A Packetization and Variable Bitrate Interframe Compression Scheme For Vector Quantizer-Based Distributed Speech Recognition

Bengt J. Borgström and Abeer Alwan

Department of Electrical Engineering,
University of California, Los Angeles
jonas@ee.ucla.edu, alwan@ee.ucla.edu

Abstract

We propose a novel packetization and variable bitrate compression scheme for DSR source coding, based on the Group of Pictures concept from video coding. The proposed algorithm simultaneously packetizes and further compresses source coded features using the high interframe correlation of speech, and is compatible with a variety of VQ-based DSR source coders. The algorithm approximates vector quantizers as Markov Chains, and empirically trains the corresponding probability parameters. Feature frames are then compressed as I-frames, P-frames, or B-frames, using Huffman tables. The proposed scheme can perform lossless compression, but is also robust to lossy compression through VQ pruning or frame puncturing. To illustrate its effectiveness, we applied the proposed algorithm to the ETSI DSR source coder. The algorithm provided compression rates of up to 31.60% with negligible recognition accuracy degradation, and rates of up to 71.15% with performance degradation under 1.0%.

Index Terms: distributed speech recognition, speech coding, VQ

1. Introduction

Speech recognition systems involving separated clients and servers have become popular due to the reduced computational load at the client and the ease of model updating at the server. However, such systems, referred to as distributed speech recognition (DSR) systems, introduce the need for additional communication between the clients and server, either over wireless channels or over IP networks.

Thus, bandwidth restrictions immediately become an issue for DSR systems, since multiple clients must simultaneously communicate with a server over a single channel or over a single network. Therefore, efficient compression of transmitted speech recognition features is an important topic of research.

Many DSR compression algorithms involve vector quantization techniques to compress speech features prior to transmission. This paper introduces a compression and packetization scheme, based on the GOP concept from video coding [1], that further compresses the source coded features. The algorithm organizes speech feature frames into Groups of Frames (GOF) structures, comprised of intra-coded I-frames, predictively-coded P-frames, or bidirectionally predictively-coded B-frames. The algorithm then applies frame dependent Huffman coding to compress the individual feature frames. Furthermore, the proposed scheme is compatible with a variety of VQ-based DSR source coding algorithms.

Existing packetization schemes do not aim to compress signals, but rather to organize information prior to transmission. For example, the ETSI standard [2] packetization scheme simply concatenates 2 adjacent 44-bit source coded frames, and transmits the resulting signal along with header information. The proposed algorithm performs further compression using interframe correlation. The high time correlation present in speech has previously been studied for speech coding [3] and DSR coding [4].

The proposed algorithm allows for lossless compression of the quantized speech features. However, the algorithm is also robust to various degrees of lossy compression, either through VQ pruning or frame puncturing. VQ pruning refers to exclusion of low probability VQ codebook labels prior to Huffman coding, and thus excludes longer Huffman codewords, and frame puncturing refers to the non-transmission of certain frames, which drastically reduces the final bitrate.

2. Vector Quantizer-Based Source Coding for Distributed Speech Recognition

There exists a variety of source coding techniques that have been presented and analyzed for DSR applications, many of which involve traditional speech processing features. For example, the ETSI standard [2] uses a subset of the Mel-Frequency Cepstral Coefficients (MFCCs), as does [5]. In [6], the authors provide performance analysis for various subsets of the Linear Prediction Cepstral Coefficients (LPCCs) and the Line Spectral Pairs (LSPs).

Due to the inherent bandwidth-restrictive nature of DSR systems, quantization of speech features is a major issue. It has been shown in [4] and [7] that speech features such as those previously discussed display high correlation both across time and across coefficients. Thus VQ techniques offer the ability to efficiently reduce the bitrate of such speech features.

Though optimal quantization of vectors is generally obtained through quantization of the entire vector, as the dimension increases, the computational load and required memory of such an algorithm becomes costly [8], which is especially problematic on distributed devices. Thus, suboptimal VQ techniques such as Split VQ (SVQ) or Multi-Stage VQ (MSVQ) algorithms have been proposed for various speech coding applications [8].

The output of a general VQ-based DSR source coding algorithm can thus be given as:

$$\mathbf{f} = SC(\mathbf{s}), \tag{1}$$

where $SC(\cdot)$ represents the source coding function. Also $\mathbf{s} = [\mathbf{s}_1, \ldots, \mathbf{s}_{N_{sc}}]^T$ is the original unquantized speech feature vector, where the $\mathbf{s}_i$ are the subvectors of $\mathbf{s}$ chosen for suboptimal vector quantization and N_{sc} is the number of subvectors.

Let a given VQ scheme be referred to as VQ_k, and let the corresponding vector of codebook labels, or states, be represented by:

$$\mathbf{c}^k = \left[c_1^k, c_2^k, \ldots, c_{N_k}^k\right]^T, \tag{2}$$

where N_k is the number of codebook labels. Also, let the vector quantization of a given vector $\mathbf{s}_i$ by the given scheme VQ_k to the new vector corresponding to codebook label c_j^k be represented by:

$$\widehat{\mathbf{s}}_j^k = VQ_k\left(\mathbf{s}_i\right). \tag{3}$$

Furthermore, define the codebook label function $CB\left(\cdot\right)$, which returns the codebook label of a given quantized vector, as:

$$c_j^k = CB\left(\widehat{\mathbf{s}}_j^k\right). \tag{4}$$

Since there is a one-to-one relationship between quantized vectors and codebook labels within a given VQ scheme, there exists an inverse codebook label function such that:

$$\widehat{\mathbf{s}}_j^k = CB^{-1}\left(c_j^k\right). \tag{5}$$

Since a VQ scheme represents a discrete group of codebook labels between which a quantized signal transitions in time, the VQ scheme can be interpreted as a discrete hidden markov model (HMM). In this case, the given VQ scheme, say VQ_k, can be completely characterized by its corresponding state probabilities, π_i^k, and its corresponding transitional probabilities, a_{ij}^k, for $1 \leq i, j \leq N_k$, where N_k represents the number of codebook labels in VQ_k [9]. The state probability vector, $\overrightarrow{\pi}_k$, and state transitional matrix, $\mathbf{P}_k$, can be formed as:

$$\overrightarrow{\pi}_k = \left[\pi_1^k, \pi_2^k, \ldots, \pi_{N_k}^k\right], \tag{6}$$

and

$$\mathbf{P}_k = \begin{bmatrix} a_{11}^k & \ldots & a_{1N_k}^k \\ \vdots & \ddots & \vdots \\ a_{N_k1}^k & \ldots & a_{N_kN_k}^k \end{bmatrix}. \tag{7}$$

In order to parameterize VQ_k, the probability vector, $\overrightarrow{\pi}_k$, and the transitional probability matrix, $\mathbf{P}_k$, must be estimated. Using training data, this can be done empirically [10]:

$$\pi_i^k \approx \frac{\text{no. of samples quantized to } c_i^k}{\text{no. of total samples}}, \tag{8}$$

and

$$a_{ij}^k \approx \frac{\text{no. of samples transitioning from } c_i^k \text{ to } c_j^k}{\text{no. of total samples quantized to } c_i^k}. \tag{9}$$

3. Group of Frames (GOF) Packetization

The Group of Frames (GOF) packetization system introduced in this paper is based on the Group of Pictures (GOP) concept used for video coding [1]. The GOP concept categorizes video frames as intra-coded frames (I-frames), predictively-coded frames (P-frames), and bidirectionally predictively-coded frames (B-frames). Within each GOP, the I-frame is compressed as a sole image. P-frames are predicted based on the I-frame and prior P-frames using block motion vectors, and the error signal is compressed. Finally, B-frames are predicted based on the I-frame, and both prior and future P-frames, and the error signal is compressed. Various GOP structures allow for various compression rates, but also induce certain decoding delays and computational loads.

3.1. GOF Intra-Coded Frames

Similar to video coding utilizing GOP structure, GOF packetized I-frames are coded as sole frames. Using the notation derived in Section 2, Equation 1 can be expressed as:

$$\begin{aligned}\mathbf{f} &= \left[CB\left(VQ_1\left(\mathbf{s}_1\right)\right), \ldots, CB\left(VQ_{N_{sc}}\left(\mathbf{s}_{N_{sc}}\right)\right)\right]^T \\ &= \left[c_{L_1}^1, \ldots, c_{L_{N_{sc}}}^{N_{sc}}\right]^T.\end{aligned} \tag{10}$$

The variable bitrate I-frame packetization of speech frame $\mathbf{f}$ can then be performed by Huffman encoding [11] of each element c^m separately, for $1 \leq m \leq N_{sc}$. Distinct codeword tables are created for each c^m using the corresponding probability vectors:

$$\mathbf{h}_I^m = \left[\pi_1^m, \pi_2^m, \ldots, \pi_{N_m}^m\right]. \tag{11}$$

The N_{sc} chosen variable rate codewords created by the Huffman algorithm are then concatenated to form the bitstream representing the current intra-coded speech frame. Once the transmitted bitstream is received at the server, Huffman decoding is performed to extract the VQ codebook labels. The reconstructed feature vector, $\widetilde{\mathbf{s}}$, can then be determined as:

$$\widetilde{\mathbf{s}} = \left[CB^{-1}\left(c_{L_1}^1\right), CB^{-1}\left(c_{L_2}^2\right), \ldots, CB^{-1}\left(c_{L_{N_{sc}}}^{N_{sc}}\right)\right]^T. \tag{12}$$

3.2. GOF Predictively-Coded Frames

GOF packetized P-frames are coded as predictions of the most recent I-frame or P-frame. Therefore, the packetization of speech frame $\mathbf{f}^P$ is dependent on the transitional probabilities defined in the matrices $\mathbf{P}_m$, for $1 \leq m \leq N_{sc}$. The packetization is also dependent on the degree of separation between the current P-frame and the frame on which it is being predicted.

The $\left(i^{th}, j^{th}\right)$ element of $\mathbf{P}_m$, given by $\left[\mathbf{P}_m\right]_{i,j}$, represents the probability of transitioning from state i to state j within the VQ scheme VQ_m in 1 iteration. This relationship can be extended to the n iteration case: $\left[\mathbf{P}_m\right]_{i,j}^n$ represents the probability of transitioning from state i to state j in n transitions, where $\left[\mathbf{P}_m\right]^n$ represents the n^{th} power of the matrix $\mathbf{P}_m$.

Just as in the I-frame case, P-frames are compressed using separate Huffman codes for each vector quantized speech feature in the current frame. However, the Huffman probability vector for VQ_m in the P-frame case is given by:

$$\mathbf{h}_P^m = \left[\left[\mathbf{P}_m\right]_{i,1}^n, \left[\mathbf{P}_m\right]_{i,2}^n, \ldots, \left[\mathbf{P}_m\right]_{i,N_m}^n\right], \tag{13}$$

given that the previous I-frame or P-frame VQ_m state was state i, and that the current frame is separated from the dependent frame by n iterations. Once again, the Huffman codewords are concatenated to form the bitstream representing the compressed P-frame. After Huffman decoding is performed to extract the transmitted VQ codebook labels, the reconstructed speech feature vector is determined using Equation 12.

3.3. GOF Bidirectionally Predictively-Coded Frames

GOF packetized B-frames are coded as predictions of both the most recent I-frame or P-frame and the nearest future P-frame. The packetization of B-frames is therefore dependent on the transitional probabilities defined in the matrices $\mathbf{P}_m$, as well as the degree of separation from the prior and the future dependent frames.

Just as in the I-frame and P-frame case, Huffman codes are created to encode each vector quantized feature separately. In the B-frame case, the Huffman probability vector for VQ_m is given by:

$$\mathbf{h}_B^m = \left[[\mathbf{P}_m]_{i,1}^n \cdot [\mathbf{P}_m]_{1,j}^q, \ldots, [\mathbf{P}_m]_{i,N_m}^n \cdot [\mathbf{P}_m]_{N_m,j}^q \right], \quad (14)$$

given that the previous dependent frame was in state i with a separation of n iterations, and the future dependent frame will be in state j with a separation of q iterations. Finally, the chosen codewords are concatenated to create the bitstream for the current packetized B-frame. Similarly to the I- and P-frame case, the codebook labels are extracted through Huffman decoding of the bitstream, and the speech features are reconstructed via Equation 12.

3.4. Lossy Compression Through VQ Pruning

The packetization scheme introduced in Sections 3.1 through 3.3 provides lossless compression of source coded features. In general, for many information sources, lossless compression is ideal and at times necessary. However, due to the inherent bandwidth restrictions placed on DSR systems, lossy compression may be acceptable if performance degradation is not significant. Thus, we introduce an algorithm for lossy compression based on the GOF packetization structure and VQ pruning.

The lossy GOF algorithm is identical to the lossless scheme described in Section 3, except that certain codewords are excluded from each of the VQ schemes for each frame. A minimum probability, p_{min}, is chosen, which controls the amount of pruning, and therefore determines the amount of compression and the resulting performance degradation.

It is important to note that pruning can only be done on VQ tables that are not relied on for further compression. For example, in a GOF structure involving I-, P-, and B-frames, pruning can only be done on B-frames.

Pruning entails the exclusion from a VQ table of those entries for which the corresponding Huffman probability does not exceed the given minimum probability. For a given B-frame to be packetized, a Huffman codebook is created for each quantized feature using the probability vector $\mathbf{h}_B^m$, as shown in Equation 14. The resulting quantized codebook label using the pruned VQ table, given by $\widetilde{VQ}_m$, can then be determined by:

$$\widetilde{c}_i^m = \min_{j,\ \text{s.t. } \mathbf{h}_B^m\left(c_j^m\right) > p_{min}} \left| CB^{-1}\left(c_j^m\right) - CB^{-1}\left(c_i^m\right) \right|^2, \quad (15)$$

where c_i^m represents the original quantized state. Note that:

$$\mathbf{h}_B^m\left(c_i^m\right) > p_{min} \Leftrightarrow \widetilde{c}_i^m = c_i^m. \quad (16)$$

Huffman encoding of the quantized elements of the current frame is then performed on elements $\widetilde{c}_i^m$.

3.5. Lossy Compression Through Frame Puncturing

The results of the lossy compression and packetization algorithm described earlier are highly dependent on the predetermined value of the minimum probability p_{min}. As p_{min} is increased, the compression rate increases, but the performance of the system may degrade. We now introduce the concept of frame puncturing, which can be interpreted as VQ pruning as $p_{min} \rightarrow 1.0$.

In a punctured frame, if the minimum probability p_{min} approaches 1.0, there will be only one remaining valid codebook label for each suboptimal VQ scheme. Thus, there will be only one valid Huffman codeword for every subvector within each punctured frame. However, since it is known at the transmitter and the receiver that the most likely Huffman codeword will be chosen for every subvector, there is no need to transmit the actual bitstream for the punctured frames. Thus, the bitrate can be drastically reduced.

Assume that B-frames where punctured, and thus for given prior and future degrees of separation, the corresponding transitional probability vector is given by $\mathbf{h}_B^m$, as shown in Equation 14. Then, the reconstructed speech feature, $\widetilde{\mathbf{s}}$, vector can be determined as:

$$\widetilde{\mathbf{s}} = \left[CB^{-1}\left(c_{max}^1\right), \ldots, CB^{-1}\left(c_{max}^{N_{sc}}\right) \right]^T, \quad (17)$$

where

$$c_{max}^m = \max_i \ \mathbf{h}_B^m\left(c_i^m\right). \quad (18)$$

3.6. Delay and Complexity Analysis

The majority of computations required for lossless compression and packetization of the proposed algorithm is a result of creating frame dependent Huffman tables, which is of order $O\left(n \log n\right)$ [11], where n is the number of codebook entries. During lossy compression through VQ pruning, an additional search step of order $O\left(n\right)$ is required at the transmitter. During frame puncturing, however, the computational load is decreased, since the creation of the frame dependent Huffman tables is replaced by a simple maximization search of order $O\left(n\right)$ at the transmitter.

The GOF structure requires a certain buffer, since reconstruction of B-frames is dependent on future P-frames. Define N_p as the number of P-frames per packet, and define N_b as the number of B-frames per P-frame. The required buffer for transmission of the proposed compression and packetization scheme is given by d_{GOF}=$(N_b + 1)$ frames.

4. Experimental Results

The algorithms developed in this paper were generalized to be compatible with various VQ-based DSR source coders. However, for testing purposes, we used a source coding scheme similar to the ETSI standard coder [2] [5]. We extracted the first 13 MFCCs along with the log-energy value, to form a 14-element feature vector. The source coder used an SVQ algorithm that allocated 8 bits to the log-energy and 0^{th} MFCC coefficient pair, and 6 bits to each consecutive pair. The speech windowing frequency was set to $f_w = 100$ Hz, resulting in a payload transmission rate of 4.400 kbps as in [2].

For the ETSI front end, $N_{sc} = 7$, and each $\mathbf{s}_i$ corresponds to a MFCC vector pair. Furthermore, N_k is either equal to 256 or 64, depending on the number of bits allocated to the corresponding s_i. Lossy compression was achieved by VQ pruning

Table 1: *Word Recognition Results for Various Compression Types Applied to the ETSI Front End [2] and Tested on the Aurora-2 Database: Results include compression rates (R_c), payload transmission rates, the SNR of the compressed speech features (SNR_q), word recognition accuracy (WAcc), and absolute difference in word recognition accuracy relative to the ETSI Standard (ΔWAcc). Note that the accuracy for unquantized features is* $98.47\ \%$

GOF ($[N_p, N_b]$)	Compression Type	R_c	Rate (kbps)	SNR_q (dB)	WAcc	ΔWAcc
ETSI Standard [2]	None	$0.00\ \%$	4.400	∞	$98.20\ \%$	$0.00\ \%$
$[3, 3]$	Lossless	$21.83\ \%$	3.440	∞	$98.20\ \%$	$0.00\ \%$
$[3, 3]$	Pruning, $p_{min} = 5 \cdot 10^{-4}$	$31.60\ \%$	3.006	13.57	$98.20\ \%$	$0.00\ \%$
$[3, 3]$	Pruning, $p_{min} = 10^{-2}$	$41.65\ \%$	2.567	7.87	$97.46\ \%$	$-0.74\ \%$
$[3, 3]$	Punctured	$71.15\ \%$	1.270	7.69	$97.36\ \%$	$-0.84\ \%$

and B-frame puncturing. Finally, the decoding delay for the ETSI front end with the $[3, 3]$ GOF structure is determined to be $d_{GOF} = 40$ ms. However, for GOF structures which include B-frame puncturing, the induced decoding delay is $d_{GOF} = 0$ ms.

The algorithms discussed in Section 3 were tested on the Aurora-2 database, specifically 1000 training utterances and 500 testing utterances. The Aurora-2 database consists of digit strings spoken by both males and females. The recognition engine used word-level HMMs with 16 states and 3 mixtures per state.

Table 1 shows the results obtained for various compression types. The results shown include compression rate (R_c), payload transmission rate, the SNR of the compressed speech features (SNR_q) averaged over time and over SVQ channels, the word recognition accuracy (WAcc), and the relative absolute difference in word recognition accuracy (ΔWAcc). As can be concluded from Table 1, the proposed algorithm can achieve lossless compression of up to 21.83%. Furthermore, the proposed algorithms can achieve compression rates of up to 31.60% with negligible effect on the recognition accuracy, and can achieve compression rates of up to 71.15% with only a 0.84% drop in recognition accuracy. Note that numerous other rate-performance points can be acheived with the proposed GOF scheme by varying the GOF structure, varying the value of p_{min}, and using the option of frame puncturing. Furthermore, it is the relative compression rate achieved, R_c, as opposed to the absolute bitrate achieved, that serves as the most accurate metric of success for the proposed scheme, since many DSR source coders operate at various bitrates [4] [6].

5. Conclusions

This paper proposes a novel packetization and variable bitrate compression algorithm compatible with VQ-based DSR source coding schemes. The proposed algorithm organizes speech feature frames as Groups of Frames (GOFs) structures, and compresses corresponding frames. Huffman coding is utilized to perform lossless compression of the VQ-based source coded speech features. However, the proposed algorithm is robust to various degrees of lossy compression through VQ pruning or frame puncturing. When applied to the ETSI DSR source coder, the GOF packetization and compression algorithm is shown to provide lossless compression rates of up to 21.83%, and is shown to provide lossy compression rates of up to 71.15% with performance degradation of only 0.84% recognition accuracy. Future work includes studying the proposed scheme in the presence of background noise and over a noisy channel.

6. Acknowledgments

This work was supported in part by UC Micro and ST Microelectronics and by a Fellowship from the Radclife Institute for Advanced Study to Professor Alwan.

7. References

[1] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, *Overview of the H.264/AVC Video Coding Standard*, IEEE Trans. on Circuits and Systems for Video Technology, Vol. 13, No. 7, July 2003.

[2] ETSI ES 201 108 v1.1.2, 2000, *Distributed Speech Recognition; Front-end Feature Extraction Algorithm; Compression Algorithms*, April 2000.

[3] J. Linden, J. Skoglund, and T. Eriksson, *Improving Predictive Vector Quantizers in Speech Coding Applications*, Proc. Conf. NORSIG, Helsinki, Finland, 1996.

[4] Q. Zhu, and A. Alwan, *An Efficient and Scalable 2D DCT-Based Feature Coding Scheme For Remote Speech Recognition*, Proc. ICASSP, vol. 1, May 2001, pp. 113-116.

[5] A. M. Peinado, V. Sanchez, J. L. Perez-Cordoba, and A. J. Rubio, *Efficient MMSE-Based Channel Error Mitigation Techniques. Application to Distributed Speech Recognition Over Wireless Channels*, IEEE Trans. on Wireless Communications, Vol. 4, No. 1, January 2005.

[6] A. Bernard, and A. Alwan, *Low-Bitrate Distributed Speech Recognition for Packet-Based and Wireless Communication*, IEEE Trans. on Speech and Audio Processing, Vol. 10, No. 8, November 2002.

[7] A. Bernard, and A. Alwan, *Source and Channel Coding for Remote Speech Recognition Over Error-Prone Channels*, in Proc. ICASSP, vol. 4, May 2001, pp. 2613-2616.

[8] A. M. Kondoz, *Digital Speech: Coding for Low Bitrate Communication Systems*, John Wiley and Sons, 2004.

[9] R. G. Gallager, *Discrete Stochastic Processes*, Kluwer Academic Publishers, 1996.

[10] A. M. Peinado, A. M. Gomez, V. Sanchez, J. L. Perez-Cordoba, and A. J. Rubio, *Packet Loss Concealment Based on VQ Replicas and MMSE Estimation Applied To Distributed Speech Recognition*, Proc. ICASSP, Vol. 1, pp. 329-332, 2005.

[11] T. M. Cover, and J. A. Thomas, *Elements of Information Theory*, Wiley Interscience, 2006.