A SCALABLE PHONETIC VOCODER FRAMEWORK USING JOINT PREDICTIVE VECTOR QUANTIZATION OF MELP PARAMETERS

Alan McCree

MIT Lincoln Laboratory
Lexington, MA 02420
E-mail: mccree@ll.mit.edu

ABSTRACT

We present the framework for a Scalable Phonetic Vocoder (SPV) capable of operating at bit rates from 300 - 1100 bps. The underlying system uses an HMM-based phonetic speech recognizer to estimate the parameters for MELP speech synthesis. We extend this baseline technique in three ways. First, we introduce the concept of predictive time evolution to generate a smoother path for the synthesizer parameters, and show that it improves speech quality. Then, since the output speech from the phonetic vocoder is still limited by such low bit rates, we propose a scalable system where the accuracy of the MELP parameters is increased by vector quantizing the error signal between the true and phonetic-estimated MELP parameters. Finally, we apply an extremely flexible technique for exploiting correlations in these parameters over time, which we call Joint Predictive Vector Quantization (JPVQ). We show that significant quality improvement can be attained by adding as few as 400 bps to the baseline phonetic vocoder using JPVQ. The resulting SPV system provides a flexible platform for adjusting the phonetic vocoder bit rate and speech quality.

1. INTRODUCTION

The phonetic vocoder [1] is an attractive approach for speech coding at very low bit rates. In this system, the information content of the speech signal is extracted with a phonetic speech recognizer, and the prosody of the particular utterance is encoded with a separate scheme such as pitch contour quantization. Conceptually this approach reduces the speech waveform down to its lowest information content: the particular phrase being spoken and the speaking style of the utterance. This has led to reasonable speech quality at bit rates around 300 bps [1, 2, 3, 4].

However, the speech quality of phonetic vocoders is inherently limited by factors such as speech recognition errors, acoustic background noise, and speaker variation. This paper presents the framework for a *scalable* phonetic vocoder, capable of operating at bit rates from 300 - 1100 bps with continually increasing quality.

This work was sponsored by the Defense Advanced Research Projects Agency under Air Force Contract F19628-00-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States Government.

2. BASELINE PHONETIC VOCODER

Our system combines a phonetic speech recognition algorithm with a speech synthesizer, as shown in Figure 1. The recognizer uses Hidden Markov Models (HMM's), with each of 39 monophones characterized by five states. For each phone state, the feature vector distribution is modelled by a Gaussian Mixture Model (GMM). Feature vectors represent the 16 kHz-sampled speech signal every 10 ms using mel cepstra, deltas, and delta-deltas. The speech recognition models were developed from the training partition of the TIMIT [1] corpus.

The speech synthesizer in this system uses the Mixed Excitation Linear Prediction (MELP) model [5]. This model is based on the traditional LPC vocoder with either a periodic impulse train or white noise exciting an all-pole filter, but contains four additional features: mixed pulse and noise excitation, periodic or aperiodic pulses, adaptive spectral enhancement, and pulse dispersion filter.

In our phonetic vocoder, all MELP parameters except pitch are estimated from the speech recognition output. Every 10 ms, the LPC coefficients, gain, and voicing mixture are updated based on the estimated phone state and then interpolated for synthesis. Since the recognition model parameters are in the form of cepstra and deltas, we separately train the mean MELP parameters for each phone state estimated by the recognizer over the training data (TIMIT). To make parameterization easier for clustering, the five bandpass voicing decisions are converted to a single voicing cutoff frequency assuming a lowpass/highpass mixing structure. This is accomplished by analyzing the number of voiced bands N_v, assuming that the lowest N_v bands are actually voiced, and using the cutoff frequency of the highest voiced band.

We estimate that the total bit rate of this phonetic vocoder is about 300 bps, as shown in Table 1. As with much previous work on phonetic vocoders, we have chosen to focus our initial experiments on the most important issues of coding the spectrum, gain, and voicing. Therefore, we have left implementation of the state path and pitch contour quantization for future work; our bit-rate estimates for these are based on [3] and [6], respectively.

We have evaluated the performance of this baseline phonetic vocoder using informal listening with speech material

[1] Available from http://www.ldc.upenn.edu

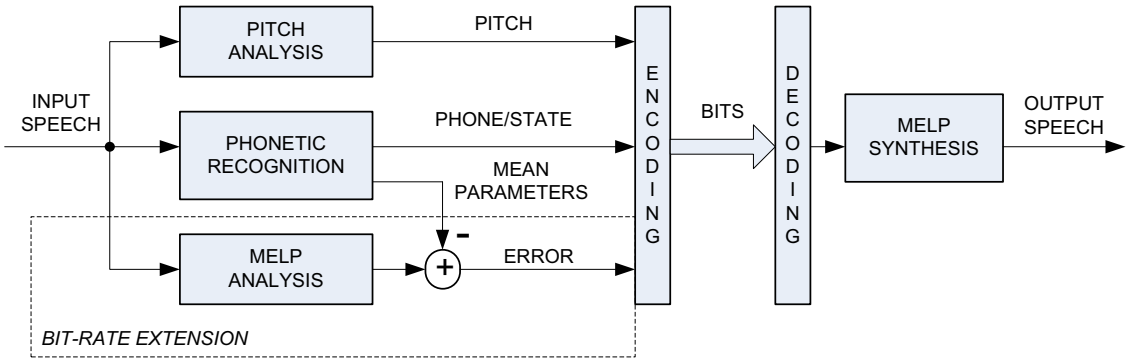

Fig. 1. Scalable Phonetic Vocoder

Parameter	Bits/phone	Bit Rate
Phone	5	50
State path	10	100
Pitch contour	15	150
Total	30	300

Table 1. Estimated baseline phonetic vocoder bit allocation with mean phone duration of 100 ms.

from the testing partition of TIMIT. At best the intelligibility and quality of this system can be described as "marginal". Surprisingly, even though the coder is speaker-independent, some character of each individual speaker is maintained. Since there is only one set of MELP parameters for each phone state over all speakers, presumably this comes from the prosody information (timing and pitch).

3. PREDICTIVE TIME EVOLUTION

A problem with the phonetic vocoder speech output is a lack of smoothness. Since the MELP parameters change relatively abruptly from phone to phone, the speech sometimes sounds "jumpy". One solution to this problem is to include delta cepstral information (both forward and backward) in the synthesis [3], but for our application a left-to-right evolution in the MELP parameter space would be preferrable.

We introduce this smoothing with a predictive time evolution. For each speech frame, the new MELP parameter vector is generated with a combination of the past frame vector and the new mean value for the phone state; this is the equivalent of 0-bit predictive vector quantization. We assume a diagonal prediction matrix, so that each parameter has its own prediction coefficient. For each phone state, we now train a codevector and a prediction coefficient vector. In informal listening, this predictive model provides a noticeable improvement in quality without increasing the bit rate of the phonetic vocoder. However, the speech quality is still quite low, so we would like to find a way to increase the performance of the phonetic vocoder even if this requires an increase in bit rate.

4. SCALABLE PHONETIC VOCODER

The principle behind our Scalable Phonetic Vocoder (SPV) system is to encode the error between the mean MELP parameters for the current phone state and the true parameters analyzed for the speech frame. Coding this parameter error signal at low bit rates using vector quantization (VQ) improves quality while staying within the phonetic vocoder framework; as the bit rate increases the system becomes a full MELP vocoder. Note that this can be interpreted as a tree-structured vector quantization, where the index of the first stage of the tree is the speech recognition estimate of the phone state.

4.1. MELP Parameter Vector Quantization

To fully exploit the relationships between MELP parameters within a frame, we quantize them jointly in one supervector. This 12-dimensional vector consists of 10 Line Spectral Frequencies (LSF's), the frame gain in dB, and the voicing cutoff frequency. Since these parameters have different units, we use a weighted Euclidean distance for quantization composed as follows. First, the LSF's are weighted as in [7]. Then, low-bit rate quantizers are designed for the LSF's, gain, and cutoff frequency. Finally, a composite weighting function is generated by weighting each of these parameters by the inverse of its individual codebook quantization errors.

To exploit the correlation between MELP parameters over successive time frames, we use first-order predictive vector quantization. It is well known that this approach can provide significant performance improvement by quantizing the difference between the current input vector and a predictive estimate from the previous quantized vector. Since our 10 ms MELP analysis frames are significantly overlapping, there is strong redundancy between frames resulting in a high prediction gain. In some ways this approach is similar to oversampled differential waveform coders such as CVSD.

4.2. Joint Predictive Vector Quantization

To maximize the flexibility of our predictive VQ system, we use a structure that we call Joint Predictive Vector Quanti-

P1	C1
	C2
	C3
	C4
	C5
	C6
	C7
	C8

Table 2. Predictive VQ.

P1	C1
	C2
	C3
	C4
P2	C1
	C2
	C3
	C4

Table 3. Switched Predictive VQ, one codebook.

P1	C11
	C12
	C13
	C14
P2	C21
	C22
	C23
	C24

Table 4. Switched Predictive VQ, two codebooks.

P1	C1
P2	C2
P3	C3
P4	C4
P5	C5
P6	C6
P7	C7
P8	C8

Table 5. Joint Predictive VQ.

zation (JPVQ). This is essentially an application of the technique originally proposed for image coding in [8] to the problem of speech spectral quantization. To illustrate this approach, we discuss the predictor and codebook structure of a number of common predictive VQ methods, each requiring three bits per vector. First, Table 2 shows that a predictive VQ system has one predictor and eight codevectors. For speech coding, it can be very helpful to have strong and weak predictors for steady-state and transitional regions, respectively. Therefore, a conventional switched predictive VQ uses two predictors, each sharing a 4-level codebook, as in Table 3. The method presented in [9], illustrated in Table 4, uses different codebooks for each predictor to allow better codebook design at the same bit rate, at the price of additional storage. Moving past the switched predictive approach, we can use a separate predictor for each codevector, as shown in Table 5. In this Joint Predictive VQ method, each codeword represents a predictor/codevector pair, so that prediction and quantization are jointly optimized. Note that all the previous cases of predictive VQ can be viewed as constrained special cases of JPVQ, where the predictor is "pooled" across multiple codevectors. While this reduces codebook storage and search complexity somewhat, it also reduces quantization performance by limiting the available range of predictor/codevector pairs as compared to the full unconstrained JPVQ.

4.3. Training Algorithm

We have found that a joint training optimization is essential for the performance of JPVQ. Here we derive the formulas for optimal predictor and codevector update using scalar notation and unweighted Euclidean distance for simplicity. Since our prediction and weighting matrices are diagonal, the extension of the algorithm to the vector case with weighted distance is straightforward.

We optimize the squared error over all i training vectors in a given cluster, $E = \sum_i (y_i - \hat{y}_i)^2$, with each reconstructed vector given by $\hat{y}_i = a\hat{x}_i + c$, where $\hat{x}_i$ is the previous quantized vector, a is the prediction coefficient for this cluster, and c is the corresponding codevector. Taking the derivative of E, we find that the optimal a given c is

$$a_{opt}|c = \frac{\sum_i \hat{x}_i y_i - c\sum_i \hat{x}_i}{\sum_i \hat{x}_i^2} \quad (1)$$

$$= \frac{\sum_i \hat{x}_i (y_i - c)}{\sum_i \hat{x}_i^2}. \quad (2)$$

Similarly, the optimal codevector c given the current predictor coefficient a is

$$c_{opt}|a = \sum_i y_i - a\sum_i \hat{x}_i \quad (3)$$

$$= \sum_i (y_i - a\hat{x}_i). \quad (4)$$

An iterative sequential approach would be to alternate each of these updates across iterations; however, we find it simple and effective to update both at once with a stepsize $\mu = 0.5$. While either form of update equations can be used, implementation is simpler using Equations 2 and 4.

We note that a full joint design can be achieved in theory, since we can combine the above two equations to get

$$c_{opt} = \frac{\sum_i y_i - \sum_i \hat{x}_i \frac{\sum_i \hat{x}_i y_i}{\sum_i \hat{x}_i^2}}{1 - \frac{(\sum_i \hat{x}_i)^2}{\sum_i \hat{x}_i^2}}.$$

This value for c can then be used to compute a_{opt} in Equation 1. However, this training algorithm is more complex than the one mentioned previously, which we have found to converge quickly and reliably in our application.

Coder	Bit Rate	Predicted MOS
mean-only PV	300	2.01
predictive PV	300	2.23
4-bit SPV	700	2.80
8-bit SPV	1100	3.04
LPC10e	2400	2.64
MELPe	1200	2.89
MELP	2400	3.06

Table 6. Estimated bit rate and predicted MOS scores for phonetic vocoder variations and reference coders.

5. PERFORMANCE

We have evaluated multiple versions of this Scalable Phonetic Vocoder, along with the reference coders LPC-10e, MELP, and MELPe, using informal listening on files from the TIMIT test partition (not used in training). We tested the following SPV variations:

1. The baseline phonetic vocoder using mean MELP parameters for each phone state. Pitch and state information are unquantized; estimated total bit rate is 300 bps.
2. System (1) using mean and prediction coefficients.
3. System (2) with additional 4-bit/frame JPVQ (700 bps).
4. System (2) with additional 8-bit/frame JPVQ (1100 bps).

From this listening we have reached a number of conclusions. First, a speaker-independent phonetic vocoder can provide modest intelligibility, quality, and speaker recognizability (from prosody only). Second, the modelling of time evolution by predictive estimation provides smoother, higher-quality output speech. Finally, even very small (16 level) parameter codebooks provide significant improvement in all three areas. We find this improvement remarkable given that only four bits per frame are encoding a 12-dimensional parameter vector.

We also generated the predicted MOS score, shown in Table 6, for each system over 96 files using the ITU PESQ objective measure. While this measure is not calibrated for low rate vocoders, these results are consistent with our informal quality assessments.

6. CONCLUSION

We have developed the framework for a Scalable Phonetic Vocoder with bit rates from 300 - 1100 bps. We showed the benefit of using predictive time evolution to allow the MELP synthesizer parameters to move smoothly from phone to phone, and extended this approach with Joint Predictive Vector Quantization of the MELP parameters. Based on these initial encouraging results, we plan to develop and test a fully-quantized SPV, and then extend it using non-acoustic sensor information to increase robustness in noise [10].

7. ACKNOWLEDGEMENTS

The author gratefully acknowledges many interesting technical discussions with colleagues Kevin Brady, Tom Quatieri, Carl Quillen, Joe Campbell, and Cliff Weinstein. Thanks especially to Carl for providing his speech recognition software.

8. REFERENCES

[1] J. Picone and G. Doddington, "A Phonetic Vocoder," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 1989, pp. 580–583.

[2] C. Ribeiro and I. Trancoso, "Phonetic Vocoding with Speaker Adaptation," in *Proc. EUROSPEECH '97*, 1997, pp. 1291–1294.

[3] T. Masuko, K. Tokuda, and T. Kobayashi, "A Very Low Bit Rate Speech Coder using HMM with Speaker Adaptation," in *Proc. ICSLP '98*, 1998.

[4] R. S. Maia, R. J. R. Cirigliano, D. Rojtenberg, and F. G. V. Resende Jr., "Mixed-Excited Phonetic Vocoding at 265 bps," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 2003, pp. I796–I799.

[5] A. McCree and T. P. Barnwell III, "A Mixed Excitation LPC Vocoder Model for Low Bit Rate Speech Coding," *IEEE Trans. Speech and Audio Processing*, vol. 3, no. 4, pp. 242–250, July 1995.

[6] K. Lee and R. Cox, "A Very Low Bit Rate Speech Coder Based on a Recognition/Synthesis Paradigm," *IEEE Trans. Speech and Audio Processing*, vol. 9, no. 5, pp. 482–491, July 2001.

[7] K. K. Paliwal and B. S. Atal, "Efficient Vector Quantization of LPC Parameters at 24 Bits/Frame," *IEEE Trans. Speech and Audio Processing*, vol. 1, no. 1, pp. 3–14, Jan. 1993.

[8] K. M. Holt and D. L. Neuhoff, "Coding by Selective Linear Prediction: a New Scheme for Predictive Vector Quantization," in *Proc. IEEE Int. Conf. Image Processing*, 2002, pp. II657–II660.

[9] A. McCree and J. C. DeMartin, "A 1.7 kb/s MELP Coder with Improved Analysis and Quantization," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Seattle, 1998, pp. II 593–596.

[10] K. Brady, T. F. Quatieri, J. P. Campbell, W. M. Campbell, M. S. Brandstein, and C. J. Weinstein, "Multisensor MELPe Using Parameter Substitution," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Montreal, 2004, pp. I477–480.