AN EFFICIENT AND SCALABLE 2D DCT-BASED FEATURE CODING SCHEME FOR REMOTE SPEECH RECOGNITION

Qifeng Zhu and Abeer Alwan

Department of Electrical Engineering, UCLA
Los Angeles, CA 90095
{qifeng,alwan}@icsl.ucla.edu

ABSTRACT

A 2D DCT-based approach to compressing acoustic features for remote speech recognition applications is presented. The coding scheme involves computing a 2D DCT on blocks of feature vectors followed by uniform scalar quantization, run-length and Huffman coding. Digit recognition experiments were conducted in which training was done with unquantized cepstral features from clean speech and testing used the same features after coding and decoding with 2D DCT and entropy coding and in various levels of acoustic noise. The coding scheme results in recognition performance comparable to that obtained with unquantized features at low bitrates. 2D DCT coding of MFCCs together with a method for variable frame rate analysis [Zhu and Alwan, 2000] and peak isolation [Strope and Alwan, 1997] maintains the noise robustness of these algorithms at low SNRs even at 624 bps. The low-complexity scheme is scalable resulting in graceful degradation in performance with decreasing bit rate.

1. INTRODUCTION

In certain applications, such as speech recognition over the World Wide Web and dictation via low-power cellular phones, there is a need for client-server recognition systems in which the recognition system is located at a remote server and the client performs less complex tasks such as feature extraction or signal compression.

There are two approaches to the remote recognition problem. The first involves coding the speech signal, transmitting the data, decoding the bitstream and performing feature extraction for ASR (e.g., [6]) or the bitstream is directly transformed to ASR feature vectors (e.g., [1]). In the second approach, which is the focus of this paper, feature extraction is first performed, then the features are compressed and transmitted to a remote server for recognition. This approach may be preferred if one had access to uncompressed speech signals and no playback is necessary, since transmitting the feature vectors can greatly reduce the bit rate with relatively low-computational cost.

In [3], the authors evaluated uniform and non-uniform scalar quantization, vector quantization, and product-code quantization of ASR features and achieved bit rates between 1.2 kbps-10.4 kbps with corresponding degradation in recognition performance. Ramaswamy and Gopalakrishnan [7] compressed acoustic features for speech recognition by using linear prediction and a two-stage vector quantizer to quantize prediction errors resulting in a 4 kbps scheme with nearly no loss in recognition performance. In [8], the authors used first order linear prediction and entropy constrained scalar quantization to compress Mel-Frequency Cepstral Coefficients (MFCCs), which are commonly used as a front end for ASR. The scalable, in bit rate, system resulted in good recognition accuracy at less than 1 kbps. None of these feature coding schemes, however, were evaluated in the presence of acoustic noise.

In this paper, a two-dimensional (2D) Discrete Cosine Transform (DCT) based coding method is used to compress ASR feature vectors. The 2D DCT is widely used in image compression and has been used to compress line spectral pairs (LSP) for speech coding [4]. We will show that the 2D DCT together with entropy coding can be used to compress MFCC feature vectors effectively at low bitrates.

2. OVERALL DESCRIPTION OF THE ALGORITHM

At the client, speech is first segmented into frames, features are computed for each frame, and then blocks of features are generated. A 2D DCT is then performed on each block and components with the lowest energy are set to zero. This is followed by scalar quantization, run-length and Huffman encoding. A block diagram of the encoder is shown in Figure 1. At the receiver, decoding and IDCT are performed and feature vectors corresponding to each frame are inputted to the ASR system. Only the feature vectors are encoded and sent to the recognition server; the first and second derivatives are computed at the server based on the recovered features. In the following sections, each of these operations is explained.

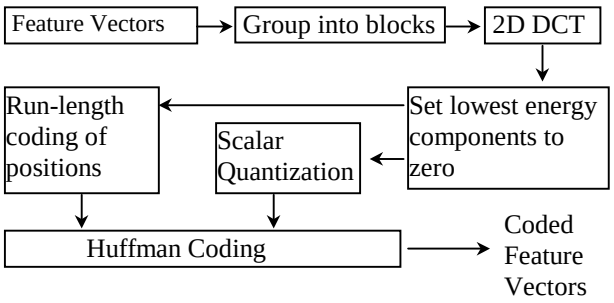

Figure 1. Block diagram of the DCT and entropy encoder.

2.1 2D DCT of Feature Vectors

Front-end processing for speech recognition systems converts the speech waveform into a sequence of feature vectors computed for 20-30 ms overlapping segments. A common set of feature vectors used for ASR are the MFCCs [2] which are computed by integrating an initial power spectrum estimate that is weighted by bandpass-filters whose bandwidths approximate those of auditory filters (typically 26 filters). A logarithmic function is then used to compress the magnitude of the power estimates, and the spectral estimate for each frame can be roughly decorrelated using a 1D DCT. The first component (c(0)), which is related to signal energy, is usually not considered for ASR but the following 12 DCT components with their first and second derivatives are.

Several techniques for making the MFCCs noise robust have been proposed such as liftering [5], and peak isolation (enhancement of the peak-to-valley ratio) [9]. In addition, in [10] we showed that variable frame rate (VFR) processing can decrease the average frame rate of transmitting feature vectors while improving recognition performance in noise. VFR is based on using energy weighted distance metrics.

In this paper, feature vectors are transmitted in a stream of blocks and for each block a 2D-DCT is applied. The motivation for performing a 2D-DCT is to exploit inter-frame correlations among feature vectors which are attributed to underlying temporal redundancies in the speech signal. Signals are first windowed with 25 ms overlapping Hamming windows (window shift is 10 ms). A block of features is generated by stacking together feature vectors for 12 frames. Hence, each block is 12×12 where the columns are MFCC vectors for each frame and the rows are MFCCs of the same order in 12 frames. If we denote each N×N block of feature vectors as a matrix, U, then the 2D DCT transformed matrix, V, can be computed as: $V=AUA^T$. The elements of the matrix A are:

$$a(i,j) = \begin{cases} \frac{1}{\sqrt{N}}, & i = 0, 0 \le j \le N-1 \\ \sqrt{\frac{2}{N}} \cos\frac{\pi (2j+1)i}{2N}, & 1 \le i \le N-1, 0 \le j \le N-1 \end{cases}$$

(Eq. 1)

The DCT results in energy compaction, with energy concentrated at the low-order components, which makes effective compression possible. Since MFCCs are generated by computing a DCT in the first place, intra-frame correlation of MFCCs is small, even after truncation and liftering. Inter-frame correlation of MFCCs of the same order, however, is high, so after the 2D-DCT, energy is compacted to the lower order components (in the row direction) resulting in large values for the first column in each block.

Figures 2 and 3 illustrate the energy compaction property of the DCT. Figure 2 shows the MFCCs for the digit /one/ as spoken by a female talker. The result of computing the 2D DCT on three 12×12 blocks of that utterance is shown in Figure 3. Note that the beginning of each block (corresponding to its first column) has the highest energy components.

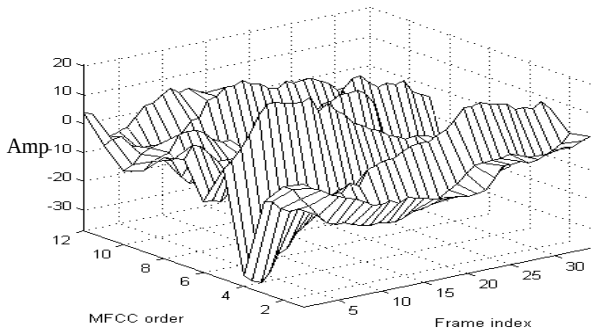

Figure 2. MFCCs for the digit /one/ spoken by a female talker.

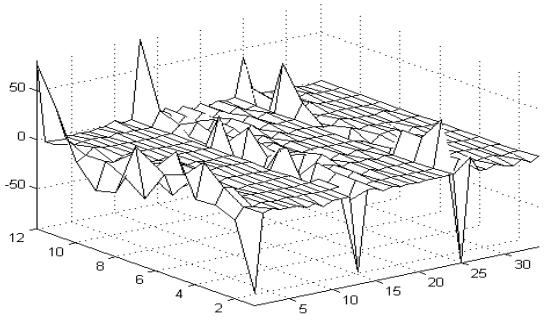

Figure 3. Three 12×12 blocks of MFCCs after 2D DCT for the same digit shown in Figure 2.

To reduce the bit rate, the lowest energy elements in each block are set to zero. We define α as the ratio of the 2D DCT size (144) to the number of components in each block which are not set to zero. We illustrate the distortion effects from this operation with an example. Consider the MFCC transformed blocks in Figure 3. If we set to zero the lowest energy components such that $\alpha = 8$, and then perform a 2D IDCT, we obtain the MFCC features shown in Figure 4. Note that the general shape of the MFCCs is preserved but finer details are not.

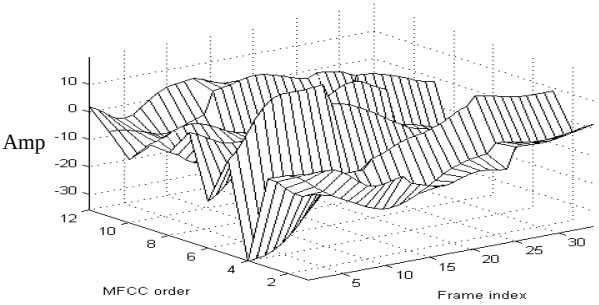

Figure 4. MFCCs for the same digit used in Figure 2 when α=8 (after performing a 2D IDCT).

The nonzero components in each block are then quantized using uniform scalar quantization.

2.2 Quantization, Run-Length and Huffman Coding of DCT Blocks

Since the first column of each block has the highest energy components, we assign β-1 bits for quantizing the components of all columns except the first one; the first column is assigned β bits per component.

Run length encoding of the position of non-zero components is performed. Each 2D DCT block is converted into a one-dimensional signal by concatenating the columns in each block. For the first 12 numbers (which constitute the first column of the 2D-DCT block) we label the zero components and for the rest, we label the non-zero components. This is because the components of the first column of each block are often not set to zero. Consequently, the number of labeled components is often smaller than the number of non-zero components. The run lengths are computed between pairs of two adjacent labeled components. The run length of the first labeled component is computed as its absolute position minus one. One extra run length of -1 is used as the symbol for the end-of-block (EOB). Based on the statistics of the run length and MFCC quantized values, one can compute the number of bits per point needed for Huffman coding. The overall bit rate of the coding scheme can be approximated as follows:

block_rate[bits_per_point*number_of_nonzero_points + bits_per_run_length*(number_of_labeled_points+1)]*

(Eq. 2)

The coding scheme is scalable allowing the user to choose the appropriate bit rate. In the event of heavy internet traffic, for example, the number of bits assigned to each component (β) could be decreased and/or the DCT preservation rate (α) increased, resulting in lower overall bit rates.

3. RECOGNITION RESULTS

Digit recognition experiments were done with HTK2.1 with the speaker-dependent TI46 database (8 males and 8 females, 12.5 kHz sampling rate). For each digit a 4-state left-to-right Hidden Markov Model (HMM) with 2 Gaussian mixtures was trained using 160 utterances. Two steps of Maximum Likelihood (ML) and Expectation Maximization (EM) training, and diagonal covariance matrices were used. Silence portions were not included. Several feature vectors were compared in terms of recognition performance: 1) MFCCs, 2) Peak isolated MFCCs (MFCCP) [9], 3) Variable frame rate MFCCs with peak isolation (VFR_MFCCP) [10], and 4) DCT coded-decoded version of MFCCs, MFCCPs, and VFR_MFCCPs.

The Hamming window length was 25ms, and window shift was 10ms. Training and testing were done with MFCCs (or modified MFCCs) together with their first and second derivatives. Training was performed on un-quantized feature vectors from clean speech while testing was done with signals corrupted by various levels of additive speech-shaped noise. Testing was performed with features after being coded and decoded using the techniques described above with 480 balanced utterances for the same talkers (the first 3 utterances from each talker from the test database). The models were re-trained with peak isolation and VFR in Experiments 2-4.

Recognition results are shown in Tables 1-4 as a function of SNR. In Table 1, results are shown for unquantized MFCCs, MFCCPs, and VFR_MFCCPs. The best performance is obtained with VFR_MFCCPs. The ratio of the average frame rate of VFR_MFCCP to a fixed rame rate version is 1:1.7 [10].

For the 2D DCT and entropy coding scheme, we experimented with several values of α (2,4,6,8) and β (4,5,6,7,8). At β=4 bits/component, recognition results were significantly worse than testing with unquantized MFCCs. At β=8 bits/component, recognition results were as good as using β=7 bits/component. When β=5, 6, and 7 bits/component, the recognition accuracy/bitrate tradeoff was best for α=6, 4, and 2, respectively. These results are shown in Tables 2-4 for the three feature vectors (MFCC, MFCCP, and VFR_MFCCP). The corresponding bit rates are 1248, 2057 and 3783 bps (Table 2), 1087, 1770, and 3291 bps (Table 3), and 624, 1030, and 1936 bps (Table 4). Note that even though some of feature components were set to zero and the remaining components were represented by few bits, the scheme maintained recognition accuracy which is comparable to that obtained with unquantized features. Also note that the degradation in recognition accuracy is less when using peak-isolated MFCCs than with MFCCs. The reason for this is that inter-frame correlations are higher for the MFCCPs, thus resulting in a higher degree of energy compaction in the 2D DCT blocks. This is most striking for VFR_MFCCP where even at a bitrate of 624 bps, recognition results are significantly improved over the baseline system with unquantized MFCCs at low SNRs. Table 5 shows an example of bit distribution for three different (α,β) pairs for the TI46 database using variable frame rate analysis with MFCCP. The overall bitrate is computed using Eq. 2 with average values obtained from the test database. The block rate is 4.9 blocks/second.

Table 6 illustrates the graceful degradation in recognition performance with decreasing bitrate. The table shows recognition results as a function of SNR when β=6 bits/component and α varies between 2 and 8.

We also evaluated the scheme using the TIDIGIT speaker-independent database (80 talkers for training and 32 different talkers for testing) and similar trends were observed. For that database, a scheme with α=4 and β=6 results in recognition performance which is comparable to that with unquanitzed MFCCP and VFR_ MFCCP features as shown in Table 7.

SNR:	20dB	15dB	10dB	5dB	0dB
MFCC	99.2	98.1	92.5	66.3	34.0
MFCCP	98.5	97.3	92.3	75.2	44.0
VFR_MFCCP	99.2	98.8	97.3	88.3	61.0

Table 1. Digit recognition accuracy (in percent) as a function of SNR for unquantized feature vectors (MFCC, MFCC with peak isolation, and MFCC with variable frame rate and peak isolation) for the TI46 database.

SNR:	20dB	15dB	10dB	5dB	0dB
α=6, β=5	98.1	98.1	80.8	52.7	30.4
α=4, β=6	99.6	97.9	89.2	62.7	31.3
α=2, β=7	99.4	98.5	92.1	65.8	32.9

Table 2. Recognition accuracy when using MFCCs after coding/decoding by the 2D DCT and entropy coding scheme at 3 bitrates: 1248, 2057 and 3783 bps for rows 1-3, respectively.

SNR:	20dB	15dB	10dB	5dB	0dB
α=6, β=5	99.2	97.5	91.0	74.6	44.0
α=4, β=6	98.8	97.1	91.5	74.2	43.1
α=2, β=7	98.5	97.3	92.1	74.6	44.0

Table 3. Recognition accuracy when using MFCCPs after coding/decoding by the 2D DCT and entropy coding scheme at 3 bitrates: 1087, 1770, and 3291 bps for rows 1-3, respectively.

SNR:	20dB	15dB	10dB	5dB	0dB
α=6, β=5	97.9	97.7	93.1	83.1	55.4
α=4, β=6	98.8	98.3	96.7	86.7	56.3
α=2, β=7	99.2	99.0	97.3	87.9	59.4

Table 4. Recognition accuracy when using VFR_MFCCPs after coding/decoding by the 2D DCT and entropy coding scheme at 3 bitrates : 624, 1030, and 1936 bps for rows 1-3, respectively.

(α, β)	average number of labeled points per block	bits per run-length (avg.)	number of nonzero points per block	bits per point (avg.)	overall bit rate (bps)
6, 5	16.86	3.48	24	2.72	624
4, 6	27.49	3.17	36	3.33	1030
2, 7	62.13	2.23	72	3.53	1936

Table 5. An example of the distribution of bits for three different (α,β) pairs after Huffman coding using VFR_MFCCPs with the TI46 database.

SNR:	20dB	15dB	10dB	5dB	0dB
α=2	99.2	98.3	96.9	88.5	59.0
α=4	98.8	98.3	96.7	86.7	56.3
α=6	98.8	97.7	93.5	82.5	51.0
α=8	98.1	97.3	92.5	77.3	46.9

Table 6. Graceful degradation in recognition accuracy as the bitrate decreases (α=2-8). VFR_MFCCPs after coding/decoding by the 2D DCT and entropy scheme are used. β=6.

SNR:	20dB	15dB	10dB	5dB	0dB
MFCCP	98.1	94.7	87.1	80.9	65.2
MFCCP+DCT	98.4	94.7	86.2	81.5	65.2
VFRMFCCP	99.4	98.1	93.7	89.0	75.9
VFRMFCCP +DCT	99.4	98.2	93.4	89.0	76.2

Table 7. Digit recognition accuracy using the TIDIGIT database at different SNRs with α=4 and β=6 for MFCCP, and VFR_MFCCP with and without 2D DCT and entropy coding/decoding.

4. SUMMARY AND CONCLUSION

In this paper, a 2D DCT-based approach is used for coding feature vectors to achieve a scalable scheme with graceful degradation in recognition performance at the lower rates. The low-complexity scheme maintains the robustness of unquantized features in noise. When using MFCCs together with a method for variable frame rate analysis [10] and spectral peak isolation [9], error rates are significantly lower than those with unquantized MFCCs even at 624 bps for an isolated digit recognition task. While the method was tested for MFCC-based features, it could be applied to other ASR feature vectors as well. It could also be easily extended to continuous ASR tasks which is the focus of our current work.

The current version of the technique introduces a block-sized delay (approximately 120 ms for the fixed frame rate version and 204 ms for VFR). For the Internet, or any packet switching network, this delay may not be critical since packetization delay is inevitable. In fact, if a packet, for example, contains at least one block of coded features then the coding delay will not be noticeable. Delay could be reduced by using smaller block sizes. Future work will examine the effects of block size on recognition performance.

Acknowledgment

Work supported in part by NSF, HRL and Broadcom through the UC MICRO program, and ST Microelectronics.

5. REFERENCES

[1] S. Choi; H. Kim; H. Lee and R. Gray "Speech recognition method using quantized LSP parameters in CELP-type coders". Electron. Lett. , Vol. 34, no.2, IEE, 1998, p.156-7.

[2] S. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," IEEE Trans. on ASSP, Vol. 28, No. 4, 1980, p. 357-366.

[3] V. Digalakis, L. Neumeyer, and Perakakis. M "Quantization of Cepstral parameters for Speech Recognition over the World Wide Web," IEEE JSAC, Vol. 17. No. 1 Jan. 1999, p. 82-90.

[4] Farvardin, N and Laroia, R. "Efficient encoding of speech LSP parameters using the discrete cosine transformation". Proc. ICASSP 1989, Vol. 1, p 168-171.

[5] B. Juang, L. Rabiner, and J. Wilpon, "On the use of bandpass liftering in speech recognition," IEEE Trans. ASSP, Vol. 35, pp. 947-954, July 1987.

[6] B. Lilly and K. Paliwal, "Effect of speech coders on speech recognition performance," Proc. ICSLP 1996, Vol.4, p.2344-47.

[7] G. Ramaswamy, and P. Gopalakrishnan, "Compression of Acoustic Features for Speech Recognition in Network Environments," Proc. IEEE ICASSP 1998, p. 977-980.

[8] N. Srinivasamurthy, A. Ortega, Q. Zhu, and A. Alwan, "Towards Efficient and Scalable Speech Compression Schemes for Robust Speech Recognition Applications," Proc. IEEE ICME 2000, p. 249-52 Vol.1.

[9] B. Strope and A. Alwan, "A model of dynamic auditory perception and its application to robust word recognition." IEEE Trans. on SAP, Vol. 5, No. 2, p. 451-464, Sep. 1997.

[10] Q. Zhu and A. Alwan, "On the use of variable frame rate analysis in speech recognition," Proc. IEEE ICASSP 2000, Vol. III, p. 1783-1786.