Enhanced Speech Coding Based on Phonetic Class Segmentation

Adriane Swalm Durey, Venkatesh Krishnan, and Thomas P. Barnwell III

Center for Signal and Image Processing
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, Georgia 30332
United States of America

asdurey@ieee.org, {vkrish,barnwell}@ece.gatech.edu

Abstract

Given a baseline speech coder and speech with an available phonetic class segmentation, a number of potential enhancements to that coder become possible. While the quality of speech segmentation by phoneme and phonetic class is constantly improving, we use TIMIT to generate phonetic class segmentation as a basis for initial testing of these techniques. Using coders drawn from the MELP family, we explore specialized phonetic codebooks, phonetically-driven superframing, and improved modeling of specific phonetic classes and the transitions between them. We compare the reconstructed speech from these enhancements against the base coder using the metrics of computational cost, transmission cost, and the quality of the reconstructed speech. In most cases, we find that segmentation-based coders can produce speech with quality comparable to that of MELP, using fewer transmitted bits and at no additional computational cost. With phonetic codebooks and transition modeling, CCR tests show these segmentation-based coders produce speech of better quality than is produced by MELP.

1. Introduction

Speech coding allows us to transmit the rich information present in an speech signal at a relatively small cost. Direct sampling of a speech waveform can run from 8 kHz for toll quality speech up to 44.1 kHz for CD quality audio. The transmission of such high bit rates is impractical, leading us to seek alternative methods by which to code specific classes of audio, such as speech.

In model-based speech coding, a model for speech is developed, parameters relevant to that model are extracted from the audio waveform, only those parameters are encoded and transmitted, then the speech is reconstructed at the decoder based only upon the parameters and the model. Many successful speech coders transmit significantly fewer bits than waveform coding at an 8 kHz sampling rate, but produce audio of similar quality. Code excited linear prediction (CELP) [1] coders produce toll quality audio at 4800 bps by transmitting codes for pitch, linear prediction coefficients (LPCs), and a residual selected from a codebook. Mixed excitation linear prediction (MELP) [2] coders produce better quality audio than CELP at only 2400 bps; its extension MELPe [3] also includes a 1200 bps mode. These coders also rely on an LPC model of speech, but transmit additional data describing a multi-band excitation. As low in transmission rate as these coders are, we would like to reduce these numbers further; as good as the quality of these coders is, they suffer some deficiencies that should be addressed.

One shortcoming of such coders is that, while they are designed to process speech, they do not use information about language extensively. Most coders distinguish only between voiced and unvoiced speech. Speech is generalized using an LPC model of the vocal tract and some combination of random and periodic excitation similar to that produced by the vocal system, but the characteristics of the language being spoken are seldom exploited.

One reason for this is that the speech characteristics that have been modeled in speech coders are easily generalizable. The coder does not need to identify what is being said to accurately determine whether a frame is primarily voiced or unvoiced. To take advantage of details about how language effects the speech waveform, it is necessary to know more details about what is being said, that is, to segment and classify the spoken content.

It is non-trivial to automatically generate a transcription of a speech signal which marks the phonemes spoken and their beginning and ending boundaries. We propose using a more general process—to go from the speech signal to a higher level phonetic class transcription. This will allow us to combine similar classes of sounds at a level above the actual spoken content, which allows more detailed language modeling than most speech coders have employed. We observe that, given a baseline coder and speech which has been segmented by phonetic class, a number of potential enhancements in the coder in terms of coding cost and quality of the reconstructed speech become possible. Using TIMIT and base coders drawn from the MELP family, we provide proof-of-concept tests for several such enhancements.

2. Framework

The availability of a phonetic class segmentation for a speech waveform enables us to make several enhancements to standard speech coders. The acquisition of phonetic class segmentation from speech is addressed in the next section. The base speech coders that we use to test these techniques are described in Section 2.2. Section 2.3 provides details about the test metrics employed here.

2.1. Phonetic class segmentation

For the purposes of this research, we require only phonetic class segmentation. This requires identification of both the phonetic class and its beginning and ending times in the waveform. It does not require lower-level phoneme segmentation (i.e., the actual spoken content). The generalizations that can be made about the member phonemes (for example, /f/, /s/, /S/, and /T/) of a phonetic class (unvoiced fricatives) allow us to extract substantial savings from the parameter sets transmitted by a speech coder while avoiding a more difficult segmentation problem.

In the future, it is likely that we will be able to automate phonetic class segmentation of speech (and auxiliary) data [4] [5]. For feasibility testing, we have used phonetic class data extracted from the phoneme-level labels provided in the TIMIT database. We compact those labels into ten phonetic classes, including: voiced and unvoiced fricatives, voiced and unvoiced plosives, affricates, vowels, nasals, liquids, glides, and silence. We provide this information to the coder in one of two ways. In the first case, we make a frame-level decision about the phonetic class by selecting the dominant class across the frame. In the second, we provide a sample-level phonetic class determination aligned to the audio stream at 8 kHz. TIMIT speech data is sampled at 16 kHz; it has been resampled to 8 kHz for coding, except where otherwise noted.

2.2. Speech coder

These segmentation-based coding enhancements were implemented on the platform of two coders in the MELP family. The first of these is the standard MELP implementation at 2400 bps [2]. This coder uses a five band mixed excitation model to represent speech, based on pitch, bandpass voicing, aperiodicity, gain, line spectral frequencies (LSFs), and Fourier magnitudes of the residual. These parameters are extracted every 180 samples from speech sampled at 8 kHz and are encoded using 54 bits per frame.

Testing was also performed on an improved version of that coder, called MELP-I [6]. This variant focuses on accurate pitch detection and pitch synchronous processing, using methods such as a circular LPC. The sampling rate, frame rate, and parameter encodings are the same as for MELP. Additionally, MELP-I has an object oriented framework [7] well suited to rapid prototyping of speech coders that is advantageous for the type of work proposed in Section 3.

2.3. Testing

For initial testing, we evaluated these phonetically modified speech coding enhancements using three metrics. The first was the computational cost of the segmentation-based technique compared to that of the base coder (MELP or MELP-I). The second was the cost of transmitting the parameters for the speech signal relative to MELP at 2400 bps. The third metric was the quality of the reconstructed audio from the enhanced coders as compared to that produced by the base coder.

Informal listening tests were conducted to detect audible artifacts and obvious degradations in quality. Formal comparison category rating (CCR) tests were conducted on the best two of these coding methods. 15 sentences were selected from TIMIT and processed by the MELP coder and by a phonetic segmentation-based coder. They were presented in random order to 15 listeners (primarily native English speakers) who rated the quality of the second sentence compared to the first on a scale from -3 (much worse) to 3 (much better). Four null comparisons were introduced to eliminate inconsistent subjects. Comparison mean opinion scores (CMOS) were produced from the aggregated scores and checked for statistical significance using a one-sided t-test.

3. Phonetic segmentation in speech coding

In this section, we will propose several speech coding enhancements based on the availability of phonetic class data. We will discuss the motivation behind these improvements, the methods used, and the results of applying them in terms of cost of computation, cost of transmission, and the quality of the reconstructed speech when compared to the base coder. Estimated bit rates include only the technique discussed in a given section.

First, we will look at constructing codebooks tied to specific phonetic classes. In Section 3.2, we will take advantage of the hysteresis present in successive frames within the same phonetic class to drive intelligent superframe aggregation. In Sections 3.3 and 3.4, we will look at alternative models for specific phonetic classes. The first of these is to extend the bandwidth of specific phonemes which suffer degradation due to the 8 kHz sampling rate used by MELP. Section 3.4 demonstrates how we may avoid the analysis of fluctuating phoneme transition regions to improve the quality of the reconstructed speech in those areas, while still reducing the bit rate.

3.1. Phonetic class-based codebooks

Speech frames with the same phonetic class exhibit considerable similarity in their MELP parameters, particularly when compared to the amount of similarity between the MELP parameters that represent a general frame of speech [8]. We can take advantage of this during speech coding when phonetic class segmentation is available by basing codebook selection on the phonetic class of the current frame of the speech signal.

The line spectral frequencies (LSFs) are the most expensive parameters to transmit in the MELP coder, requiring 25 bits per frame. By focusing the codebook on one class only, rather than requiring it to be general enough to represent any frame of speech, we achieve two things. We can greatly reduce the number of transmitted bits by reducing the size of the LSF codebooks, and we can improve the ability of the codebook to accurately model that class.

Two methods were used to create new LSF codebooks targeted to specific phonetic classes. The first was to train new vector quantization (VQ) codebooks based only on frames drawn from a single phonetic class. While expensive to train, this method is well suited to rich and varied phonetic classes, such as vowels. New codebooks were generated from samples of vowel frames drawn from TIMIT. The resolution of those codebooks was selected to meet an average log spectral distortion (SD) close to 1 dB, fewer than 1% of the frames having more than 2 dB of SD, and no frames having more than 4 dB of SD. This allowed us to reduce the 25 bit LSF codebook used in MELP to as few as 14 bits for a codebook targeted only to vowels.

The second codebook generation technique was based on the standard MELP multi-stage VQ (MSVQ) codebook. LSFs from frames in a single phonetic class were encoded using the MELP MSVQ, and the most frequently selected codewords from the first stage of the MELP MSVQ were used to build a smaller codebook for that class alone. This technique was used to build small codebooks from 16–128 words (4–7 bits) for smaller phonetic classes like unvoiced and voiced fricatives.

These techniques were tested using both the MELP and MELP-I coders. In informal listening tests, the MSVQ codebook reduction method used for the fricative classes in MELP-I did not audibly reduce the quality of the reconstructed speech at the decoder, even for codebooks as small as 16 elements. CCR testing using the retrained vowel codebook produced a CMOS of 0.1889 ± 0.1374, leading to a 95% confidence interval of

$[0.1819, 0.1959]$; this is a statistically significant improvement in quality when compared to MELP.

The retrained vowel codebook led to an estimated bit rate of 1925 bps. The use of the reduced voiced and unvoiced fricative codebooks resulted in an estimated bit rate of 2164 bps. Together, the estimated reduction in bit rate for both codebooks combined was 1775 bps. The reduction in codebook size generally more than offset the need to transmit phonetic class information to allow the decoder to select the proper codebook.

There is little difference in computational cost between MELP with its original codebooks and one using these alternate codebooks. While generating the codebooks is time-consuming and computationally expensive, it is a one time cost. During execution, the cost of searching those codebooks is generally reduced, since all of the tested codebooks were both smaller than the MELP MSVQ codebook and consisted of a single stage.

3.2. Phonetically-driven superframing

In the previous section, we looked at how we could exploit the similarity of speech frames with the same phonetic class to build specialized codebooks for that class. It is also possible to make use of within-class hysteresis in the construction of superframes from single MELP frames. When we compare superframes generated from three frame blocks, as used in MELPe [3], to blocks created from phonetic class-related frames, we find much less variance between the parameters in the phonetic blocks than those in the "randomly" aggregated blocks [8]. This is observed across all parameters, not simply in the LSFs.

This technique was tested within the MELP framework in two ways. Standard MELP parameter sets were extracted from each 180 sample frame of speech. Then, based on the phonetic class segmentation, frames were grouped into phonetically similar superframes composed of 1–3 MELP frames. Those superframes were reduced to a single MELP parameter set, which was transmitted in place of the grouped frames. Alternatively, the superframe size was determined and then a single set of MELP parameters was extracted from the frame as a whole (as a frame of 180–540 samples). At the decoder, the representative parameter set for the superframe was simply duplicated to cover the required number of frames, then synthesis proceeded as usual.

This technique reduced the bit rate of the transmitted parameters to as low as 900 bps. For all coders tested using this technique, the quality of the decoded speech was less than that of speech encoded with the 1200 bps version of MELPe. This could be improved by changing the method used to compact the parameters into the representative set. Instead of sending a single frame, we could add inter-frame information describing changes throughout the superframe as MELPe does. It is additionally noted that some phonetic classes, such as unvoiced fricatives, responded better to this technique than did more rapidly evolving classes, such as diphthongs.

In terms of computation, selecting a single frame to transmit in place of a phonetic superframe reduces the computational cost by eliminating the need for analysis for the other frames, although two additional frames of delay are introduced. The calculation of parameters from the larger, aggregated frames does increase the computational effort per superframe, while eliminating the analysis of 1–3 smaller frames.

3.3. Bandwidth extension

Many phonetic classes, like vowels, exhibit most of their energy in the range from 0–4 kHz; this is easily captured at an 8 kHz sampling rate. Others, like fricatives, exhibit most of their energy above 4 kHz [9]; this information is lost when a coder like MELP is used to transmit speech. Bandwidth extension has been used to restore such lost information above 4 kHz [10, 11]. When phonetic segmentation is available, bandwidth extension can be targeted to those areas most affected by the limitations of the selected sampling rate without damaging other regions through unneeded processing.

Fricative regions (both unvoiced and voiced) were identified in the coded audio stream using the phonetic class segmentation. Parameters in non-fricative regions were extracted as usual for MELP at the standard 8 kHz sampling rate. When a fricative region was encountered, its parameters were extracted from the signal at 16 kHz (the original sampling rate in TIMIT). The primary difference in extracted parameters is that, for a fricative, 20 LSFs were extracted from the higher rate signal. At the decoder, the residual in fricative regions was generated at 8 kHz, then upsampled to 16 kHz. It was then full wave rectified, mean subtracted, and high pass filtered to extend the bandwidth of the residual. Finally, the extended residual was filtered by the larger set of 21 LPCs obtained from the LSFs. Segments of the speech with other phonetic classes did not undergo further processing; they were synthesized by MELP then upsampled to 16 kHz.

While not providing any real bit rate reduction, this method does provide an improvement in the perceived quality of fricative regions by restoring the energy in the upper portion of the spectrum that was lost during coding. While the regions with extended bandwidth are sometimes noticeable in the context of longer speech segments, the novelty quickly fades, leaving only the perception of an improvement in those regions, and, therefore, in the speech overall. Computationally, the most expensive aspect of bandwidth extension is the need to operate at a 16 kHz sampling rate. Most additional processing comes from the larger fricative LSF set, creation of the bandwidth extended residual, and upsampling of the remaining signal. It is unlikely that there is as much value in applying this technique to other classes, except perhaps the fricative portion of affricates, which is likely to suffer the same deficiencies.

3.4. Transition modeling

The most difficult regions to analyze when extracting parameters for speech coding are transitions between one phoneme and the next. Assuming an average of twelve phonemes per second for normal speech [12] and the 44.44 analysis frames per second used in the MELP family of coders, approximately 27% of the analyzed frames will contain transitions from one phoneme to another. This means that most of the parameters being extracted by the coder during that frame are going to be non-stationary because the speaker's articulators will be motion during that time. This makes the likelihood of parameter extraction errors much higher for these frames.

Phonetic segmentation can easily be used to identify the frames during which one phonetic class transitions into another. If the transition is close to the beginning or end of a frame, then analysis can safely be focused on the phoneme that dominates the largest portion of the frame. If the amount of time spent in each phoneme is more balanced, then we must recognize that there may be no stationary region of sufficient size in which to conduct analysis during that frame.

One simple method of avoiding analysis in unstable regions is to avoid that analysis entirely. When a phonetic class transition was encountered, the transition frame was dropped (not transmitted) and a simple code was transmitted describing the action to be taken to replace the dropped frame at the decoder. This action was to copy the previous frame, copy the next frame, or interpolate between the next and previous frames. This required only two bits of information to represent the transition frame, rather than the 54 bits transmitted for a standard MELP frame. An extra frame of delay was added to the MELP processing time. All non-transition frames were transmitted using standard MELP parameter frames.

The replacement action was decided based on phonetic class transition information [9]. For transitions with sharply delineated articulation, such as entering and leaving silence regions, plosives, and nasals, the sharp transition was mimicked by copying a neighboring frame. Which frame was copied depended on the dominance of the phonetic classes in the current frame. The remaining class transitions, whose characteristics are smoother, were interpolated by default. Some parameters were given an explicit replacement behavior based on testing; pitch was always replaced by copying a neighboring frame and gain by interpolation.

This technique was tested using the MELP-I coder framework using sample-level segmentation data. Despite the lost information, it led to better results than standard MELP, because it was not doing analysis in non-stationary coarticulation regions. It only conducts analysis in the stable areas surrounding the transitions. In formal CCR testing, transition dropping led to a CMOS of 0.5819 ± 0.2720 and a 95% confidence interval of $[0.5681, 0.5957]$; this is a statistically significant improvement in quality over the base MELP coder. Dropping phonetic transitions results in approximately 12 dropped frames out of every 44.44 (in a second), or an estimated bit rate of 1776 bps. Computationally, dropping and replacing transitions is as expensive as standard MELP-I. The additional cost for the interpolation of replacement frames is offset by the quantization that is avoided for that frame.

4. Conclusions

We have shown that the combination of standard speech coding methods with phonetic class segmentation enables the implementation of a variety of enhancements to the coder. With most of the techniques proposed, we find that the modified MELP coder does not reduce the quality of the reconstructed audio when compared to MELP alone, while reducing the amount of data that must be transmitted by up to 50%. The most distorting method, simple phonetic class superframing, shows a quality similar to 1200 bps MELPe. Two of the methods, vowel codebook training and transition modeling, have been shown by CCR testing to improve the quality of coded speech over that of MELP alone. Most of these methods are no more expensive in terms of computation than the base coder, though several introduce additional delay. None requires that more bits be transmitted than were required for the original MELP parameters and several do not require transmission of the phonetic segmentation.

Additional testing of these techniques is still necessary, including tests that consider errors in segmentation. Testing the system with an automatically generated phonetic class segmentation will also be done as that processing comes of age. We recognize that the addition of the segmentation extraction stage to the coder will increase its computational complexity. We also note that some uses of segmentation reduce the generality of the base coder, tying it closely to a specific language.

This framework also provides fertile ground for further enhancement of a base speech coder using phonetic class segmentation. The obvious next step is to continue development of the discussed techniques. The phonetic class codebooks and superframes, while promising in their current form, could benefit significantly from additional development. One possibility not addressed above is variable framing of the input based on its phonetic class; for example, transitions could be avoided by aligning them with the analysis frame boundaries. We can also further explore models specific to each class, as was done for fricatives using bandwidth extension and demonstrated for plosives by Unno [13]. Another goal is to combine the complementary methods into a single speech coder, so the benefits of each may be accrued; they could also be combined with techniques not related to phonetic segmentation. Additionally, while these techniques were developed on a MELP coder base, there is no reason to assume that they would be limited to use with that coder.

5. References

[1] J. R. Deller, J. G. Proakis, and J. H. L. Hansen, *Discrete-Time Processing of Speech Signals.* Upper Saddle River, NJ: Prentice Hall, 1987.

[2] U. S. Dept. of Defense, "Analog to digital conversion of voice by 2400 bit/second mixed excitation linear prediction," MIL-STD-3005, Dec. 1998, draft.

[3] J. S. Collura, D. F. Brandt, and D. J. Rahikka, "The 1.2kps/ 2.4kbps MELP speech coding suite with integrated noise preprocessing," in *Proc. IEEE Mil. Comm.*, vol. 2, Atlantic City, NJ, Oct.–Nov. 1999, pp. 1449–1453.

[4] D. T. Toledano, L. A. H. Gomez, and L. V. Grande, "Automatic phonetic segmentation," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 6, pp. 617–625, 2003.

[5] Y. Zheng and M. Hasegawa-Johnson, "Acoustic segmentation using switching state Kalman filter," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1, 2003, pp. 752–755.

[6] 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.

[7] A. E. Ertan and T. P. Barnwell-III, "A C++ research and development environment for speech and audio processing applications," in *Proceedings of the Asilomar Conference on Signals, Systems and Computers*, vol. 2, Oct 2000, pp. 1449–1453.

[8] V. Krishnan, "A framework for low bit-rate speech coding in noisy environments," Ph.D. Dissertation, Georgia Institute of Technology, Mar. 2005.

[9] G. Fant, *Speech Sounds and Features.* Cambridge, MA: The MIT Press, 1973.

[10] J. Makhoul and M. Berouti, "High-frequency regeneration in speech coding systems," in *Proc. IEEE ICASSP*, vol. 4, Apr. 1979, pp. 428–431.

[11] H. Gustafsson, I. Claesson, and U. Lindgren, "Speech bandwidth extension," in *Proc. IEEE ICME*, Aug. 2001, pp. 1016–1019.

[12] D. O'Shaughnessy, *Speech Communications: Human and Machine*, 2nd ed. New York, NY: IEEE Press, 2000.

[13] T. Unno, "An improved mixed excitation linear predictive (MELP) coder," Masters Thesis, Georgia Institute of Technology, 1998.