

# NON-LINEAR COMPRESSION OF FEATURE VECTORS USING TRANSFORM CODING AND NON-UNIFORM BIT ALLOCATION

Ben Milner

School of Computing Sciences, University of East Anglia, Norwich, UK

b.milner@uea.ac.uk

## ABSTRACT

This paper uses transform coding for compressing feature vectors in distributed speech recognition applications. Feature vectors are first grouped together into non-overlapping blocks and a transformation applied. A non-uniform allocation of bits to the elements of the resultant matrix is based on their relative information content. Analysis of the amplitude distribution of these elements indicates that non-linear quantisation is more appropriate than linear quantisation. Comparative results, based on speech recognition accuracy, confirm this. RASTA filtering is also considered as is shown to reduce the temporal variation of the feature vector stream.

Recognition tests demonstrate that compression to bits rates of 2400bps, 1200bps and 800bps has very little effect on recognition accuracy for both clean and noisy speech. For example at a bit rate of 1200bps, recognition accuracy is 98.0% compared to 98.6% with no compression.

## 1. INTRODUCTION

The technique of distributed speech recognition (DSR) has been shown to give useful gains in speech recognition performance [1]. This is achieved by replacing the low bit-rate speech codec on the terminal device by the feature extraction component of the speech recogniser, thereby removing codec-based distortion. Incorporating robust speech features, noise compensation and packet loss compensation into the DSR system gives good recognition performance across a range of environmental conditions.

The relatively low bit rates of the networks across which speech features may be transmitted means that a compressed representation of the speech features is necessary. The ETSI Aurora DSR standard [1] defines a split vector quantisation scheme where pairs of coefficients are allocated their own codebook. The resulting bit rate is 4.8 kbps. VQ-based compression has also been combined with linear prediction of the feature vectors [2] to give a bit rate of 4 kbps. This work aims to extend previous work on transform coding [3] to further reduce the bit-rate whilst retaining good recognition performance for both clean and noisy speech.

Section 2 outlines the transform-based compression scheme which uses a non-uniform allocation of bits and non-linear quantisation. The effect of applying RASTA filtering to the input feature vector stream is also considered. An evaluation of the compression scheme at bit rates of 2400, 1200 and 800bps is presented in section 3 across both clean and noisy speech. A conclusion is made in section 4.

## 2. TRANSFORM-BASED COMPRESSION

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

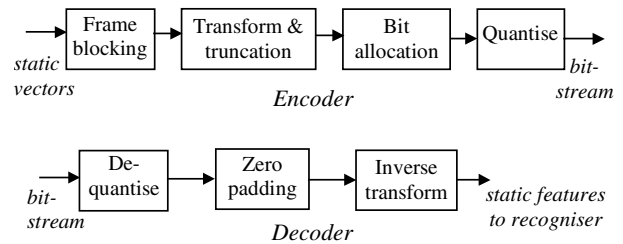


Figure 1: Overview of feature vector compression system.

The front-end processing component of the speech recogniser delivers a stream of  $N$ -dimensional feature vectors to the encoder at a rate of  $f_v$  per second. Currently, ETSI Aurora-defined MFCCs are used (MFCCs 0 to 12 and log energy which results in an  $N=14$  dimensional vector at a rate of  $f_v=100$  frames per second) although others front-ends are equally applicable. After coding and quantisation a bit stream is output for transmission or storage purposes. At the decoder the bitstream is converted back into a stream of static feature vectors which are delivered to the recogniser back-end.

### 2.1. Frame Blocking

$N$ -dimensional feature vectors,  $x_i$ , are grouped together in non-overlapping blocks,  $B_k$ , each containing  $M$  consecutive frames, as illustrated in figure 2, where

$$B_k = [x_{(k-1)M+1}, \dots, x_{kM}]^T \quad (1)$$

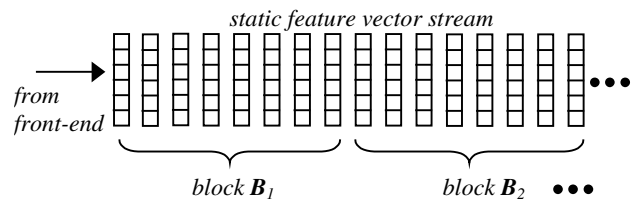


Figure 2: Blocking of static feature vectors.

The block rate,  $f_b$ , is related to the frame rate,  $f_v$ , and the number of frames in each block,  $M$ , and is given as,

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

Earlier tests [3] established that a suitable block width is  $M=8$  frames which results in a rate of  $f_b=12.5$  blocks per second.

### 2.2. Transform Coding

The overlapping nature of feature extraction, together with the underlying speech production mechanism, means that the feature vector stream exhibits a high level of temporal correlation. Transform coding can exploit this correlation and thereby reduce the number of coefficients needed to encode a block of feature vectors.

### 2.2.1. Encoding of Feature Vectors

Each block of feature vectors,  $\mathbf{B}$ , is encoded by the following matrix transformation,

$$\mathbf{D} = \mathbf{W} \mathbf{B} \quad (3)$$

where the rows of  $\mathbf{W}$  are the basis functions of the transform. This work uses the discrete cosine transform (DCT), although in practice many transforms are suitable [4]. This results in an  $M \times N$  matrix,  $\mathbf{D}$ , where the  $N$  columns correspond to the  $N$  elements of the feature vector while the  $M$  rows encode their temporal movement.

### 2.2.2. Truncation of Matrix

Lower-order rows of the matrix represent stationary or slow moving temporal variations of the feature vector stream, whilst higher-order rows contain faster moving information. The modulation frequency of each row 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 row are:

$$0\text{Hz}, 7.1\text{Hz}, 14.3\text{Hz}, 21.4\text{Hz}, 28.6\text{Hz}, 35.7\text{Hz}, 42.9\text{Hz}, 50.0\text{Hz}$$

Perceptual and automatic speech recognition studies [5] have shown that modulation frequencies between 1Hz and 16Hz are most useful for discrimination. Therefore higher-order rows can be removed to give a truncated  $M' \times N$  matrix where  $M'$  specifies the number of rows retained.

### 2.2.3. Effect of Applying RASTA Filtering

RASTA filtering has been applied successfully to many types of feature vector to reduce their susceptibility to channel distortion [8]. The RASTA filter exhibits a bandpass characteristics whereby the high-pass component of the filter removes the cepstral mean which is associated with channel distortion. The low-pass component reduces the temporal variation of the feature vector stream which can aid robustness in noisy speech.

It is interesting to investigate whether applying RASTA filtering to the MFCC feature vector stream will reduce the temporal components present in higher rows of the matrix,  $\mathbf{D}$ . To examine this, MFCC vectors from a set of 100 utterances were encoded into blocks of width  $M=8$  vectors and a DCT applied. As an indication of information content, the standard deviation of these blocks was then calculated and is shown in figure 3-a. Figure 3-b shows the standard deviation computed for the same features but with RASTA filtering applied.

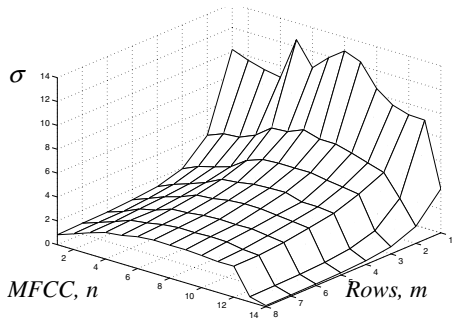


Figure 3-a: Standard deviation of MFCC transformed block

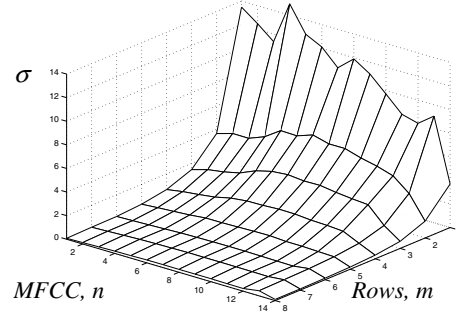


Figure 3-b: Standard deviation of RASTA-MFCC block

Figures 3-a and 3-b clearly show the reduction in standard deviation that RASTA filtering has produced. Higher-order rows of the transformed matrix after RASTA filtering have significantly lower standard deviation which implies that they can be removed from further consideration.

### 2.3. Allocation of Bits to Coefficients

The total number of bits,  $R$ , available to quantise the elements of the transformed matrix,  $\mathbf{D}$ , after truncation depends on the overall bit rate,  $c$ , (as governed by the channel) and the block rate,  $f_b$ , where

$$R = \frac{c}{f_b} \quad (4)$$

The allocation of bits to each of the  $M' \times N$  elements in the truncated matrix,  $\mathbf{D}$ , should be based on a measure of their relative discriminative content. Several studies [6] 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 rows of the matrix represent stationary or slow moving information which is more important for reconstructing the feature vector stream than higher-order rows. These rows 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 [7] for optimising bit allocation is based on minimising the variance of the reconstruction error after quantisation. Using this scheme, the bit allocation,  $r_{m,n}$ , to each element,  $d_{m,n}$ , in the matrix can be computed as,

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

where,  $\sigma_{m,n}^2$ , is the variance of element  $d_{m,n}$  computed from a set of training data. 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 allocation for MFCCs 0 to 12 can be adequately determined using equation (5). However the column representing log energy is considered separately and at

present is allocated 1 bit more than the column for encoding MFCC(0). Table 1 shows an example of bit allocation for an  $N=14$ ,  $M=8$ ,  $M'=4$ ,  $c=2400$  bps system. The total number of bits available for encoding the block is  $R=192$  which gives an average allocation of 3.43 bits per coefficient.

	$m=0$	$m=1$	$m=2$	$m=3$
$c_0$	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 resulting bit allocation reveals a structure that conforms to what would be expected. More bits are allocated to lower-order MFCCs and to rows representing slower moving temporal information.

#### 2.4. Non-Linear Quantisation of Coefficients

Earlier work [3] used linear quantisation to represent each element,  $d_{m,n}$ , of the matrix as one of  $2^{r_{m,n}}$  linearly spaced levels. However, examination of the amplitude levels for these elements revealed a series of non-linear distributions. For illustration, figure 4-a shows the distribution of amplitude levels for element  $d_{1,1}$  of the matrix, taken from 500 connected digit strings. Imposed on the graph is both a Laplacian (solid line) and Gaussian (dotted line) approximation to the distribution. For comparison figure 4-b shows the distribution of amplitude levels from element  $d_{1,7}$  of the matrix.

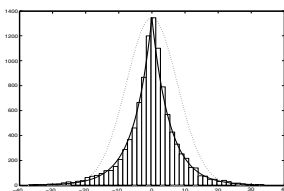


Figure 4-a: Distribution of amplitudes for element  $d_{1,1}$

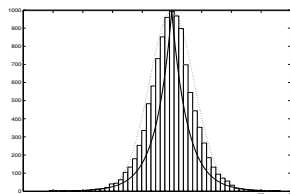


Figure 4-b: Distribution of amplitudes for element  $d_{1,7}$

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.

To accurately quantise these amplitude values it is therefore appropriate to use non-linear quantisation based on the underlying probability density function (PDF) of each element. An effective technique for determining these non-linear quantisation levels and boundaries is the Lloyd-Max algorithm [7]. This uses an assumption of the underlying PDF of the amplitude values and iteratively adjusts the levels and boundaries to minimise the resultant quantisation error.

To determine the set of non-linearly spaced quantisation levels and boundaries, the Lloyd-Max algorithm was applied to each of the  $M' \times N$  elements,  $d_{m,n}$  of the truncated matrix. The number of quantisation levels was determined from the bit allocation defined in equation 5. This results in a set of centroid and boundary positions for each of the  $M' \times N$  elements of the matrix. To determine whether the amplitude distributions are better modeled by Laplacian or Gaussian PDF, the Lloyd-Max algorithm was applied twice; first to establish boundary and centroid positions based on a Laplacian PDF and secondly based on a Gaussian PDF. The overall quantisation error of the decoded feature vector stream was then measured where upon it was found that using a Laplacian distribution gave less error.

#### 2.5. 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 the reverse of the encoding process. The received matrix is zero padded by  $M-M'$  rows and then inverted back into a block of  $M$  static feature vectors. These form the input into the back-end of the recogniser where temporal derivatives can be computed and recognition take place.

### 3. RESULTS

The experiments compare the performance of linear and non-linear quantisation and also the effect of applying a RASTA filter to the MFCC vector stream. For testing, the Aurora TI digits database has been used 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 data.

The systems are also compared under both clean and noisy (10dB SNR) conditions. Baseline performance of MFCC features with no compression is 98.6% for clean speech and 92.9% at 10dB. For RASTA filtered MFCCs, clean speech accuracy is slightly lower at 98.4% and noisy at 91.6%.

Tables 2, 3 and 4 show digit accuracy for the MFCC features encoded at bit rates of 2400bps, 1200bps and 800bps for a block size of  $N=14$  and  $M=8$ . Tables 5, 6 and 7 use the same configuration except these use RASTA filtered MFCCs. The tables compare linear and non-linear quantisation and use the allocation of bits from equation 5. The truncated matrix width,  $M'$ , is varied from 2 to 8 columns and the average number of bits per coefficient,  $\bar{r}$ , is also shown.

	$r$	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 2: Compression of MFCC vectors to 2400bps

	$r$	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 3: Compression of MFCC vectors to 1200bps

	$r$	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 4: Compression of MFCC vectors to 800bps

	$r$	Clean, %		Noisy, %	
		Linear	Non-lin.	Linear	Non-lin.
$M'=8$	1.7	94.3	98.4	83.6	91.1
$M'=4$	3.4	98.1	98.4	91.1	91.6
$M'=2$	6.9	97.8	97.8	88.6	88.6

Table 5: Compression of RASTA-MFCC vectors to 2400bps

	$r$	Clean, %		Noisy, %	
		Linear	Non-lin.	Linear	Non-lin.
$M'=8$	0.9	12.8	95.0	9.4	76.6
$M'=4$	1.7	50.7	97.3	26.7	86.1
$M'=2$	3.4	97.6	98.0	89.1	88.6

Table 6: Compression of RASTA-MFCC vectors to 1200bps

	$r$	Clean, %		Noisy, %	
		Linear	Non-lin.	Linear	Non-lin.
$M'=8$	0.6	7.3	18.6	7.2	8.0
$M'=4$	1.1	7.4	33.3	7.3	17.4
$M'=2$	2.3	72.8	96.8	45.5	83.2

Table 7: Compression of RASTA-MFCC vectors to 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 for both MFCC and RASTA-MFCC features.

At lower bit rates the non-linear quantisation scheme clearly outperforms the linear quantisation scheme for both feature types. For example, at a bit rate of 800bps, best MFCC 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.

Overall best performance for both MFCC and RASTA-MFCC features is attained at 2400bps with  $M'=4$  ( $\bar{r}=3.4$ ). Increasing the number of rows 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 rows with a higher number of quantisation levels than to encode more rows with a more coarse quantisation. This situation is repeated at a bit rate of 1200bps when moving from  $M'=2$  ( $\bar{r}=3.4$ ) to  $M'=4$  ( $\bar{r}=1.7$ ). It is therefore important to carefully select the truncation points carefully for a given bit rate.

#### 4. CONCLUSIONS

This work has shown that a non-linear transform coding-based approach for compressing feature vectors 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 these remaining elements, together with non-linear quantisation, provided fine levels of quantisation for those elements identified as being important for classification. A series of recognition tests demonstrated that this approach gives good performance in clean and noisy speech down to bit rates of 800bps. The results also indicated the importance of carefully selecting the level of truncation to ensure that sufficient bits can be allocated to the elements for quantisation. Some deterioration in performance was observed when recognising noisy speech at low bit rates. This may, in part, be due to quantisation levels being estimated from clean speech.

#### 5. REFERENCES

- [1] ESTI document - ES 201 108 – STQ: DSR – Front-end feature extraction algorithm, 2000.
- [2] G.N. Ramaswamy and P.S. Gopalakrishnan, "Compression of acoustic features for speech recognition in network environments", Proc. ICASSP, 1998.
- [3] B.P. Milner and X. Shao, "Transform-based feature vector compression for DSR", Proc. ICSLP, 2002.
- [4] B.P. Milner, "Inclusion of temporal information into features for speech recognition", Proc. ICSLP, 1996.
- [5] H. Hermansky and P. Jain, "Downsampling speech representation in ASR", Proc. Eurospeech, 1999.
- [6] S. Nicholson, B.P. Milner and S.J. Cox, "Evaluating feature set performance using F-ratios", Eurospeech, 1997
- [7] K. Sayood, "Data compression", Academic Press, 2000.
- [8] H. Hermansky and N. Morgan, "RASTA processing of speech", IEEE Trans SAP, vol. 2, pp. 578-589, 1994.