JOINT ENCODING OF THE WAVEFORM AND SPEECH RECOGNITION FEATURES USING A TRANSFORM CODEC

Xing Fan* , Michael L. Seltzer, Jasha Droppo, Henrique S. Malvar, and Alex Acero

Microsoft Research, One Microsoft Way, Redmond, WA 98052

xxf064000@utdallas.edu, {mseltzer, jdroppo, malvar, alexac}@microsoft.com

ABSTRACT

We propose a new transform speech codec that jointly encodes a wideband waveform and its corresponding wideband and narrowband speech recognition features. For distributed speech recognition, wideband features are compressed and transmitted as side information. The waveform is then encoded in a manner that exploits the information already captured by the speech features. Narrowband speech acoustic features can be synthesized at the server by applying a transformation to the decoded wideband features. An evaluation conducted on an in-car speech recognition task show that at 16 kbps our new system typically shows essentially no impact in word error rate compared to uncompressed audio, whereas the standard transform codec produces up to a 20% increase in word error rate. In addition, good quality speech is obtained for playback and transcription, with PESQ scores ranging from 3.2 to 3.4.

Index Terms— transform coding, speech coding, distributed speech recognition, Siren codec

1. INTRODUCTION

In client-server speech recognition applications, a key issue is how speech information is sent to the server. One approach, called *network speech recognition* (NSR), uses a traditional voice codec on the client device. At the server, the decoded speech is is passed to the recognizer for feature extraction and recognition. Alternatively, features can be extracted directly from the codec parameters, if access to the bitstream is available [1]. In *distributed speech recognition* (DSR) [2], features are computed directly on the client device, encoded and then transmitted to the server [3, 4].

Both approaches have advantages and disadvantages. One one hand, NSR enables reconstruction of the speech waveform for transcription and diagnostic tests. However, voice codecs are typically optimized for perceptual quality, so recognition accuracy degrades compared to that with uncompressed audio, especially in noisy environments. On the other hand, DSR typically has minimal accuracy loss compared to uncompressed audio, and can operate at a lower bit rates than voice codecs, as only a small number of coefficients are transmitted per frame. However, with DSR it's very difficult to recover the speech signal. While approaches to speech reconstruction from ASR features have been proposed [5, 6], the quality is severely degraded compared to that of voice codecs. This makes it difficult to generate both transcription of the spoken words and other signal metadata, e.g. related to the speaker and environment.

We present here a new voice codec that jointly encodes both the speech waveform and the speech recognition features, with the goal of designing a system that achieves both high ASR performance and high reconstructed audio quality. Our new codec operates by encoding the acoustic features as in a DSR system, and then encoding the waveform in a manner that exploits the spectral information already captured by the ASR features, keeping the total bit rate the same. The proposed codec is called SirenSR, and is based on Siren, a transform codec for wideband audio that operates at 16 kbps and has been standardized as ITU-T recommendation G.722.1 [7].

The feature encoding scheme in SirenSR has several benefits over previously proposed DSR approaches. First, we encode the full cepstral vector, rather than a truncated version. This allows perfect reconstruction of the log mel spectrum, which enables spectral-domain speech enhancement algorithms to be applied at the server. Second, we quantize the ASR features without codebooks; this minimizes the risk that codec performance degrades when speech is observed in deployments with speech statistics different from those seen in development. Finally, our codec jointly and efficiently encodes both wideband and narrowband speech features providing compatibility with existing server-side recognizers trained from narrowband data, while enabling wideband features to be collected for building future recognizers. Our experimental results indicate that SirenSR produces recognition accuracy essentially equivalent to uncompressed speech (compared to a 20% relative increase in word error rate with the original Siren codec), with only minimal degradation in perceptual quality.

In Section 2 we present our new method for feature encoding. In Section 3 we describe our new waveform encoding using transmitted ASR features. In Section 4 we present experimental results that confirm the advantages of SirenSR.

2. DSR SYSTEM

The primary goal of SirenSR is to support distributed speech recognition (DSR) applications. As a result, we mimic conventional DSR techniques by directly encoding mel-frequency cepstral coefficients (MFCC). One disadvantage of conventional DSR techniques is that they implicitly assume that the front-end processing of the speech recognition system will not change over time. Systems built from data collected under such a codec will may not be able to take advantage of advances in codec design.

2.1. The full cepstral vector

Compared to previous DSR approaches, we extend the DSR bitstream in three ways: we retain and encode the full cepstral vector, we use simple time-series quantization and compression techniques, and jointly encode narrowband and wideband cepstral sequences.

A typical MFCC front end will calculate on the order of 24 log mel-spectral energies, rotate the vector with a DCT, and then keep

* This work was completed while the first author worked as an intern at Microsoft Research, Redmond, WA, USA.

only the first 13 resulting MFCCs. By encoding all 24 MFCCs, we maintain a more precise and dimension-independent spectral estimate. This enables the development of server feature enhancements that operate in the log mel-spectrum domain. It also enables alternative transforms for dimensionality reduction other than the conventional truncated DCT to be applied prior to recognition.

2.2. Compression of speech features

In SirenSR we use a scalar quantization technique, without codebooks. Although scalar quantization can be less efficient than vector quantization on matched data, it is desirable for two reasons. First, codebooks take up significant memory, which can be an issue on mobile devices. Second, the codebooks will only be effective at quantizing the residual error if the speech seen in deployment has statistics similar to those seen when training the codebooks.

To encode the MFCC feature vectors, we assume that the time-series sequences for each cepstral coefficient are statistically independent. Thus, we apply adaptive differential pulse code modulation (ADPCM) quantization [8] to each sequence independently.

The quantization works best when the sequences to quantize have zero mean and are temporally decorrelated. To achieve zero mean, we apply dynamic mean normalization in the encoder and decoder. Empirically, we found that this is only necessary for the first two MFCC, C0 and C1. Bits can also be assigned according to the order of MFCCs. For example, lower order of MFCCs will use more bits than higher order of MFCCs. We allocate bits uniformly for all sequences; we don't expect significant gains to come from nonuniform allocation, as the ADPCM encoders are operating at relatively low bit rates. To decorrelate the coefficients over time, we use a simple first-order predictor, subtracting a scaled version of the previously quantized value from the current sample:

$$e(n) = x(n) - \alpha\hat{x}(n-1) \tag{1}$$

The prediction error $e(n)$ is then fed to a uniform scalar quantizer.

As in conventional ADPCM, we dynamically adjust the quantization step size based on the most recent decoded value of $e(n)$. We implemented this adaptation strategy via a two-stage lookup table. The current quantized value is used as an index to look up a step size adjustment factor. This value is added to the current adjustment factor and the resulting new value is used as an index to look up a step size in the step-size table.

2.3. Joint encoding of narrowband and wideband MFCCs

Historically, telephony channels have been bandlimited to no more than than 4 kHz. As a result, the vast majority of in-domain acoustic training data is narrowband (8 kHz sampling). Deploying a narrowband DSR solution would allow for the most accurate system today, but would preclude collecting data for more accurate wideband (16 kHz sampling) systems in the future.

SirenSR incorporates acoustic features for both wideband and narrowband speech. Thus, if wideband speech acoustic models are not available, the ASR server can apply the acoustic features of narrowband speech to existing HMMs, while accumulating data for later training of wideband speech HMMs.

The differences between the MFCC of narrowband and wideband speech are mainly due to differing mel frequency filter locations, and a different number of filters. For example, in this study, the number of filters for the wideband speech is 24, ranging from 0-8 kHz, while narrowband speech is parameterized using 22 filters ranging from 0-4 kHz.

The narrowband and wideband MFCCs are highly correlated. Instead of encoding the acoustic features of each set independently, SirenSR predicts the MFCC of narrowband speech from the wideband MFCC through an affine transform. The transform parameters are estimated by minimizing the mean square error (MMSE) computed between a parallel set of narrowband and wideband MFCCs. Although the MMSE estimation matrix is obtained through a training set, it is essentially an interpolator and should be robust to acoustic mismatch in held-out data.

$$\{\mathbf{A}, \mathbf{b}\} = \underset{\{\mathbf{A},\mathbf{b}\}}{\operatorname{argmin}} \sum_i \left|\mathbf{A}\mathbf{x}_{\mathrm{WB}}(i) + \mathbf{b} - \mathbf{x}_{\mathrm{NB}}(i)\right|^2 \tag{2}$$

The estimated narrowband feature for any frame i can be obtained through Eq.3 given the corresponding decoded wideband feature $\mathbf{x}_{WB}(i)$:

$$\mathbf{x}'_{\mathrm{NB}}(i) = \mathbf{A}\mathbf{x}_{\mathrm{WB}}(i) + \mathbf{b} \tag{3}$$

To obtain a more accurate estimation of the narrowband feature, an enhancement layer can be added, where the error between the estimated narrowband features and the original narrowband features is also encoded using ADPCM. This requires the calculation of the original narrowband features at the client side, and costs additional bits for encoding the error during transmitting. Such a layer is optional, depending on performance requirements.

3. SPEECH WAVEFORM ENCODING

Most voice codecs utilize a source-filter model to encode the speech signal. The signal is modeled as the convolution of excitation signal generated by the vocal chords with a time-varying all-pole filter that represents the shape of the vocal tract. The parameters of the all-pole model are the well-known LPC coefficients that are central to most codecs. In contrast, so-called *transform codecs* do not assume a signal model at all. Instead, the waveform is converted into the frequency domain using a signal transform such as the DCT or lapped transform. The transform-domain signal is then directly quantized and encoded in some manner. Because speech recognition features are derived from the Fourier transform of the input signal, it is more efficient to compute them from transform-domain frequency representations than from LPC coefficients.

3.1. The Siren Codec

Siren is based on Siren, which is a codec originally proposed by PictureTel and standardized by the ITU-T as G.722.1[1]. We briefly review the Siren encoding algorithm here; more details can be found in the ITU-T standard [7]. Siren is a wideband codec, that is, it encodes audio sampled at 16 kHz, with a reconstruction bandwidth of 50 Hz–7 kHz. It operates on 40 ms frames (640 samples) with a 50% frame overlap. Each frame is processed by a modulated lapped transform (MLT), which results in 320 real-valued MLT coefficients per frame. Frames are independently processed. At the encoder a smooth spectral estimate is computed as follows. The MLT coefficients for each frame are first divided into 14 uniform regions between 50 Hz and 7 kHz, corresponding to a width of 500 Hz. The root-mean-square (RMS) energy in each region is computed from the MLT coefficients to provide a coarse representation of the spectral envelope.

Based on the RMS energy values, the MLT coefficients in each of the 14 regions are quantized using a process called categorization, during which a deterministic search is performed to find the set of

[1] Siren operates at 16, 24, or 32 kbps, while G.722.1 is only standardized for 24 and 32 kbps.

quantization and coding parameters that most accurately represents the MLT coefficients in each region, not exceeding the bit budget corresponding to the operating bit rate.

3.2. Waveform encoding using ASR features

As described in the previous section, the encoding performed by Siren is based on two stages 1) the computation of the smooth spectral estimate consisting of the RMS energy in 14 spectral bands and 2) the categorization procedure that encodes all of the MLT coefficients using the RMS energy values. In our proposed SirenSR codec, the 14 RMS energy values are derived from the encoded MFCC coefficients, rather than computed directly from the MLT coefficients. For that we need to address three main challenges. First, we need to compute the energy in 14 uniformly spaced frequency bands from the energy values of 24 mel-spaced frequency bands. Second, the ASR front-end and the Siren codec work with different frame sizes and frame rates. MFCCs are computed from 25 ms frames at a rate of 100 frames/sec, while Siren uses 40 ms frames at 50 frames/sec. Third, the ASR front end uses an spectral representation based on the FFT, while the Siren codec uses a spectrum derived from the MLT.

To compute the energy in 14 uniformly spaced bands, we essentially invert the MFCC processing pipeline to obtain an estimate of the power spectrum. Because of the mel-filtering operation, this process is not actually invertible and so only a smoothed estimate of the power spectrum can be obtained. In SirenSR we compute the estimated smoothed power spectrum by

$$\mathbf{x}_{\mathrm{POW}} = \mathbf{M}^{\dagger} \exp(\mathbf{C}^{-1} \hat{\mathbf{x}}_{\mathrm{MFCC}}) \tag{4}$$

where $\mathbf{M}^{\dagger}$ is the pseudoinverse of the matrix that contains the mel filterbank, $\mathbf{C}^{-1}$ is the square IDCT, and the $\exp()$ operator applies to all vector elements. From this smooth power spectrum, we can estimate the RMS energy in 14 uniformly spaced subbands between 50 Hz and 7 kHz by averaging values in the appropriate FFT bins.

The ASR front end and Siren use frequency representations based on different transforms. This means that the RMS energy values estimated from an FFT-based power spectrum may be biased when compared to those values computed from an MLT. To see this, we can write down the expression for the RMS energy in one of the 500 Hz subbands, computed from the average of 20 MLT coefficients

$$\begin{aligned} \mathrm{MLT}_{\mathrm{RMS}} &= \sqrt{\frac{1}{20}\sum_{m=0}^{19} |\mathrm{mlt}(m)|^2} \\ &= \sqrt{\frac{2}{20N}\{\sum_{m=0}^{19} [|\mathrm{fft}(m+0.5)|^2 - (-1)^m O(m)]\}} \\ &\approx \sqrt{\frac{2}{N}}\sqrt{\frac{1}{20}\sum_{m=0}^{19} |\mathrm{fft}(m+0.5)|^2} \approx \sqrt{\frac{2}{N}}\mathrm{FFT}_{\mathrm{RMS}} \end{aligned} \tag{5}$$

where $O(m) = (R^2 - I^2)\sin(2A)$, $A = (m+0.5)\pi/N$, R and I are the real and imaginary part of $\mathrm{fft}(m+0.5)$, and N is the size of the MLT. Thus, the RMS computed from the MLT differs from that computed from the FFT by a constant scale factor. Thus, the RMS energy values derived from the MFCCs must be appropriately scaled prior to use by Siren.

Because the front end uses 25 ms windows compared to the 40 ms windows used by Siren, the RMS estimate computed from the MFCC features is only accurate for a portion of the correponding codec frame. As the MFCC frame rate is twice the default Siren frame rate, we average RMS energy estimates from two consecutive MFCC feature vectors to get the estimate for the corresponding Siren frame. While this is an approximation, we have found it to work well in practice.

Once we have an estimate of the 14 RMS energy values derived from the MFCC feature vectors, the rest of the encoding process (categorization and entropy coding of the quantized values) can proceed unaltered, with the bit budget for waveform encoding reduced by the bits used in MFCC encoding, keeping the total operating bit rate unchanged. At the decoder end, the same conversion process to map from MFCC feature vectors to Siren RMS energy values is performed. These values are then used with the quantized MLT values to reconstruct the speech waveform. Note that under poor network conditions, with SirenSR we can drop the operating bit rate by sending just the MFCC representation, falling back to an DSR mode, which is not possible with the original Siren codec.

4. EXPERIMENTS

We evaluated SirenSR using an automotive command-and-control task with a vocabulary of 5,000 words, with utterances including media control, commands, and phone dialing. The acoustic data was collected from a far-field microphone; it covered three driving conditions (parked with engine on, city driving, and highway driving) and three dashboard conditions (fan off, moderate fan noise and maximum defrost). The training, development, and test sets contained 33.6, 3.8, and 4.1 hours of speech, respectively.

The feature parameters used in this study are 39-dimensional MFCCs that consist of 13 cepstral features, plus delta and delta-delta features. The frame length is 25 ms, with a 10 ms frame shift, for both wideband and narrowband speech. The acoustic models in all experiments are context-dependent HMMs with 5,500 tied states and 16 Gaussians per state.

4.1. ASR performance for wideband and narrowband speech

In SirenSR, for wideband feature compression, we quantize each dimension of the MFCC to 2 bits, corresponding to a total of 4.8 kbps. Table 1 demonstrates development-set accuracy across three different testing scenarios, where the acoustic models were trained on uncompressed PCM-derived features. The upper bound is illustrated by the PCM result, where the test features are also derived from uncompressed PCM speech. When the test signal is compressed with Siren at 16 kbps, decoded to a PCM waveform, and then converted to MFCC, there is a significant drop in accuracy, especially for the Park and City conditions. When the test MFCC are decoded from SirenSR, the performance is nearly identical to that with PCM data.

We trained two sets of HMMs using features from SirenSR and Siren, to conduct matched/mismatched recognition experiments. The results for the test set are shown in Table 2. SirenSR shows a

Test	Quiet	Park	City	Highway
PCM	91.61	84.82	89.34	76.63
Siren	90.74	75.04	80.28	73.34
SirenSR	91.67	85.02	89.69	75.31

Table 1. *Accuracy of wideband speech recognition using features extracted from PCM, Siren, and SirenSR, based on PCM trained HMMs on Dev set.*

Training/Test	Test data		
Training data	PCM	Siren	SirenSR
PCM	79.67	78.75	78.92
Siren	74.75	75.54	74.77
SirenSR	80.53	72.72	80.54

Table 2. *Average Accuracy of wideband speech recognition using features extracted from PCM, Siren, SirenSR under matched/mismatched testing condition on Test set.*

consistent significant improvement over Siren under both matched and mismatched training/test conditions.

For narrowband speech, the MMSE estimation matrix is trained using the training set using Eq. 2. An optional enhancement layer using ADPCM is added to encode the difference between the original narrowband features and the estimated narrowband features with 2.2 kbps. Thus a total of 7 kbps is spent on DSR features in this configuration.

Table 3 shows ASR accuracy for PCM, GSM, Siren, SirenSR, and SirenSR with enhanced layer (SirenSR-e). The accuracy of GSM and Siren degrades significantly with decreases in SNR, whereas both SirenSR and SirenSR-e maintain good performance, compared to the PCM matched training/test scenario.

Test	Quiet	Park	City	Highway
PCM	77.29	81.67	84.26	73.76
GSM	75.71	75.71	72.03	62.67
Siren	74.51	62.07	74.27	67.88
SirenSR	78.43	81.58	83.72	72.41
SirenSR-e	79.58	82.32	83.83	72.56

Table 3. *Accuracy of narrowband speech recognition using features extracted from PCM, GSM, Siren, SirenSR based on PCM trained HMMs.*

4.2. Speech Reconstruction

We reconstruct the encoded speech as described in Section 3. Figure 1 illustrates how the PESQ score (which measures speech reconstruction quality) varies with acoustic condition and the number of bits dedicated to reconstruction. SirenSR suffers from a 0.2 absolute drop in PESQ for the quiet condition, and matches Siren to within 0.05 for the other three conditions.

Siren at 16 kbps uses about 14 kbps to encode the speech reconstruction. At this bit rate on our data, the average PESQ score is 3.51. SirenSR uses 4.8 kbps for DSR features, leaving 11.2 kbps for speech reconstruction. At that bit rate, the average PESQ of reconstructed speech is 3.35. SirenSR-e uses an additional 2.2 kbps for the DSR features, which reduces the bits available for speech reconstruction. As a result, the average PESQ drops to 3.23. Annecdotally we've found the SirenSR reconstructed speech quality is higher than other DSR-based reconstruction algorithms (e.g. [5, 6]), which is in fact expected, as such systems typically use a lower overall bit rate. Better speech quality will decrease listener fatigue for transcription.

5. CONCLUSION

We proposed the new SirenSR voice codec for jointly encoding ASR features and speech waveforms. At 16 kbps, SirenSR results in ASR accuracy comparable to that of uncompressed speech and significantly better than the original Siren codec. In addition, PESQ evaluations show that the reconstructed waveforms have minimal loss in quality compared to Siren. The experimental results confirm that the proposed codec successfully combines the advantages of DSR and NSR to obtain both high performance speech recognition and good quality audio. The proposed methodology can be extended to many other voice codecs.

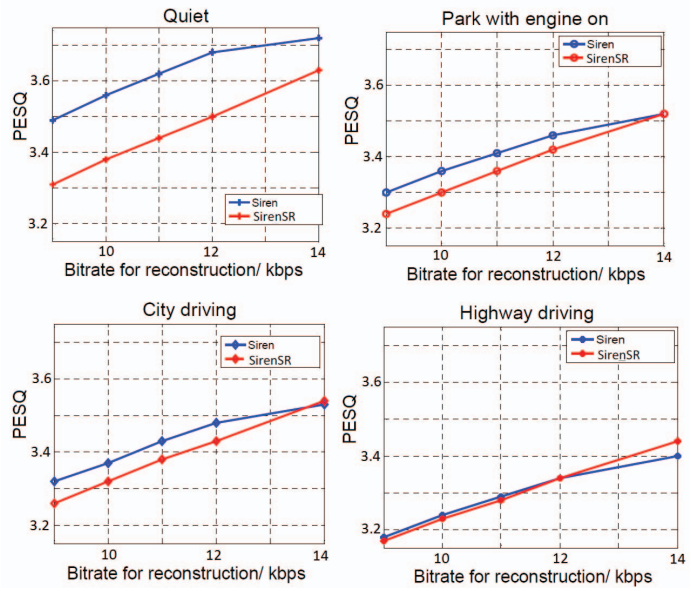

Fig. 1. *PESQ comparison under different noise conditions*

REFERENCES

[1] J. M. Huerta and R. M. Stern, "Speech recognition from GSM codec parameters," in *ICSLP*, 1998, pp. 1463–1466.

[2] ETSI ES 201 108 v1.1.2, "Speech Processing, Transmission and Quality aspects (STQ); Distributed Speech Recognition; Frontend Feature Extraction Algorithm; Compression Algorithms," *Tech. Rep. ETSI*, April 2000.

[3] G.N. Ramaswamy and P.S. Gopalakrishnan, "Compression of acoustic features for speech recognition in network environments," in *IEEE ICASSP 1998*, 1998, pp. 977–980.

[4] V. Digalakis, L. Neumeyer, and M. Perakakis, "Quantization of cepstral parameters for speech recognition over the world wide web," *IEEE Journal on selected areas in communications*, vol. 17, pp. 82–90, August 1999.

[5] D. Chazan, R. Hoory, G. Cohen, and M. Zibulski, "Speech reconstruction from mel frequency cepstral coefficients and pitch frequency," in *IEEE ICASSP 2000*, 2000, pp. 1299–1302.

[6] T. Ramabadran, J. Meunier, M. Jasiuk, and B. Kushner, "Enhancing distributed speech recognition with back- end speech reconstruction," in *ISCA Eurospeech*, 2001, pp. 1859–1862.

[7] "Low-complexity coding at 24 and 32 kbit/s for hands-free operation in systems with low frame loss," Tech. Rep. G722.1, ITU-T, 1999.

[8] N. S. Jayant P. Cummiskey and J. L. Flanagan, "Adaptive quantization in differential pcm coding of speech," *Bell System Technical Journal*, vol. 52, pp. 1105–1118, September 1973.