LOW BIT-RATE FEATURE VECTOR COMPRESSION USING TRANSFORM CODING AND NON-UNIFORM BIT ALLOCATION

Ben Milner and Xu Shao

School of Information Systems, University of East Anglia, Norwich, UK
bpm@sys.uea.ac.uk, x.shao@uea.ac.uk

ABSTRACT

This paper presents a novel method for the low bit-rate compression of a feature vector stream with particular application to distributed speech recognition. The scheme operates by grouping feature vectors into non-overlapping blocks and applying a transformation to give a more compact matrix representation. Both Karhunen-Loeve and discrete cosine transforms are considered. Following transformation, higher-order columns of the matrix can be removed without loss in recognition performance. The number of bits allocated to the remaining elements in the matrix is determined automatically using a measure of their relative information content. Analysis of the amplitude distribution of the elements indicates that non-linear quantisation is more appropriate than linear quantisation. Comparative results, based on both spectral distortion and speech recognition accuracy, confirm this. Speech recognition tests using the ETSI Aurora database demonstrate that compression to bits rates of 2400bps, 1200bps and 800bps has very little effect on recognition accuracy. For example at a bit rate of 1200bps, recognition accuracy is 98.0% compared to 98.6% with no compression.

1. INTRODUCTION

In recent years the accuracy of speech recognition systems has reached a level where useable services can be deployed. This, coupled with the enormous growth in mobile and internet areas, has lead to a range of speech-based services being deployed, or planned, for both fixed and mobile users.

A recent improvement has been distributed speech recognition (DSR) [1] where the speech codec on the terminal device is replaced by the feature extraction component of the speech recogniser. This removes harmful codec distortions from the speech recogniser input. Incorporating robust speech features, noise compensation and packet loss compensation into the DSR system enables good recognition performance across a range of environmental conditions.

An important issue concerning DSR is the compression of the speech feature vectors and their resulting bit-rate. The ETSI Aurora DSR standard [1] defines a split vector quantisation compression scheme where pairs of coefficients are allocated their own codebook. The resulting bit rate is 4800bps. The aim of this work is to further reduce the bit-rate needed for encoding speech feature vectors whilst retaining good recognition performance on both clean and noisy speech.

The proposed scheme is described in section 2 and extends previous work [2] by combining transform coding, non-uniform allocation of bits and non-linear quantisation. A set of experimental results is described in section 3 which tests the scheme at bit rates of 2400bps, 1200bps and 800bps. Finally a conclusion is made in section 4.

2. TRANSFORM-BASED COMPRESSION

This section describes the proposed transform-based feature vector compression scheme as illustrated in figure 1.

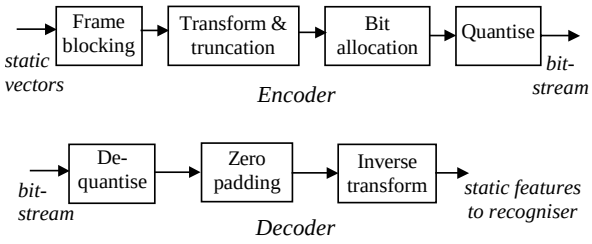

Figure 1: General feature vector compression system.

The encoder is located on the terminal device and receives N-dimensional feature vectors from the front-end processing component at a rate of f_v per second. Currently this work uses an ETSI Aurora-based front-end (MFCCs *0* to *12* and log energy resulting in an $N=14$ dimensional feature vector at a rate of $f_v=100$ frames per second) although other front-ends are equally applicable. Coding and quantisation results in a low bit representation for transmission or storage purposes.

At the decoder the bitstream is converted back into a stream of static feature vectors which can be augmented with temporal information and delivered to the recogniser for classification.

2.1. Frame Blocking

The stream of N-dimensional feature vectors, $\boldsymbol{x}_t$, are grouped together into non-overlapping blocks, $\boldsymbol{B}_k$, each containing M frames, as illustrated in figure 2, where

$$\boldsymbol{B_k} = \{\boldsymbol{x}_{(k-1)M+1}, \ldots, \boldsymbol{x}_{kM}\} \quad (1)$$

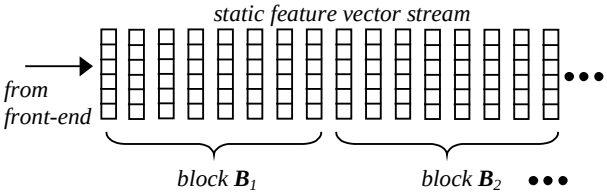

Figure 2: Blocking of static feature vectors.

The number of blocks generated per second, f_b, is related to the feature vector frame rate, f_v, and the number of frames in each block, M, and is calculated

$$f_b = f_v / M \qquad (2)$$

Preliminary tests [2] established that a suitable block width is $M=8$ which results in a rate of $f_b=12.5$ blocks per second.

2.2. Transform Coding

The overlapping nature of the feature extraction process together with the underlying speech production mechanism results in feature vectors exhibiting high levels of temporal correlation. This correlation can be exploited through transform coding to reduce the number of coefficients necessary to represent a block of feature vectors. A number of transforms have been proposed [3] for encoding temporal variations of feature vectors and include the Karhunen-Loeve transform (KLT) and the discrete cosine transform (DCT).

2.2.1. Comparison of KLT and DCT for Encoding

The KLT is the optimal transform for encoding temporal variations of the feature vectors within a block and is derived from a set of training data. For encoding the block of feature vectors a separate KLT was computed for each of the N (14) rows of the block. In practice it was found that the N transforms were almost identical to one other. Figure 3 shows the similarity between the basis functions of the data derived KLT and the explicit DCT – for the first four basis functions.

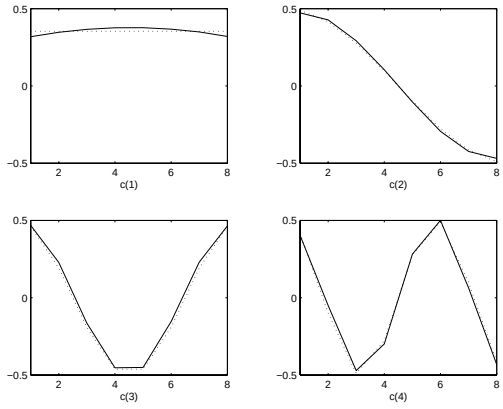

Figure 3: Basis functions of KLT (solid) and DCT (dotted).

Encoding is achieved by applying the transform to the time component of each cepstral coefficient contained within the block, $\boldsymbol{B}_k$. This results in an NxM matrix, $\boldsymbol{D}_k$, where each element, $d_{n,m}$, is calculated,

$$d_{n,m} = \sum_{j=0}^{M-1} x_{j,n}\, w_{j,m} \qquad 0 \le m \le M\text{-}1, \quad 0 \le n \le N\text{-}1 \qquad (3)$$

with $x_{j,n}$ being the n^{th} cepstral coefficient of the j^{th} MFCC vector in the block, $\boldsymbol{B}_k$, and $w_{j,m}$ the j^{th} element of the m^{th} basis function of the transform (KLT, DCT or other suitable transform). The N rows of matrix $\boldsymbol{D}_k$ still correspond to the N elements of the original feature vector while the M columns encode their temporal movement.

2.2.2. Truncation of Matrix

Lower-order columns of the matrix represent either stationary or low frequency temporal variations of the feature vector stream whilst higher-order columns contain faster moving temporal information. The particular modulation frequency represented by each column of the matrix is determined by the frequency of the associated basis function. For example, with $M=8$ and $f_v=100$, the modulation frequencies associated with each basis function, and hence each column, are:

0Hz, 7.1Hz, 14.3Hz, 21.4Hz, 28.6Hz, 35.7Hz, 42.9Hz, 50.0Hz

Both perceptual and automatic speech recognition studies [4] have shown that modulation frequencies between 1Hz and 16Hz are most useful for discrimination. This implies that higher-order columns of the matrix can be removed which results in a truncated NxM' matrix where M' specifies the number of columns retained.

2.3. Quantisation of Coefficients

This section considers the allocation of bits to the remaining elements of the matrix and their subsequent quantisation using both linear and non-linear methods.

2.3.1. Allocation of Bits to Coefficients

The average number of bits, $\bar{r}$, available to represent each element in the matrix depends on the overall bit rate, c, (governed by the channel), the block rate, f_b, and the total number of elements, NxM' in the truncated matrix, where

$$\bar{r} = \frac{c}{f_b \times N \times M'} \qquad (4)$$

To maximize the information contained in the quantised matrix, the allocation of bits to each element should be based on a measure of their relative discriminative content. Several studies [5] have shown that the amount of discriminative information varies for different cepstral coefficients. Coefficients such as energy and lower-order MFCCs contain more discriminative information than higher-order MFCCs and should therefore be allocated more bits. Similarly, lower-order columns of the matrix represent stationary or slow moving information which is more important for reconstructing the feature vector stream than higher-order columns. These columns should also be allocated relatively more bits.

This indicates that a non-uniform allocation of bits to each element will give better utilisation of their limited availability. A useful method [6] for optimising bit allocation is based on minimising the variance of the reconstruction error after quantisation. Using this scheme, the bit allocation, $r_{n,m}$, to each element, $d_{n,m}$, in the matrix can be computed as,

$$r_{n,m} = \bar{r}NM' + \frac{1}{2}\log_2\left[\frac{\sigma_{n,m}^2}{\prod_{n=0}^{N-1}\prod_{m=0}^{M'-1}\left(\sigma_{n,m}^2\right)^{1/NM'}}\right] \qquad (5)$$

where, $\sigma_{n,m}^2$, is the variance of element $d_{n,m}$ computed from a set of training data and the $\bar{r}NM'$ term represents the total

number of bits available for the quantising the matrix. The resultant bit allocation will not necessarily be an integer or even positive. Bit allocations are rounded to the nearest integer and those with zero or negative allocation are discarded.

Bit allocations for MFCCs *0* to *12* are determined using equation (5). However the row representing log energy is considered separately and at present is allocated 1 bit more than the row for encoding the zeroth cepstral coefficient. Table 1 shows bit allocation for an *N=14, M=8, M'=4, c=2400 bps* system. The number of bits available to the block is 192 which gives an average allocation (cf. eq. 4) of 3.43 bits/coefficient.

	Col.0	*Col.1*	*Col. 2*	*Col. 3*
c_0	6	4	3	2
c_1	5	4	3	2
c_2	5	3	3	2
c_3	5	3	3	2
c_4	5	4	3	2
c_5	5	4	3	2
c_6	5	4	3	2
c_7	5	4	3	2
c_8	5	3	3	3
c_9	4	3	3	2
c_{10}	4	3	3	2
c_{11}	4	3	3	2
c_{12}	4	3	3	2
E	7	5	4	3

Table 1: Bit allocation at 2400 bps for *M'=4, N=14* matrix.

The bit allocation follows that which intuition would suggest - more bits to lower-order MFCCs and lower-order columns

2.3.2. Non-Linear Quantisation of Coefficients

Previous work [2] used linear quantisation to encode each element, $d_{n,m}$, of the matrix as one of $2^{r_{n,m}}$ linearly spaced levels. Inspection of the amplitude levels for each element of the matrix revealed a set of non-linear distributions. For example, figure 4 shows the distribution of amplitude levels for coefficients $d_{1,1}$ and $d_{7,1}$ of the matrix averaged over 500 digit strings. Imposed on the graphs are Laplacian (solid line) and Gaussian (dotted line) approximations to the distributions.

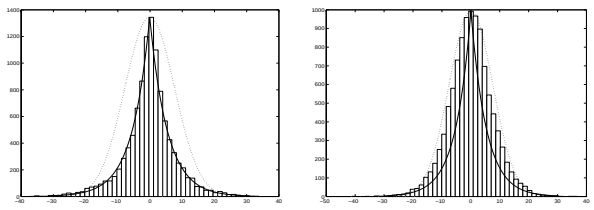

Figure 4: Distribution of amplitudes for element $d_{1,1}$ and $d_{7,1}$

Both illustrations confirm that the distribution of amplitude values is far from uniform. The distribution of amplitudes for element $d_{1,1}$ is very close to the Laplacian distribution. Conversely, the distribution of amplitudes for element $d_{7,1}$ is much closer to the Gaussian distribution. Similar observations were made for other elements in the matrix.

It is therefore more appropriate to use non-linear quantisation of the amplitudes based on the underlying probability density function (PDF). An effective technique for determining these non-linear quantisation levels and boundaries is the Lloyd-Max algorithm [6]. This uses an assumption of the underlying PDF of the amplitudes and iteratively adjusts the levels/boundaries to minimise quantisation error.

To establish the non-linearly spaced levels and boundaries the Lloyd-Max algorithm was applied to each element, $d_{n,m}$ of the truncated matrix with the number of quantisation levels determined from the bit allocation – i.e. $2^{r_{n,m}}$. This results in a set of centroid and boundary positions for each of the *NxM'* elements of the matrix. To determine whether the amplitude distributions are better modeled by a Laplacian or Gaussian PDF, the Lloyd-Max algorithm was applied twice; first to establish boundary/centroid positions based on a Laplacian PDF and secondly based on a Gaussian PDF.

As a preliminary test to compare the effect of non-linear quantisation with linear quantisation a distortion measure was used. It was decided to measure the distortion in the spectral domain as this is more meaningful than the cepstral domain where some coefficients dominate the measurement. The amount of spectral distortion is also important when considering speech reconstruction from MFCCs. The spectral distortion was computed by measuring the RMS error in the resulting log filterbank domain between a quantised and un-quantised (original) MFCC vector. In each case the log filterbank vector was obtained from an inverse DCT of the zero padded MFCC vector. Table 2 shows the average spectral distortions across 500 digit strings using an *N=14, M=8* and *M'=2* configuration. Comparisons are made between linear, non-linear Laplacian-based and non-linear Gaussian-based quantisation at bit rates of 2400bps, 1200bps and 800bps.

Bit rate	*r*	*Linear*	*Non-linear (Laplacian)*	*Non-linear (Gaussian)*
2400bps	6.9	1.995	1.995	2.030
1200bps	3.4	2.215	2.126	2.166
800bps	2.3	3.250	2.692	2.717

Table 2: Spectral distortion resulting from linear and non-linear quantisation with *M=8* and *M'=2* at varying bit rates.

As the bit rate falls, the spectral distortion introduced by all quantisers increases. However at lower bit rates the spectral distortion introduced by the two non-linear quantisers is significantly less than that introduced by the linear quantiser. Modelling the distributions by the Laplacian PDF gives slightly less spectral distortion than with the Gaussian PDF.

2.4. Decoding from Bitstream to Feature Vectors

After quantisation the resulting sequence of bits is ready for transmission. Extra bits for channel coding may be added but this work addresses only the issue of source coding.

Decoding is essentially the reverse of the encoding process. The received matrix is zero padded by *M-M'* columns and inverse transformed back into a block of *M* static feature vectors. These are input into the back-end of the recogniser.

3. RESULTS

The experiments to analyse the effectiveness of the proposed compression techniques are based on the Aurora TI digits

database which comprises 28000 digit strings for testing and 8440 for training. The speech is sampled at 8kHz and parameterized into *14*-dimensional static feature vectors, comprising MFCCs *0* to *12* and log energy. Velocity and acceleration components are computed from the quantised features at the back-end of the recogniser. The digits are modeled using 16-state, 3-mode, diagonal covariance matrix HMMs, trained from uncompressed digits strings.

To evaluate the performance of the compression schemes in both quiet and noisy conditions, two sets of experiments have been used. The first set is tested under clean conditions and the other at an SNR of 10dB. The baseline performance of the MFCC features with no compression is 98.6% for clean speech and 92.9% for the 10dB condition.

Tables 3, 4 and 5 show digit accuracy at bit rates of 2400bps, 1200bps and 800bps for a block size of *N=14* and *M=8*. The tables compare linear and non-linear quantisation and use the allocation of bits as described in section 2.3.1. The DCT is used for transform coding and the non-linear quantiser is based on the Laplacian PDF. The truncated matrix width, *M'*, is varied from 2 to 8 columns. Results are shown for both clean and 10dB SNR contaminated speech. The average number of bits per coefficient, $\bar{r}$, is also shown.

	$\bar{r}$	*Clean, %*		*Noisy, %*	
		Linear	*Non-lin.*	*Linear*	*Non-lin.*
M'=8	1.7	92.6	98.2	75.6	92.8
M'=4	3.4	98.4	98.6	93.1	93.4
M'=2	6.9	98.3	98.3	91.1	91.2

Table 3: Compression of Aurora digits at 2400bps.

	$\bar{r}$	*Clean, %*		*Noisy, %*	
		Linear	*Non-lin.*	*Linear*	*Non-lin.*
M'=8	0.9	7.7	94.3	7.2	73.1
M'=4	1.7	39.7	97.7	8.4	86.8
M'=2	3.4	98.0	98.0	90.6	90.1

Table 4: Compression of Aurora digits at 1200bps.

	$\bar{r}$	*Clean, %*		*Noisy, %*	
		Linear	*Non-lin.*	*Linear*	*Non-lin.*
M'=8	0.6	7.7	21.4	7.2	8.1
M'=4	1.1	10.4	89.13	7.2	61.6
M'=2	2.3	80.1	97.1	50.9	84.0

Table 5: Compression of Aurora digits at 800bps.

The results demonstrate that transform-based compression achieves good recognition performance at bit rates down to 800bps. In particular with a bit rate of 800bps only a 1.6% fall in recognition performance is observed for clean speech.

At lower bit rates the non-linear quantisation scheme is shown to clearly outperform the linear quantisation scheme. For example, at a bit rate of 800bps the best performance with non-linear quantisation is 97.1% with clean speech and 84.0% with noisy speech. This contrasts with 80.1% and 50.9% respectively for linear quantisation. In fact this can be generalised to the fact that the non-linear quantisation is able to make better use of a limited number of average bits per coefficient, $\bar{r}$, than linear quantisation. For example, with an average of $\bar{r}$ *=1.7* bits/coefficient (i.e. *M'=8* at 2400bps and *M'=4* at 1200bps) the non-linear quantisation scheme gives considerably better performance than the linear scheme for both clean and noisy speech.

Best performance throughout for both linear and non-linear quantisation is attained at 2400bps with *M'=4* ($\bar{r}$ =3.4). Increasing the number of columns retained to *M'=8* reduces recognition accuracy as the available number of bits for quantisation is halved (to $\bar{r}$ =1.7). This indicates that it is better to encode fewer columns with a higher number of quantisation levels than to encode more columns with a more coarse quantisation. This is situation is repeated at a bit rate of 1200bps when moving from *M'=2* ($\bar{r}$ =3.4) to *M'=4* ($\bar{r}$ =1.7). Linear quantisation performance falls drastically from 90.6% down to 8.4% in noise and from 98.0% to 39.7% for clean speech. Non-linear quantisation performance also reduces but is considerably more robust, falling from 90.1% to 86.8% in noise and from 98.0% to 97.7% for clean speech. It is therefore important to carefully select the truncation points carefully for a given bit rate.

4. CONCLUSIONS

This work has shown that a transform coding based approach for compressing an MFCC-based feature vector stream is effective at bit rates down to 800bps. This is equivalent to just 8 bits per feature vector, or 0.57 bits per mel-frequency cepstral coefficient.

Analysis has shown that the inherent temporal correlation of the feature vector stream can be exploited through transform coding to reduce the number of elements needed for encoding. A non-uniform allocation of bits to the remaining elements provides more quantisation levels to those elements important for classification. Analysis of the distribution of amplitude levels of the elements in the matrix implies that non-linear quantisation is more suitable than linear quantisation. This is confirmed from both spectral distortion measurements and from recognition tests over a range of bits rates and matrix sizes for both clean and noisy speech.

Some deterioration in performance was observed when recognising noisy speech at low bit rates. This may be in part due to the quantisation levels being estimated from clean speech and also due to the fluctuations which the noise adds.

5. REFERENCES

[1] ESTI document - ES 201 108 – STQ: DSR – Front-end feature extraction algorithm, 2000.

[2] B.P. Milner and X. Shao, "Transform-based feature vector compression for DSR", Proc. ICSLP, 2002.

[3] B.P. Milner, "Inclusion of temporal information into features for speech recognition", Proc. ICSLP, 1996.

[4] H. Hermansky and P. Jain, "Downsampling speech representation in ASR", Proc. Eurospeech, 1999.

[5] S. Nicholson, B.P. Milner and S.J. Cox, "Evaluating feature set performance using F-ratios", Eurospeech, 1997

[6] K. Sayood, "Data compression", Academic Press, 2000.