MELP-I decoder [3], as illustrated in Figure 4. It accepts pitch and bandpass voicing decisions for five frequency bands from the EHMM decoder, then sets default values for the gain, aperiodic flag, and Fourier magnitudes based on the frame voicing. A set of pitch epoch locations is estimated by interpolation of the pitch period across the frame; the remaining parameters are also interpolated at the pitch epoch locations. Synthesis is then performed one pitch cycle at a time.

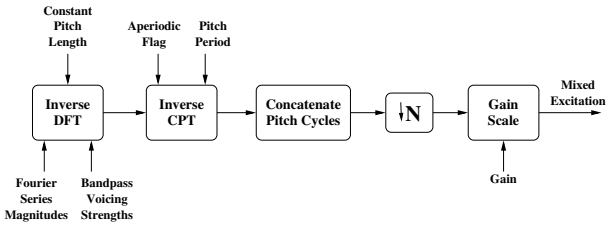

Fig. 4. MELP-I mixed excitation procedure.

The mixed excitation is constructed as follows. The Fourier magnitudes are set as the first ten magnitudes of the Fourier series of the residual. The phase of each Fourier series term is then set such that either a noise source or a periodic signal is emulated for that frequency, as indicated by the bandpass voicing strengths. Next, an inverse DFT is applied to generate a constant length pitch cycle in the constant pitch transform (CPT) domain. The inverse CPT of that signal is then taken, creating one synthesized pitch cycle at ten times the desired sampling rate; this allows fractional pitch periods to be accurately represented. These steps are iterated to synthesize each pitch cycle in the current frame. The consecu-

tive pitch periods are concatenated, then the signal is downsampled to the original 8 kHz sampling rate and gain scaled.

Once the mixed excitation generation procedure has been completed, the MLSA and pulse dispersion filters are applied to produce the final decoded speech. The pulse dispersion filter is a fixed FIR filter which introduces time-domain spread to the synthetic speech signal in order to increase the level of naturalness.

4. DISCUSSION

4.1. Experiments

A database of 1100 sentences [11] spoken by a male speaker in North American English was used to test the performance of the system in a speaker-dependent coding scenario. The training procedure described in Section 3.1 was administered using this data set to train a set of EHMMs with various configurations. The model parameters that were varied include the number of states (64, 128, 256, 512) and frame shift lengths (5 ms, 10 ms, 15 ms, 20 ms). A separate set of utterances by the same speaker were then processed with the coding and decoding algorithms. Based on these experiments, it was observed that a 128-state EHMM offered higher levels of quality and intelligibility over a 64-state EHMM, but further gains were smaller with larger models. Similarly, it was observed that 15 ms frame shifts presented a good combination of quality and compactness. Although no formal testing has been performed, the overall quality and intelligibility are good with speaker characteristics well preserved.

4.2. Entropy and bit rate

The two-state entropy of an HMM can be calculated as [12]:

$$H = -\sum_{i=1}^{N}\sum_{j=1}^{N} \pi_i a_{ij} \log_2(a_{ij}), \tag{1}$$

where $a_{ij} = P(q_t = S_j | q_{t-1} = S_i)$ given a recognized state sequence $\mathbf{Q} = \{q_1, q_2, ..., q_T\}$ and a finite state alphabet $\mathbf{S} = \{s_1, s_2, ..., s_N\}$. The term π_i is the steady state distribution of the states which can be obtained from the transition matrix, $\mathbf{A}$, by solving the linear system of equations:

$$\begin{cases} \sum_{i=1}^{N}(a_{ij} - \delta_{ij})\pi_i = 0, \quad j = 1, ..., N-1 \\ \sum_{i=1}^{N} \pi_i = 1 \end{cases} \tag{2}$$

The entropy represents the average number of bits necessary to encode the state sequence. Table 1 shows the calculated entropy values and associated bit rates for transmitting the state sequence using the 128-state EHMMs trained with various frame shift lengths. While 20 ms frame shifts enable state sequences to be transmitted with an average of 128 bps, decoded speech signals often blur transition regions or sounds with short durations. These degradations, however, are mitigated with frame shifts of 15 ms or less.

Frame Shift	Entropy (bits/frame)	Bit Rate (bits/sec)
5 ms	1.615	323
10 ms	2.104	210
15 ms	2.372	158
20 ms	2.555	128

Table 1. Entropy values and associated bit rates for a 128-state EHMM based on frame shift lengths.

5. CONCLUSION

In this work, a new framework is presented for an ultra low bit rate speech coder using HMM-based speech recognition and synthesis algorithms. By combining a flexible HMM structure that is not restricted by predetermined speech units with a proven mixed excitation model, natural-sounding speech coding at bit rates of under 300 bps can be achieved. Additionally, because of the flexibility of the model, the coder can be efficiently trained on unlabelled databases. This enables a coder based on this framework to be quickly trained on new speakers or new languages. Although many improvements for this technique are obvious and are under investigation, the relatively simple model described performs surprisingly well.

6. REFERENCES

[1] C. M. Ribeiro, I. M. Trancoso, and D. A. Caseiro, "Phonetic vocoder assessment," in *ICSLP*, 2000, pp. 830–833.

[2] R. da S. Maia, R. J. da R. Cirigliano, D. Rojtenberg, and F. G. V. Resende Jr., "Mixed-excited phonetic vocoding at 265 bps," in *ICASSP*, 2003, pp. 796–799.

[3] A. E. Ertan, *Pitch-Synchronous Processing of Speech Signal for Improving the Quality of Low Bit Rate Speech Coders*, Ph.D. dissertation, Georgia Institute of Technology, 2003.

[4] D. J. Pepper, *Large Hidden Markov Model State Interpretation as Applied to Automatic Phonetic Segmentation and Labeling*, Ph.D. dissertation, Georgia Institute of Technology, 1990.

[5] E. Farges and M. Clements, "Hidden Markov models applied to very low bit rate coding," in *ICASSP*, 1986, pp. 433–436.

[6] D. Pepper and M. Clements, "On the phonetic structure of a large hidden Markov model," in *ICASSP*, 1991, pp. 465–468.

[7] A. McCree and J. Carlos de Martin, "A 1.7 kb/s MELP coder for with improved analysis and quantization," in *ICASSP*, 1998, pp. 593–596.

[8] K. Tokuda, T. Kobayashi, and S. Imai, "Speech parameter generation from HMM using dynamic features," in *ICASSP*, 1995, pp. 660–663.

[9] K. Tokuda, T. Masuko, T. Yamada, and T. Kobayashi, "An algorithm for speech parameter generation from continuous mixture HMMs with dynamic features," in *EUROSPEECH*, 1995, pp. 757–760.

[10] T. Fukada, K. Tokuda, T. Kobayashi, and S. Imai, "An adaptive algorithm for mel-cepstral analysis of speech," in *ICASSP*, 1992, pp. 137–140.

[11] J. Kominek and A. W. Black, "CMU ARCTIC databases for speech synthesis," Tech. Rep. CMU-LTI-03-177, Carnegie Mellon University, 2003.

[12] S. Roucos, J. Makhoul, and R. Schwartz, "A variable-order Markov chain for coding of speech spectra," in *ICASSP*, 1982, pp. 582–585.