

2014 AASRI Conference on Sports Engineering and Computer Science (SECS 2014)

Binary Search Vector Quantization

Ning-Yun Ku^a, Shun-Chieh Chang^a, Shaw-Hwa Hwang^{a,*}

^a NTUT, Electrical Engineering #1, Sec. 3, Chung-hsiao East Rd., Taipei, Taiwan, ROC

Abstract

This paper proposes a fast search algorithm for vector quantization (VQ) based on a binary search space (BSS-VQ). The trade-off and learning aspects (TLA) were used to enhance the line spectrum pair (LSP) encoder of the G.729 standard. In the trade-off aspect, a slight loss occurred in the quantization quality; however, substantial computational savings were achieved. In the learning aspect, the binary search space was developed using the learning process, which uses full search VQ (FSVQ) as an inferred function. In the experiment, computational savings of 86.19% and a quantization accuracy of 98.15% were achieved, which confirmed the excellent performance of the BSS-VQ approach.

© 2014 The Authors. Published by Elsevier B. V. This is an open access article under the CC BY-NC-ND license (<http://creativecommons.org/licenses/by-nc-nd/3.0/>).

Peer-review under responsibility of Scientific Committee of American Applied Science Research Institute

Keywords: binary search; G.729; line spectrum pair (LSP); speech codec; vector quantization (VQ);

1. Introduction

Vector quantization (VQ) is a lossy and powerful method for multimedia communications because of its excellent rate-distortion performance and simple structure. However, full search algorithms have substantial computation requirements. Thus, numerous studies have focused on the computational savings that can be gained by using the VQ approach.

Among existing fast search methods, the tree-structured VQ (TSVQ) method [2] was proposed to reduce the VQ search space. The search spaces of the TSVQ code-book are reduced to $2\log_2(N)$ which makes the

* Corresponding author. Tel.: +886930204341.

E-mail address: s94310393s@hotmail.com.

method powerful; however, the method's loss of quantization quality is problematic. The TSVQ is a binary search method that uses a single path to traverse the appropriate codewords. Thus, the TSVQ encoder does not ensure that the selected codeword is the closest to the input vector.

A previous study [1] applied triangular inequality elimination (TIE) to VQ-based image coding, achieving computational savings of more than 90%. However, the TIE approach is dependent on the correlation characteristics of input signals. The noise-like (weak correlation characteristic) input vector reduces the performance of the TIE approach.

In [3], the quasi-binary search (QBS) algorithm and trade-off aspect between TSVQ and TIE were proposed to reduce the computational complexity of the VQ algorithm. The performance was superior to that of TIE, particularly regarding the noise-like input signal. The QBS algorithm is not dependent on the correlation characteristics of input signals. Although the quantization accuracy of the QBS algorithm was 99.16%, which was reasonably good, the computational savings of 59.43% for the LSP encoder of the G.729 standard were unsatisfactory.

Although VQ is a powerful method, it confronts a number of problems in gaining computational savings for the LSP encoder of the G.729 standard. First, a moving average (MA) filters the LSP parameter, and the redundant source data is then extracted and removed. The MA filter thereby eliminates the correlation characteristics of the bias of LSP. Thus, the performance of these TIE-based algorithms decreases in the G.729 standard. Second, a small codebook size is provided in the multistage VQ of the G.729 standard. All codewords are adjacent and close to each other; therefore, the TIE algorithm cannot work efficiently. Third, the TSVQ achieves considerable computational savings, but loses substantial quantization quality. The QBS approach loses slight quantization quality and achieves unsatisfactory computational savings.

Another study [4] proposed the fast search method for VQ. However, none of those methods focus on the LSP encoder of the G.729 standard. This paper proposes a binary search space VQ (BSS-VQ) approach, which balances computational savings and quantization quality, and achieves considerable computational savings with only a slight loss in quantization quality. Moreover, the learning aspects of neural networks and VQ were also employed to improve the performance of the BSS-VQ approach. The LSP encoder with the G.729 standard was used to verify the performance of the BSS-VQ. The following section describes the BSS-VQ algorithm. The last two sections present the experimental results and conclusion.

2. BSS-VQ algorithm

This section presents a fast search BSS-VQ algorithm that employs a fast locating approach to determine a small search space and fully search it to obtain an optimally matched codeword. A learning algorithm was proposed to build a BSS, which uses full search VQ (FSVQ) as an inferred function. The BSS was trained using large amounts of training data, which efficiently obtained codewords for each subspace. However, some trifling regions existed between splitting boundaries and codeword boundaries, which implied that the training data fell in trifling regions with little possibility. Therefore, a learning algorithm with insufficient training may lead to missed encoding in the trifling region which no training data fall in when the training process, resulting in a decline in quantization accuracy; conversely, each subspace can be fully trained using a learning algorithm with sufficient training data, but it with highly overlapped codewords; a BSS with highly overlapped codewords results in a decline in computational savings. Based on the aforementioned analysis, Algorithm 1 describes the BSS generated by the learning process in the BSS-VQ algorithm. Algorithm 2, concerning the BSS-VQ encoding process, is also presented in detail.

2.1. Algorithm 1: BSS generation for BSS-VQ

Step1: BSS determination.

Step 1.1: Select

$$CB = \{cw_{i,j} | 1 \leq i \leq N, 1 \leq j \leq D\} \quad (1)$$

which is the original codebook of VQ, to train the BSS. The term N is the codebook size and D is the codeword dimension.

Step 1.2: Calculate

$$C_j = \frac{1}{N} \sum_{i=1}^N cw_{i,j}, 1 \leq j \leq D \quad (2)$$

as the centroid of CB in the j th dimension.

Step 1.3: Determine

$$BSS = \{bss_k | 1 \leq k \leq 2^D\} \quad (3)$$

as the BSS , which is generated from dichotomy splitting according to the centroid C_j . The BSS contains 2^D subspaces, which are empty of codewords at the initialization stage.

Step 1.4: Empty

$$TPS = \{tps_{k,i} | 1 \leq k \leq 2^D, 1 \leq i \leq N\} \quad (4)$$

as the training placement statistics, which are used to determine how often the inferred values fall into cw_i of bss_k .

Step 2: Learning process.

Step 2.1: The term

$$S = \{s_i | 1 \leq i \leq M\} \quad (5)$$

is adopted as the training database, S is used to build the content codeword of the BSS , where s_i is the D -dimensional vector, and M is the data number. The term M is a large number, indicating that the training data are sufficient to fully train each binary search subspace.

Step 2.2: Obtain

$$cw_n = FSVQ(s_i) \quad (6)$$

The term $FSVQ(s_i)$ denotes a full search VQ algorithm. It is used to determine the optimally matched codeword, cw_n , from CB for input vector s_i .

Step 2.3: Determine

$$bss_p = BSSVQ(s_i) \quad (7)$$

$$p = \sum_{j=1}^D 2^j \cdot f(s_i), f(s_i) = \begin{cases} 0, & s_{i,j} \geq C_j \\ 1, & s_{i,j} < C_j \end{cases} \quad (8)$$

The $BSSVQ(s_i)$ is proposed in this paper, and is used to determine the nearest subspace bss_p for input vector s_i . The $BSSVQ(.)$ requires only D times comparison.

Step 2.4: Set

$$cw_n \in bss_p \quad (9)$$

and increase $tps_{p,n}$ by one. The optimal matched codeword cw_n is inserted into bss_p if cw_n does not belong to bss_p .

Step 2.5: Repeat Steps 2.2 to 2.4 with each input vector s_i until $i=M$. The sum of TPS is M in the end.

Step 2.6: Normalize

$$TPS = \{tps_{k,l} / \text{Sum}(tps_k) \mid 1 \leq k \leq 2^D, 1 \leq l \leq N\}, \text{Sum}(tps_k) = \sum_{l=1}^N tps_{k,l} \quad (10)$$

Step 3: BSS screening.

Step 3.1: Denote

$$BSS = \{bss_{i,j} \mid 1 \leq i \leq 2^D, 1 \leq j \leq N_i\} \quad (11)$$

as the learned binary search space, where N_i is the number of codewords in bss_i . Set the optimal thresholds σ_{gsn} and σ_{vss} .

Step 3.2: Sort bss_i . The sorted

$$bss_i = \{bss_{i,j} \mid tps_{i,j} \geq tps_{i,j+1}, 1 \leq j \leq N_i\} \quad (12)$$

The codewords of bss_i are sorted in descending order by tps_i .

Step 3.3: Screen bss_i . The new

$$bss_i = \{bss_{i,j} \mid 1 \leq j \leq M_i\} \quad (13)$$

The term

$$M_i = \begin{cases} N_i, & N_i \leq \sigma_{gsn} \\ j | tps_{i,j} \geq \sigma_{vss} > tps_{i,j+1}, & N_i > \sigma_{gsn} \end{cases} \quad (14)$$

is the effective search subspace. The dismissed subspace $dbss_i = \{bss_{i,j} | M_i \leq j \leq N_i\}$ is filtered out by threshold σ_{vss} , which contains less likely codewords of bss_i .

Step 3.4: Repeat Steps 3.2 to 3.3 until $i=2^D$.

2.2. Algorithm 2: A BSS-VQ encoding process

Step 1: Denote

$$X = \{x_i | 1 \leq i \leq t\} \quad (15)$$

as the input signals, where x_i is a D -dimensional vector.

Step 2: Determine

$$bss_q = BSSVQ(x_i), q = \sum_{j=1}^D 2^j \cdot f(x_i) \quad (16)$$

where

$$f(x_i) = \begin{cases} 0, & x_{i,j} \geq C_j \\ 1, & x_{i,j} < C_j \end{cases} \quad (17)$$

The $BSSVQ(.)$ function is used to determine the subspace bss_q , and costs D times comparison.

Step 3: Obtain

$$cw_m = FSVQ(x_i) \quad (18)$$

All codewords within subspace bss_q are set as the new codebook, and the $FSVQ(.)$ is subsequently used to obtain optimally matched codeword cw_m from bss_q . The codeword number of subspace bss_q is lower than that of the original codebook. Thus, computational complexity is substantially reduced.

Step 4: Repeat Steps 2 and 3 with each input vector x_i until $i = t$.

3. Experimental results

This study used the LSP encoder in the G.729 standard with a 10-D codebook and 128 codewords to verify the performance of BSS-VQ. A speech database, which was recorded at a sampling rate of 8 kHz and a resolution of 16 bits, was used on the inside test; it contained male speech, female speech, background noise, and silence, and its data number M was 601 422 (45.8 MB). The trained binary search space whose average, minimum, and maximum number of codewords were 16.53, 8, and 26, respectively. Moreover, the 15.3 MB

speech data for the outside test data were used to verify the performance of BSS-VQ. As shown in Table 1, computational savings of 86.19% and a quantization accuracy of 98.15% were achieved. The computational savings of BSS-VQ outperform that of the QBS-VQ approach when their quantization accuracies are similar. Moreover, the quantization accuracy of BSS-VQ outperforms that of TSVQ when their computational savings are similar. These experimental results confirmed the excellent performance of the BSS-VQ approach.

Table 1. The computational saving of LSP encoder with G.729 standard for outside test with FSVQ, TIE, QBS, BSSVQ, and TSVQ algorithms

	Computational Savings	Quantization Accuracy
FSVQ	0%	100%
TIEVQ	23.65%	100%
TSVQ	89.06%	46.61%
QBSVQ	59.43%	99.16%
BSS-VQ	86.19%	98.15%

4. Conclusion

This paper presents a BSS-VQ algorithm to enhance the LSP encoder of the G.729 standard. The trade-off and learning aspects were used to achieve optimal performance. The BSS-VQ algorithm is not dependent on the correlation characteristics of input signals and outperforms most existing algorithms in the LSP encoder of the G.729 standard.

References

- [1] Choi, S.Y. and Chae, S.I. Incremental-Search Fast Vector Quantiser Using Triangular Inequalities for Multiple Anchors. *Electronics Letters*, 1192-1193; 1998.
- [2] Djamah, M. and O'S, D. An efficient tree-structured codebook design for embedded vector quantization. *IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, 4686-4689; 2010.
- [3] Yan, L.J. and Hwang, S.H. The binary vector quantisation. *Proc. Third IEEE Int. Symp. on Communication, Control, Signal Processing (ISCCSP)*, 604–607; 2008.
- [4] Chen, S.X. and Li, F.W. Fast encoding method for vector quantisation of images using subvector characteristics and Hadamard transform. *IET Image Processing*, 18-24; 2011.