Efficient Vector Quantisation of Line Spectral Frequencies Using the Switched Split Vector Quantiser

Stephen So and Kuldip K. Paliwal

School of Microelectronic Engineering,
Griffith University, Brisbane, Australia, 4111.
s.so@griffith.edu.au
k.paliwal@griffith.edu.au

Abstract

In this paper, we investigate the use of a switched split vector quantiser (SSVQ) for coding linear predictive coding (LPC) parameters. The SSVQ is applied to quantise the LPC parameters in terms of line spectral frequencies from the TIMIT database and its performance is compared with the split vector quantiser. Experimental results show that the SSVQ provides a better trade-off between bit-rate and distortion performance than the split VQ. In addition, the SSVQ has a lower computational (search) complexity than the split VQ, though this is attained at the expense of an increase in memory requirements. In order to achieve a spectral distortion of 1 dB, the three-part SSVQ with an 8 directional switch requires 23 bits/frame, 4.41 kflops/frame of computations and 8272 floats of memory, while the corresponding values for a traditional three-part split VQ are 25 bits/frame, 13.3 kflops/frame and 3328 floats, respectively.

1. Introduction

The coding of the spectral envelope of speech, in the form of linear predictive coding (LPC) parameters, has been a problem of interest in low bit-rate speech coding. These LPC parameters are generally quantised in terms of line spectral frequencies (LSFs) using a vector quantiser (VQ) [1, 2]. Extrapolating from the operating curve of full search VQ suggests that we need about 20 bits/frame to achieve transparent coding of these parameters [3]. This corresponds to a codebook with approximately one million code-vectors. It is not possible to design such a large codebook. In addition, the computational cost of the resulting full search vector quantiser is very high, even though it performs optimally in terms of distortion performance and number of bits required.

Vector quantisers possess a number of attributes that are interrelated and these are, namely, the bit-rate, distortion, computational complexity, and memory requirement. For a given bit-rate, full search vector quantisers generally achieve the lowest distortion but they also require a large amount of searching and memory at high bit-rates. In order to alleviate this computational and memory burden, we can apply structural constraints to the vector quantiser. With tree-structured vector quantisers (TSVQ), the search complexity is very low, though they require more memory than full search VQ and, due to suboptimal searching, coding performance is degraded [4]. Classified or switch vector quantisers, which are equivalent to a two-stage TSVQ [4], also reduce the search complexity at the expense of higher memory requirements and generally suboptimal coding performance. Product-code vector quantisers, such as split and multistage [1], reduce the computational complexity and memory requirements by designing and operating independent vector quantisers, either of smaller dimension or consisting of more stages, respectively. However, their coding performance suffers because in split VQ, correlation between sub-vectors is not exploited and in multistage VQ, code-vector searches are done in a sequential fashion [6].

Multistage, split and switch vector quantisers have been reported in the speech coding literature [1, 2, 5, 7]. In this paper, we investigate the use of a hybrid of the last two techniques, called the switched split vector quantiser (SSVQ), for quantising speech LSFs. SSVQ consists of a switch VQ combined with many split VQs. It will be shown that the SSVQ provides a better trade-off than traditional split vector quantisers in terms of bit-rate and distortion performance, and offers a lower computational complexity, though at the cost of an increase in memory requirements.

2. Switched split vector quantisation

The basic idea of our approach is to populate the vector space with many local split vector quantisers while switching to one of them based on a nearest-neighbour criterion and quantising the vector using the respective codebook. Correlation that exists across all dimensions of the vector space can be exploited as these local split VQs are positioned via an optimal vector quantiser, which we refer to as the *switch vector quantiser*, that is designed using the Linde-Buzo-Gray (LBG) algorithm [9] on all the vectors. For each local split VQ, the 10-dimensional LSF vector is split either into two parts with $(4, 6)$ division or three parts with $(3, 3, 4)$ division and bits are uniformly allocated to individual parts where-ever possible [3].

2.1. LSF representation and weighted distance measure

It is common practice in speech coders to quantise the line spectral frequency (LSF) representation of the linear prediction (LP) coefficients as they possess desirable qualities such as localisation in frequency of quantisation errors and simple verification of synthesis filter stability [1, 10, 11]. Split VQs are more suitable than the multistage VQs for quantising the LSF vectors as they can easily ensure correct LSF ordering for the filter stability.

We have used the weighted distance measure from [1], which places more emphasis on the LSFs representing low frequencies as well as formant regions. The weighted distance measure, $d_w(\boldsymbol{f}, \hat{\boldsymbol{f}})$, between the original vector, $\boldsymbol{f}$, and the approximated vector, $\hat{\boldsymbol{f}}$, is defined as [1]:

$$d_w(\boldsymbol{f}, \hat{\boldsymbol{f}}) = \sum_{i=1}^{10} \left[c_i w_i (f_i - \hat{f}_i) \right]^2 \quad (1)$$

where f_i and $\hat{f}_i$ are the ith LSF in the original and approximated vector respectively. The dynamic weights, $\{w_i\}_{i=1}^{10}$, are given by

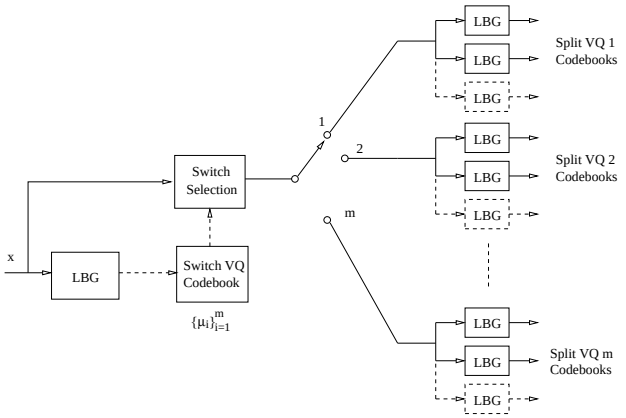

Figure 1: SSVQ Codebook Training

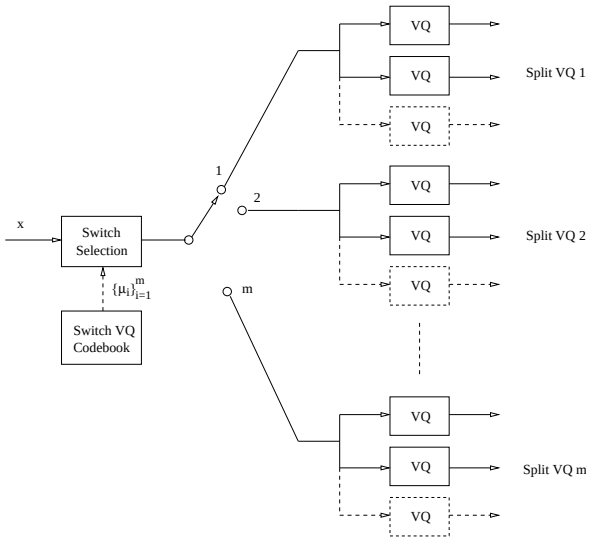

Figure 2: SSVQ Coding

[1]:

$$w_i = [P(f_i)]^r \tag{2}$$

where $P(f)$ is the LPC power spectrum and r is a constant (typical value used is 0.15). The static weights, $\{c_i\}_{i=1}^{10}$, are given by [1]:

$$c_i = \begin{cases} 1.0, & \text{for } 1 \leq i \leq 8 \\ 0.8, & \text{for } i = 9 \\ 0.4, & \text{for } i = 10 \end{cases} \tag{3}$$

2.2. SSVQ codebook training

Fig. 1 shows a block diagram of the SSVQ codebook training. The LBG algorithm [9] is first applied on all vectors to produce m centroids (or means) $\{\boldsymbol{\mu}_i\}_{i=1}^m$. In the Euclidean distortion sense, these centroids are the optimal representation of all the vectors in that Voronoi region. Hence, we can use them to form the *switch VQ codebook* which will be used for switch-direction selection. All the training vectors are classified based on the nearest-neighbour criterion:

$$j = \underset{i}{\text{argmin}}\, d_w(\boldsymbol{x} - \boldsymbol{\mu}_i) \tag{4}$$

where $\boldsymbol{x}$ is the vector under consideration and j is the cluster (or, switching direction) to which the vector is classified. With the training vectors classified to the m clusters, local split VQ codebooks are designed for each cluster (or, switching direction) using the corresponding training vectors.

2.3. SSVQ coding

Fig. 2 shows a block diagram of SSVQ coding. Each vector to be quantised is first switched to one of the m possible directions based on the nearest-neighbour criterion defined by (4), using the switch VQ codebook, $\{\boldsymbol{\mu}_i\}_{i=1}^m$, and then quantised using the corresponding split VQ.

2.4. Comparing search complexity of SSVQ with split VQ

We illustrate the search complexity of the SSVQ with respect to the split VQ through an example. Consider an SSVQ (with a 16-directional switch) operating at 24 bits/frame. Since there are $2^4 = 16$ directions, 4 bits need to be set aside to inform the decoder of the switch direction. Assuming that we are using a two-part split, each local split VQ will be given 20 bits or 10 bits per part. The total number of searches required to quantise a vector is therefore, $16 + 2 \times 2^{10} = 2064$ searches. Compare this with a two-part split VQ operating at 24 bits/frame, where a total of $2 \times 2^{12} = 8192$ searches are required. Thus, for a given bit-rate, the SSVQ has a smaller search complexity than the split VQ. This complexity is quantified in terms of floating point operations (flops) in the results section.

2.5. Minimum distortion versus nearest-neighbour selection

Quantisers that employ a soft decisioning scheme, such as those reported in [8, 12], are expected to perform better, in terms of spectral distortion. However, such schemes require a lot of of computations as each vector to be coded needs to be quantised multiple times in order to find the best representation. SSVQ uses a hard decision to determine the best split vector quantiser to use. We believe that the penalty in spectral distortion would be more than offset by the reduction in computational complexity. Fig. 3 shows the histograms of the spectral distortion from SSVQ operating at 24 bits/frame using minimum distortion (spectral distortion and weighted distance measure) and nearest-neighbour selection. As expected, the spectral distortion of the nearest-neighbour selection was the worst out of the three (0.912 dB). However, the number of computations of the nearest-neighbour selection (42 kflops/frame) is considerably less than that of the minimum distortion version (approximately 655 kflops/frame).

3. Distortion measures for LPC parameters

In order to objectively measure the distortion between a coded and uncoded LPC parameter vector, the spectral distortion is often used. For the ith frame, the spectral distortion (in dB), D_i, is defined as:

$$D_i = \sqrt{\frac{1}{F}\int_0^F \left[P_i(f) - \hat{P}_i(f)\right]^2 df} \tag{5}$$

where F is 3 kHz for speech sampled at 8 kHz [13] and $P_i(f)$ and $\hat{P}_i(f)$ are the LPC power spectra (in dB) of the coded and uncoded ith frame, given by:

$$P_i(f) = -20\log_{10}\left|A_i(e^{j2\pi f/F_s})\right| \tag{6}$$

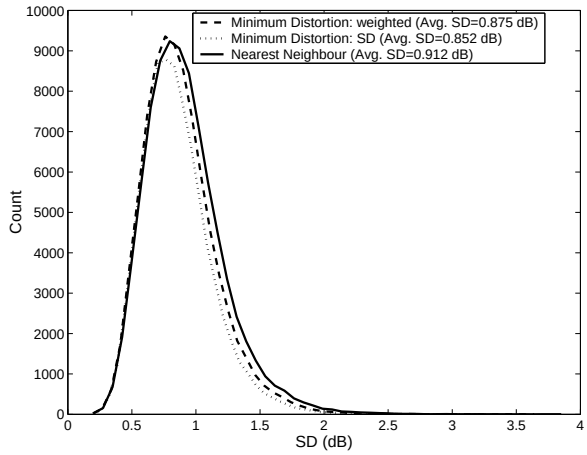

Figure 3: Spectral distortion (SD) histograms for the 24 bits/frame two-part switched split vector quantiser ($m = 16$) using minimum distortion (using weighted distance measure and spectral distortion) and nearest-neighbour selection

and

$$\hat{P}_i(f) = -20 \log_{10} \left| \hat{A}_i(e^{j2\pi f/F_s}) \right| \quad (7)$$

where $A_i(z)$ and $\hat{A}_i(z)$ are the original and quantised LPC polynomials of the ith frame respectively [1].

The conditions for transparent speech from LSF quantisation are [1]:

1. The average spectral distortion (SD) is approximately 1 dB;
2. there is no outlier frame having more than 4 dB of spectral distortion; and
3. less than 2% of outlier frames are within the range of 2–4 dB.

4. Experimental setup

The TIMIT database was used to train and test the SSVQ system. It consists of speech down-sampled to 8 kHz with a 3.4 kHz anti-aliasing filter applied. A 20 ms Hamming window is used and a tenth order linear predictive analysis is performed on each frame using the autocorrelation method [3]. High frequency compensation and bandwidth expansion of 15 Hz[1] was used to correct the effects of the anti-aliasing filter [15] as well as formant underestimation respectively [16]. The training data consists of 333789 vectors while the evaluation set, which is exclusive of the training, contains 85353 vectors. Following Atal et al. [13], spectral distortions are calculated within the frequency range of 0 to 3 kHz.

5. Results and discussion

Tables 1 and 2 show the spectral distortions, computational complexity (in flops)[2] and memory requirements (ROM) at various bit rates for a two-part and three-part split vector quantiser respectively. It can be seen that by using a two-part split VQ, we can achieve transparency on the TIMIT database using 23 bits/frame while an extra 2 bits/frame are required for transparency when using the three-part split VQ. Looking at the computational complexity, three-part split VQ requires considerably less floating point operations than the two-part split VQ since the dimensionality of the codebooks is less.

Table 1: Spectral distortion (SD), computational complexity, and memory requirements (ROM) of the two-part split vector quantiser as a function of bit-rate

Bits/frame	Avg. SD	Outliers (in %)		kflops/	ROM
$(b_1 + b_2)$	(in dB)	2–4 dB	> 4 dB	frame	(floats)
24 (12+12)	0.943	0.54	0.00	163.8	40960
23 (12+11)	1.023	1.09	0.00	114.7	28672
22 (11+11)	1.080	1.44	0.00	81.9	20480

Table 2: Spectral distortion (SD), computational complexity, and memory requirements (ROM) of the three-part split vector quantiser as a function of bit-rate

Bits/frame	Avg. SD	Outliers (in %)		kflops/	ROM
$(b_1 + b_2 + b_3)$	(in dB)	2–4 dB	> 4 dB	frame	(floats)
26 (9+9+8)	0.892	0.55	0.00	16.4	4096
25 (9+8+8)	1.001	1.38	0.00	13.3	3328
24 (8+8+8)	1.061	1.68	0.01	10.2	2560

Table 3 lists the spectral distortions for the two-part SSVQ at various bit rates and number of switch directions, m. It can be observed that, overall, the two-part SSVQ outperforms the two-part split VQ (Table 1) in all bit-rates with transparency achieved using 22 bits/frame and a 32-directional switch. This gain in spectral distortion performance over traditional split VQ may be the result of the exploitation of correlation across all dimensions by the initial switch vector quantiser, which uses the full vector dimension and is unconstrained in structure. In terms of the computational complexity, the flop counts of the two-part SSVQ are lower than traditional split VQ. Therefore, SSVQ is a low computational cost vector quantiser that provides a better trade-off in terms of bit-rate and distortion. However, these positive attributes come at the expense of a considerable increase in memory requirements, as can be seen in the ROM column of Table 3.

Table 4 shows the spectral distortions for the three-part SSVQ. Transparent coding can be achieved using 23 bits/frame as opposed to the 25 bits/frame required for a three-part split VQ (Table 2). Similar to the computational complexity reduction observed when going from two-part to three-part split VQ, three-part SSVQ requires the least number of floating point operations of all vector quantisers tested thus far. We consider the three-part SSVQ operating at 23 bits/frame and using 8 switching directions, to be the best compromise of speed and memory requirements.

6. Conclusion

In this paper, we have investigated the use of a hybrid vector quantiser called the switched split vector quantiser and applied it to the problem of coding line spectral frequencies. We have shown that the SSVQ provides a better trade-off between bit-rate and distortion performance than the traditional split VQ. For a given bit-rate, the SSVQ not only gives less spectral distortion than the split VQ, it

[1] This is the same high frequency compensation and bandwidth expansion used in the US Federal Standard 1016 4.8 kbps CELP coder described in [14]

[2] In our study, each addition, multiplication or comparison is considered a floating point operation (flop) and the flop count is used as a measure of search complexity to code each LSF vector.

Table 3: Spectral distortion (SD), computational complexity, and memory requirements (ROM) of the two-part switched split vector quantiser as a function of bit-rate and number of switch directions

Total Bits/frame ($b_m + b_1 + b_2$)	Avg. SD (in dB)	Outliers (in %) 2–4 dB	Outliers (in %) $>$ 4 dB	kflops/ frame	ROM (floats)
For $m = 8$:					
24 (3+10+11)	0.902	0.43	0.00	65.9	131152
23 (3+10+10)	0.973	0.84	0.00	41.3	82000
22 (3+9+10)	1.031	1.20	0.00	33.1	65616
For $m = 16$:					
24 (4+10+10)	0.912	0.57	0.00	41.6	164000
23 (4+9+10)	0.965	0.78	0.00	33.4	131232
22 (4+9+9)	1.038	1.29	0.00	21.1	82080
For $m = 32$:					
24 (5+9+10)	0.903	0.52	0.00	34.0	262464
23 (5+9+9)	0.972	0.94	0.00	21.8	164160
22 (5+8+9)	1.029	1.26	0.00	17.7	131392

Table 4: Spectral distortion (SD), computational complexity, and memory requirements (ROM) of the three-part switched split vector quantiser as a function of bit rate and number of switch directions

Total Bits/frame ($b_m + b_1 + b_2 + b_3$)	Avg. SD (in dB)	Outliers (in %) 2–4 dB	Outliers (in %) $>$ 4 dB	kflops/ frame	ROM (floats)
For $m = 8$:					
24 (3+7+7+7)	0.955	0.90	0.00	5.44	10320
23 (3+7+7+6)	1.006	1.21	0.00	4.41	8272
22 (3+7+6+6)	1.112	2.46	0.01	3.64	6736
For $m = 16$:					
24 (4+7+7+6)	0.925	0.75	0.00	4.73	16544
23 (4+7+6+6)	1.002	1.38	0.00	3.96	13472
22 (4+6+6+6)	1.080	1.97	0.01	3.20	10400
For $m = 32$:					
24 (5+7+7+5)	0.921	0.86	0.00	4.86	28992
23 (5+6+6+6)	0.991	1.13	0.00	3.84	20800
22 (5+6+6+5)	1.049	1.70	0.00	3.32	16704

also reduces the computational (search) complexity. These advantages come at the cost of increased memory requirements, though this can be alleviated by splitting vectors into more parts.

7. References

[1] K.K. Paliwal and B.S. Atal, "Efficient vector quantization of LPC parameters at 24 bits/frame", *IEEE Trans. Speech Audio Processing*, Vol. 1, No. 1, pp. 3–14, Jan. 1993.

[2] W.P. LeBlanc, B. Bhattarchaya, S.A. Mahmoud and V. Cuperman, "Efficient search and design procedures for robust multistage VQ of LPC parameters for 4 kb/s speech coding", *IEEE Trans. Speech Audio Processing*, Vol. 1, pp. 373–385, Oct. 1993.

[3] K.K. Paliwal and W.B. Kleijn, "Quantization of LPC parameters" in *Speech Coding and Synthesis*, W.B. Kleijn & K.K. Paliwal, Ed. Amsterdam: Elsevier, 1995, pp. 443–466.

[4] A. Gersho and R.M. Gray, *Vector Quantization and Signal Compression*, Boston: Kluwer, 1991.

[5] U. Sinervo, J. Nurminen, A. Heikkinen and J. Saarinen, "Evaluation of split and multistage techniques in LSF quantization", in *Proc. Norsig 2001*, Trondheim, Norway, Oct. 2001, pp. 18–22.

[6] V. Krishnan, D.V. Anderson and K.K. Truong, "Optimal multistage vector quantization of LPC parameters over noisy channels", *IEEE Trans. Speech Audio Processing*, Vol. 12, No. 1, pp. 1–8, Jan. 2004.

[7] S. Wang and A. Gersho, "Phonetically-based vector excitation coding of speech at 3.6 kbit/s", in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Glasgow, May, 1989, pp. I-349–352.

[8] J. Pan and T.R. Fischer, "Vector quantization-lattice vector quantization of speech LPC coefficients", in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Vol. 1, 1994, pp. 513–516.

[9] Y. Linde, A. Buzo, and R.M. Gray, "An algorithm for vector quantizer design", *IEEE Trans. Commun.*, Vol. COM-28, No. 1, pp. 84–95, Jan. 1980.

[10] F.K. Soong and B.H. Juang, "Line spectrum pair (LSP) and speech data compression", in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, San Diego, California, Mar 1984, pp. 37–40.

[11] N. Sugamura and F. Itakura, "Speech analysis and synthesis methods developed at ECL in NTT–from LPC to LSP", *Speech Commun.*, Vol. 5, pp. 199–215, Jun. 1986.

[12] A.D. Subramaniam and B.D. Rao, "PDF optimized parametric vector quantization of speech line spectral frequencies", *IEEE Trans. Speech Audio Processing*, Vol. 11, No. 2, pp. 130–142, Mar. 2003.

[13] B.S. Atal, R.V. Cox, and P. Kroon, "Spectral quantization and interpolation for CELP coders", in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Glasgow, Scotland, May 1989, pp. 69–72.

[14] J.P. Campbell Jr., V.C. Welch, and T.E. Tremain, "An expandable error-protected 4800 bps CELP Coder (U.S. Federal Standard 4800 bps voice coder)", in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Glasgow, Scotland, May 1989, pp. 735–738.

[15] B.S. Atal and M.R. Schroeder, "Predictive coding of speech signals and subjective error criteria", *IEEE Trans. Acoust., Speech, Signal Processing*, Vol. ASSP-27, No. 3, pp. 247–254, Jun. 1979.

[16] P. Kroon and W.B. Kleijn, "Linear-prediction based analysis-by-synthesis coding" in *Speech Coding and Synthesis*, W.B. Kleijn and K.K. Paliwal, Ed. Amsterdam: Elsevier, 1995, pp. 79–119.