COMPRESSION OF ACOUSTIC FEATURES – ARE PERCEPTUAL QUALITY AND RECOGNITION PERFORMANCE INCOMPATIBLE GOALS?

Roger Tucker [1], Tony Robinson [2], James Christie [2],Carl Seymour [3]

[1] Hewlett Packard Laboratories, Bristol. *roger_tucker@hp.com*
[2] Cambridge University Engineering Dept. *ajr@eng.cam.ac.uk, jdmc2@eng.cam.ac.uk*
[3] now at Commerzbank Global Equities. *carl_seymour@CommerzbankZGE.com*

ABSTRACT

The client-server model is being advocated for speech recognition over networks, where the acoustic features are calculated by the client, compressed and transmitted to the server. This has provoked a number of papers claiming that as recognition accuracy and perceptual quality are different goals, a new compression approach is needed. This is verified by experiments in which codecs such as CELP are shown to produce degraded recognition performance, but that direct quantization of acoustic features at data rates as low as 4kbps gives little or no degradation. In this paper we show that the goals are not incompatible, and that a very low bit-rate codec can be used to perform the compression. We also show that if the ability to reproduce the speech is really *not* needed, a bit rate as low as 625 bit/sec can be achieved by computing and compressing posterior phone probabilities.

1. INTRODUCTION

It is one of the ironies of speech recognition technology that the computing devices most able to run speech recognition systems are the ones that least need it. Most users are happy with a keyboard if they are seated at a desk in front of a VDU. It is when they are on the move, in a car, or their computer is too small to have a keyboard, or when they have hands or eyes busy that they need speech input. But the devices they are accessing at such times would not usually have the processor, RAM, or disk resources necessary to support large vocabulary speech recognition. However, if the device is connected to a network, the processing can take place remotely, on a machine with all the necessary resources. For this reason, the client-server approach to speech recognition is becoming of increasing interest.

Networks carrying digital speech normally use a speech compression scheme, and there have been a number of studies showing how recognition performance degrades when the speech has been compressed and decompressed by a low-bit-rate scheme running at below 16 kbit/sec [1][2]. Because recognition is about transcription accuracy but coding is about perceptual quality, recent work has proposed that new compression schemes are needed which use recognition accuracy as the optimisation criterion instead of perceptual quality. To achieve this, they compute the acoustic features used for recognition and encode them directly using standard coding methods [3][4]. Whilst these schemes achieve the required result at bit rates as low as 4kbits/sec, they have the drawback of not being able to playback the speech. As speech recognition is always prone to error, the inability to check what was said can be a serious drawback of the approach.

But is it really true that compression for listening and compression for recognition are trying to achieve different things? The most fundamental property of a speech coder is that it should be intelligible, and perceptual quality has intelligibility as its starting point. For many years military communications have been using very low bit-rate coders which are optimised entirely for intelligibility. Intelligibility and recognition accuracy are very close, if not identical, goals, and it seems plausible that a codec optimised for intelligibility would also be good for speech recognition.

The commonality goes further than that. Very-low bit-rate speech coders are parametric. Like speech recognisers, they use only basic parameters extracted from the speech. These are:

- Spectral parameters, usually LPC (Linear Predictive Coding) coefficients
- Pitch
- Voicing – either overall voicing, or in frequency bands

In fact most of the bits are used to encode the spectral parameters. Recent improvements in parametric speech coders (e.g. [5]) have brought the intelligibility of such coders to a par with higher bit-rate waveform coders such as DoD CELP (Federal Standard 1016) [6]. This shows that the basic speech parameters are being extracted quite accurately.

In this paper we investigate how a state-of-the-art parametric codec [7] can be used to both encode the speech and allow the speech to be recognised with a high degree of accuracy.

2. RECOGNITION FROM VERY LOW BIT-RATE COMPRESSED SPEECH

2.1 Wideband Extension

One advantage of explicitly encoding the acoustic features is that a wideband input signal can be used. However, we have developed a straightforward way of extending any speech codec up to 8kHz without increasing the bit-rate by more than 500 bit/sec [8]. The tests reported in [7] show that at 2.4 kbit/sec, better intelligibility is achieved through extending the codec to wideband than finer quantization of the narrowband parameters. The extension to wideband is done by splitting the band into two, encoding the lower 4kHz using standard techniques, and the upper 4kHz using 2nd order LPC.

2.2 LPC to Acoustic Features Transform

There is an important difference in the way the spectrum is parameterised in speech coding compared to speech recognition. For speech coding, LPCs are computed and converted to LSPs (Line Spectral Pairs) for encoding. For recognition, although acoustic features are sometimes derived from the LPCs, they are usually derived from a perceptually-warped (non-uniform) version of the power spectrum. The most commonly used features are MFCCs (Mel-Frequency Cepstral Coefficients). A very important question is whether acoustic features such as MFCCs can be successfully derived via the LPCs used in the speech coder.

Although the transform could be done by decoding the speech and computing the acoustic features on the decoded waveform in the normal way, this would add noise to and smooth the spectrum over time. It is faster and more accurate to transform the LPCs directly to a power spectrum as follows:

$$P(\omega) = \frac{g^2}{\left|1 - \sum_{n=1}^{p} a(n) e^{-j\omega n}\right|^2}$$

where $a(n)$ and g are the LPC coefficients and gain respectively for a frame of speech and p is the LPC model order. This in effect takes a Fourier transform of the impulse response of the LPC (all-zero) prediction filter, and then reciprocates the spectrum to derive the power spectrum of the LPC (all-pole) vocal tract filter. For a wideband codec with split bands, this needs to be done for the lower *and* upper band and then the two spectra joined together.

From this power spectrum the acoustic features can be computed in the normal way. The MFCCs which we have used for our tests are derived directly from the power spectrum, but as most front-ends represent the data as a power spectrum at some stage in the processing the transform is generally applicable.

2.2.1 Testing the Transform

To test the viability of the LPC-MFCC transform, we ran the coder without quantization, derived the MFCCs from the LPCs, and then ran a recognition task using the MFCCs as acoustic parameters.

The recogniser we have used in all our testing is the Abbot Large Vocabulary recogniser [9], which is a hybrid RNN-HMM system operating at a frame period of 16ms. The recognition task used the Resource Management (RM) database, and the baseline performance with the particular training and testing sets we used gave 5.7% error.

The results are shown in Table 1.

Train Format	Test Format	Frame Size	Error Rate
Direct MFCC	Direct MFCC	16ms	5.7%
Direct MFCC	Unquantized LPC-MFCC	16ms	13%
Unquantized LPC-MFCC	Unquantized LPC-MFCC	16ms	5.5%

Table 1: Comparison of direct MFCC and LPC-MFCC front-ends

Clearly the LPC-derived MFCCs provide a different parameterisation to the direct MFCCs as the error rate is high when training and testing are not the same. But when they are the same, the two formats give almost the same performance. This suggests that no important information is being lost by imposing an LPC model on the spectrum. Indeed, the Perceptual Linear Prediction method of acoustic analysis explicitly includes a LP step with the aim of better modelling the speech spectrum.

2.3 Performance with Quantization

In investigating the recognition performance when the LPCs are quantized, we have kept the bit-rate fixed at 2.4kbits/sec. This bit-rate is a good compromise between bit-rate and intelligibility for the *codec*, so it should be possible to get good *recognition* performance without reverting to higher bit-rates. Given a fixed bit-rate, we were able to adjust the frame-rate, having a high frame-rate and coarse quantization or a lower frame-rate and more accurate quantization. As the codec uses interframe prediction of the LSP coefficients, the higher frame-rates do not increase the quantization noise proportionately.

The results of recognising from speech compressed at 2.4 kbit/sec are shown in Table 2.

Train Format	Test Format	Frame Size	Error Rate
Unquantized	Unquantized	16ms	5.5%
Unquantized	2.4kbit/sec	16ms	7.1%
2.4kbit/sec	2.4kbit/sec	16ms	7.6%
Unquantized	Unquantized	22.5ms	10.7%
Unquantized	2.4kbit/sec	22.5ms	11.2%
2.4kbit/sec	2.4kbit/sec	22.5ms	13.4%

Table 2: Performance at 2.4 kbit/sec

The larger frame-size (and therefore more accurate spectrum) reduces the degradation due to quantization, but at the expense of a very significant increase in the overall error rate. The frame-size of the spectral data must be kept low – it would seem that the recogniser is more sensitive than human listeners to the time-smoothing of the spectrum. The slight increase in error from 5.5% to 7.1% is about what we expected in view of the level of quantization in the codec, and at the higher bit-rates of 4kbit/sec or more proposed in [3] and [4] we would expect the increase in error to be insignificant.

The better performance when the recogniser is trained on unquantized LPCs may seem strange, as one would expect best performance when training and testing are matched. But the quantization process does not add noise in any acoustic sense, rather it introduces a random distortion to the spectrum. So for a limited amount of training data, the undistorted spectrum provides better models of the quantized spectrum than the quantized spectrum itself could.

2.4 Other Speech Compression Schemes

Although we have investigated the use of a parametric codec to encode the acoustic features, if higher quality playback is required the same principles can be applied to any waveform codec that uses LPC analysis. However, in waveform codecs the LPC coefficients are used only as predictors for the waveform coding, and tend not to be as accurately coded as they would be for a parametric codec (which depends completely on the LPC parameters for reproducing the speech). For instance, they are often calculated at quite a high frame period – around 30ms.

Consequently, a waveform codec would require these modifications to guarantee good recognition performance:

- increase the frame-size of the LPC analysis and coefficient encoding to match that of the recogniser (10-16ms). Using inter-frame prediction prevents a proportional rise in the bit rate.
- ensure adequate quantization of LSPs. Because this is a smaller proportion of the overall bit rate in a waveform codec, more bits could be allocated than in the codec we have used.
- extend the codec to wideband using the generic split-band scheme in [8].
- retrain recogniser on coded (but if possible unquantized) speech.

3. COMPRESSION OF POSTERIOR PROBABILITIES

In the case where the ability to play back the speech is really *not* needed, we are interested to see if we can significantly improve on the 4 kbit/sec reported in [3] and [4]. Small computing devices (such as the current range of WinCE-based handheld computers) tend to have reasonably fast (c.100MHz) processors, and at least a Megabyte or so of RAM. This power could be used to do more processing than just the acoustic features computation.

In a hybrid RNN-HMM system like Abbot, an alternative to encoding the acoustic features is to quantize the posterior phone probabilities. Computation of these requires an extra 11 MIPS on top of the acoustic feature computation, but very little RAM and less than 500Kbytes of ROM. Because they are much further on in the processing pipeline than the front-end computations, computing these at the client allows the server to concentrate solely on the memory-intensive part of recognition (large vocabulary decoding), enabling more users to be supported with a single server.

However, even with the 255-level encoding of the log probabilities used within Abbot, 22.5 kbit/sec are needed to transmit the 45 probabilities. We have investigated reducing this to 625 bit/sec using a 10-bit vector quantizer. Initially we encoded the inter-frame differences but gave up on this when we found we got better performance encoding each frame directly.

The issue for the vector quantization process is the distance measure. Using raw probabilities, the greatest emphasis is given to the most probable phones. But the lower probability phones are often the correct ones, and these need to be encoded properly. Using log probabilities instead would work well except it would give the *very* small probability phones equal weight to all the others, even though they are rarely correct.

To solve this, we transformed the probabilities before computing distances using a $log(1+\alpha x)$ function, with α varying from 10 to 1000. This gives all the benefits of logs but prevents very small probabilities contributing to the distance.

The test setup we used for these experiments was a very demanding real-time large vocabulary task. The RNN and the vector quantizer were both trained on the WSJCAM0 read-speech database and the Language Model was derived from WSJ data. However, we tested on the SQALE test set, which gave a baseline word error rate of 20.8%.

The results for the basic distance measures are shown in Table 3 and for the *log(1+αx)* transformation in Table 4.

	% error
Baseline	20.8
Linear VQ	25.4
Log VQ	26.7

Table 3: Effect of VQ coding of posterior probabilities

α:	% error
10	23.8
30	23.6
100	23.1
300	23.7
1000	24.0

Table 4: VQ using *log(1+αx)* transform

With the *log(1+ αx)* transformation the results are better for all values of α. A value of $\alpha = 100$ gives the best performance, with just 11% increase in error rate.

The 625 bit/sec target is a very aggressive one, and depending on the application, we may well consider a higher bit-rate to avoid any drop in performance at all.

4. SUMMARY

In this paper we have argued that although perceived quality and recognition accuracy are different criteria for a speech compression scheme, they are not incompatible, and that the intelligibility criterion used for optimising very low bit-rate codecs is very similar to recognition accuracy. We have shown that the spectral encoding in a parametric codec can be used to derive acoustic features, with these provisos:

- A direct interface is created by transforming the LPC coefficients to a power spectrum from which the acoustic features can be derived
- the LPC analysis frame-rate is set to match that of the recogniser
- the codec is extended to operate on wideband speech
- the recogniser is retrained on coded, but if possible unquantized, speech.

Encoding and deriving the acoustic features in this way has the big advantage that the compressed signal can be converted back into speech and used for error correction.

If higher quality speech reproduction is needed, any LPC-based codec (eg CELP) can be adapted to give good recognition performance by incorporating these features.

In applications where there really is no advantage in being able to play back the speech, a very much lower bit rate can be achieved by compressing phone posterior probabilities instead of acoustic features. These can be computed at the client using fairly modest computing resources, allowing the server to concentrate on the memory-intensive aspects of recognition. We have investigated using a 1024 entry vector codebook to quantize the 45 phone probabilities for each frame, giving a data rate of just 625 bit/sec. We have tried a number of different transformations before quantization, the best being log(1+100x), which gives 23.1% error at 625 bit/sec against a baseline of 20.8%.

Further work will concentrate on verifying the robustness of both approaches to adverse acoustic conditions.

5. REFERENCES

[1] "The Influence of Speech Coding Algorithms on Automatic Speech Recognition", *S Euler and J Zinke*, ICASSP94, pp I-621 - I-624

[2] "Effect of Speech Coders on Speech Recognition Performance", *B.T.Lilly and K.K.Paliwal*, ICSLP 96, pp 2344-2347

[3] "Compression of acoustic features for speech recognition in network environments", *G.N.Ramaswany and P.S.Gopalakrishnan*, ICASSP98, Vol 2, pp 977-980

[4] "Quantization of cepstral parameters for speech recognition over the WWW", *V.Digalakis, L.G.Neumeyer and M.Perakakis*, ICASSP98, Vol 2, pp 989-992

[5] "A Mixed Excitation LPC Vocoder model for Low Bit Rate Speech Coding", *A.V.McCree and T.P.Barnwell III*, IEEE Trans. Speech and Audio Processing, Vol. 3, pp 242-250

[6] "The DoD 4.8 kbps Standard (Proposed Federal Standard 1016)" *J.P.Campbell Jr., T.E.Tremain and V.C.Welch*, in Advances in Speech Coding. Norwell, MA: Kluwer, 1991, pp 121-133

[7] "A low-bit-rate speech coder using adaptive line spectral frequency prediction", *C.W.Seymour and A.J.Robinson*, Eurospeech97, pp 1319-1322

[8] "Low Bit-Rate Frequency Extension Coding", *R.C.F.Tucker*, "Audio and music technology: the challenge of creative DSP", IEE Colloquium, 18 Nov 98, pp3/1-3/5

[9] "The Use of Recurrent Neural Networks in Continuous Speech Recognition", *A.J.Robinson, M.M.Hochberg and S.J.Renals*, ch. 19 in "Automatic Speech and Speaker Recognition", Kluwer 1995