DBpia

論 文

영상 부호화를 위한 벡터 양자화기에서의 고속 탐색 기법

正會員 高 鍾 錫* 正會員 金 在 均**

Fast Codebook Search for Vector Quantization in Image Coding

Jong Seog KOH*, Jae Kyoon KIM** *Regular Members*

要 約 본 논문에서는 벡터 양자화(VQ)의 탐색 복잡도를 줄이기 위한 방법을 제안한다. 본 방법은 현재 부호화하려는 벡터의 특성을 효율적으로 이용함으로써 고속 탐색 효과를 가져온다. 벡터 크기가 16인 제안하는 VQ 방식으로써 약 0.1-0.9dB의 미소한 성능 감소로 1/8-1/16의 복잡도 감소를 꾀할 수 있음을 보인다. 동시에 기존의 방식과 비교하여 더 성능이 우수함을 보인다.

ABSTRACT The paper describes a very simple algorithm for reducing the encoding complexity of vector quantization (VQ), exploiting the feature of a vector currently being encoded. A proposed VQ of 16(=4x4) vector dimension shows a slight performance degradation of about 0.1-0.9 dB, however, with only 16-32 among 256 codeword searches, i.e., with just 1/16-1/8 search complexity compared to a full-search VQ. And the proposed VQ scheme is also compared to outperform tree-search VQ with regard to their SNR performance and memory requirement.

I. INTRODUCTION

One of the most serious problems of vector quantization (VQ)[1] is the search cost or complexity to search a minimum-distortion code vector from the given codebook. It is well known that the search cost increases exponentially with both bit rate and vector dimension. During the past decade many algorithms have been developed for vector quantizers. These studies can be grouped into two approaches. First, there are efforts to reduce costs through prior mathematical manipulation, while maintaining intact the structure of the conventional exhaustive-search VQ (ESVQ). Soleymani and Morgera[2] performed a test before computing the distortion for each code vector and rejected those code vectors which

*, **韓國科學技術院 電氣및 電子工學科

Korea Advanced Institute of Science and Technology Electrical Engineering

論文番號 : 88-30 (接受 1988. 6. 7)

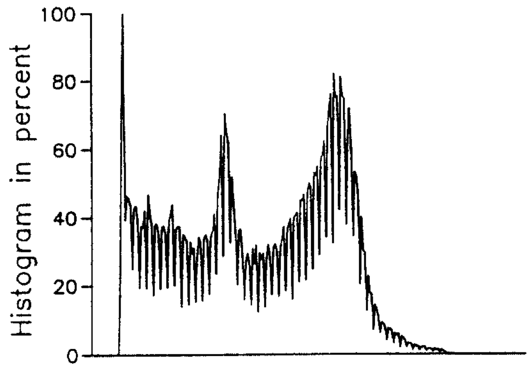
www.dbpia.co.kr

fail the test. However, the obtainable cost in worst cases is equal to that of ESVQ. Second, abandoning the conventional VQ structure, there have been researches to find new suboptimal algorithms[1] such as tree search VQ, multi-stage VQ, etc. Few works have been done for the former in contrast to the latter. This work is more related to the former approach, but different in that the proposed algorithm uses an important property of images instead of the mathematical manipulation.

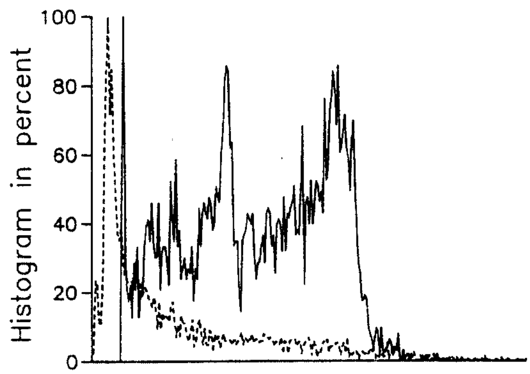
II. ALGORITHM

For a test image SMPTE1 (See Fig.4.) which has the histogram as shown in Fig.1(a), after means and standard deviations are computed for every blocks of size 8x8, their histograms are shown in Fig.1(b). X-axis cover maximum mean and standard deviation values and Y-axis represents relative occurrences of each X-axis values. Note from Fig.1(b) that mean values are more evenly distributed in wide region than standard deviation values. Additional experiments have shown that this property holds for any sizes of blocks and almost all images. Therefore, one can conclude that one of the most important features of a local block in image is the block mean.

The above observation spontaneously leads to the following algorithm. Proposed is a simple modification to the minimum-distortion search rule, which can reduce the search cost to one eighth or less with a negligible loss in performance. The idea stems from a sliding operation of a codebook as in Fig.2. A small window is sliding over the given codebook, a search window is determined using the feature of a currently encoding block or vector, and then exhaustive search is performed within the selected codebook only. Hereafter the given codebook will be called as super codebook, and a smaller codebook within



(a) Histogram of pel intensity



(b) Histograms of means and standard deviations of 8x8 blocks (Solid line: mean, dashed line : standard deviation)

Fig. 1 Histograms of a test image SMPTE1

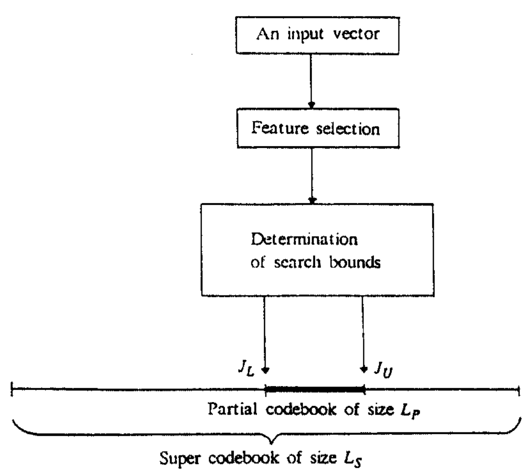


Fig. 2 Partial codebook selection method

www.dbpia.co.kr

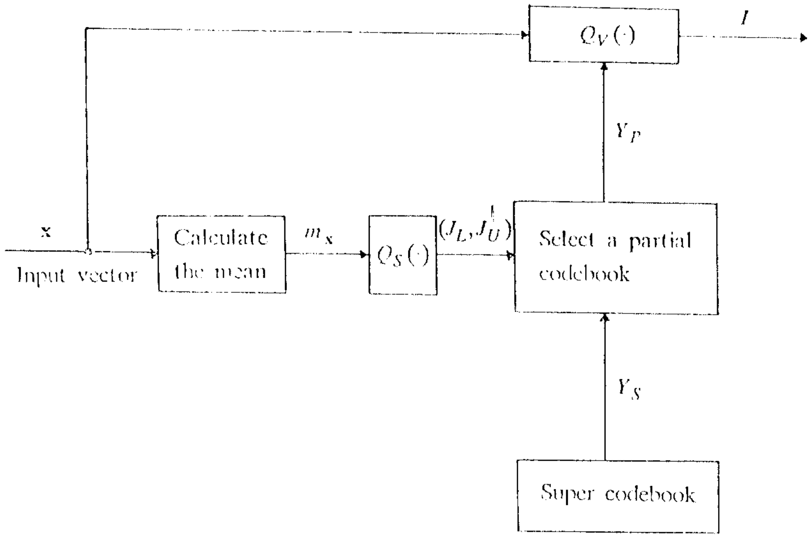


Fig. 3 Block diagram of sliding search VQ (SSVQ)

the selected window as partial codebook. Hence, the algorithm will be called as sliding search VQ (SSVQ) after the sliding feature of the algorithm.

The proposed algorithm for sliding search is illustrated in Fig.3. Given an input block or vector, a feature is first extracted from the vector. In this work, the block mean is used as the feature due to the reason discussed above. Simulation results also show a more satisfactory performance about 0.5 dB better in SNR when the mean feature is used rather than the standard deviation. The SSVQ algorithm is summarized in two parts as follows.

(b) Original image SMPTE2 for coding

(a) Original image SMPTE1 for design

(c) L_p=8, SNR=30.2, P_s= 73.0

www.dbpia.co.kr

(d) L_p=16, SNR=31.8, P_s= 92.9

(e) L_p=32, SNR=32.3, P_s= 98.9

(f) L_p=64, SNR=32.4, P_s=100.0

Fig. 4 Test and coded images by SSVQ (SMPTE1 design/SMPTE2 coding)

Codebook Design/Rearrangement Stage

(1) Design a super codebook of size L_S and vector dimension N using such an algorithm as generalized Lloyd algorithm[3].

(2) Rearrange code vectors in the super codebook such that their component means have magnitudes of increasing order, i.e., $Y_S = \{y_i,\ i = 1, 2, \cdots, L_S \mid m_i \le m_{i+1}\}$, where m_i is defined as the component mean of the i-th vector $y_i = [y_{i1}, y_{i2}, \cdots, y_{iN}]^t$, and is given by $m_i \equiv \frac{1}{N}\sum_{k=1}^{N} y_{ik}$

(3) Store the rearranged super codebook Y_S and the mean codebook $Y_m = \{m_i, i = 1, 2, \cdots, L_S\}$. In consequence, Y_m becomes increasingly ordered.

Encoding Stage

(1) Input a vector $x = [x_1, x_2, \cdots x_N]^t$, and then compute its component mean as $m_x = \frac{1}{N}\sum_{k=1}^{N} x_k$.

(2) Find an index J of the most similar value to m_x from Y_m. This is just a scalar quantization as $J = Q_S(m_x) \mid \mathrm{Y_m}$.

(3) Determine the search bound (J_L, J_U) of the partial codebook $Y_P = \{y_i,\ i = J_L, J_L + 1, \cdots, J_U\}$ of size $L_P (= J_U - J_L + 1)$ using following equations.

$$J_L = J - L_P/2$$

$$\text{If } J_L < 1, \text{ then } J_L = 1 \text{ and } J_U = L_P \quad (1)$$

$$J_U = J + L_P/2 - 1$$

$$\text{If } J_U > L_S, \text{ then } J_L = L_S - L_P + 1 \text{ and } J_U = L_S \quad (2)$$

Eq. (1) and (2) can be accomplished simultaneously with the step (2) above using two scalar quantization respectively as $J_L = Q_{S1}(m_x) \mid \mathrm{Y_m}$ and $J_U = Q_{S2}(m_x) \mid \mathrm{Y_m}$.

www.dbpia.co.kr

(4) Fully search the partial codebook Y_P, and find a nearly optimal code vector of index $I=Q_V(x)$ Y_P

The encoding stage is illustrated in a block diagram of Fig.3. Since the decoding stage is the same as in the ordinary VQ, one just need to reconstruct the I-th code vector by table lookup of the super codebook.

It is important to remind that the algorithm uses the whole structure of ESVQ intact. It should also be noted that the proposed algorithm can reduce the search complexity, but can not reduce the bit rate, since the transmission index should cover indices from 1 to L_S rather than from 1 to L_P. Therefore, the method is called as forward SSVQ since the transmission bit rate includes the positioning address of the partial codebook.

The SSVQ is conceivable as the generalized algorithm of the classified VQ (CVQ). In CVQ, several class codebooks instead of a single super codebook are prepared in order to well represent all specific edge classes. When encoding a vector, one should first select a proper class codebook, and then fully search the codebook. In SSVQ, given a super codebook of size L_S, every time a vector is encoded, a window of size L_P is sliding over the super codebook. According to the feature of the vector, a partial codebook is easily selected, and then the partial codebook is exhaustively searched. Hence, it can be conceived that there are L_S-L_P+1 class codebooks in SSVQ, which is generally larger than the number of class codebooks in CVQ. In theoretical point of view by the concept of VQ, it is more efficient to code both the class index and the code vector index rather than to code them separately as in CVQ.

III. EXPERIMENTAL RESULTS AND DISCUSSION

As test images, SMPTE1 and SMPTE2 of size 480x512 are used. Two super codebooks are designed from the images respectively using the generalized Lloyd algorithm[3]. The two images are coded using the codebooks in four different combinations, i.e., SMPTE1/SMPTE1, SMPTE2/SMPTE2, SMPTE1/SMPTE2, and SMPTE2/SMPTE1, where those before the slashes denote images used for codebook design, and those after the slashes denote images used for coding. The latter two combinations represent cases of quantizer mismatch. Vector dimension is fixed to $N=16$ by vectorizing a 4x4 image block, and the super codebook size is also fixed to L_S =256; Hence the bit rate is $R=(\log_2 L_S)/N=0.5$ bits/pel. The partial codebook size L_P is made to vary 8, 16, 32, 64, 128, and 256. The case when $L_P=L_S$ corresponds to just an ESVQ.

Objective performances are measured by SNR given by

$$\mathrm{SNR} \equiv 10\log_{10}\frac{255^2}{E_T} \quad [\mathrm{dB}] \tag{3}$$

where E_T is the total reconstruction error power, and search accuracy that is defined by

$$P_S=\frac{K_S}{K_T}\times 100 \quad [\%] \tag{4}$$

where K_T is the total number of input vectors, and K_S is the number of optimum vectors that can be sought by SSVQ.

Table 1 shows performances of SSVQ and tree search VQ (TSVQ). It should be noted with regard to SNR and storage requirement that SSVQ generally outperforms binary-tree or hexa-tree TSVQ of equal search complexity. In hexa tree, a node has 16 branches. Note also that for

www.dbpia.co.kr

Table 1. Performances of SSVQ and tree search VQ

Deign image / Coding image	Partial codebook size of SSVQ						TSVQ	
	8	16	32	64	128	256	binary	hexa
SMPTE1/SMPTE1	28.3	30.1	31.2	31.5	31.5	31.5	30.3	31.1
SMPTE2/SMPTE2	31.6	32.9	33.6	33.7	33.7	33.7	32.7	33.3
SMPTE1/SMPTE2	30.2	31.8	32.3	32.4	32.4	32.4	31.5	32.3
SMPTE2/SMPTE1	27.9	28.9	29.7	29.9	29.9	29.9	28.9	29.5
Search Complexity	1/32	1/16	1/8	1/4	1/2	1	1/16	1/8
Memory Requirement	1	1	1	1	1	1	2	16/15

Table 2. Search accuracies of SSVQ

Design image / Coding image	Partial codebook size of SSVQ					
	8	16	32	64	128	256
SMPTE1/SMPTE1	71.9	90.8	98.4	100.0	100.0	100.0
SMPTE2/SMPTE2	76.7	95.1	99.5	100.0	000.0	100.0
SMPTE1/SMPTE2	73.0	92.9	98.9	100.0	100.0	100.0
SMPTE2/SMPTE1	71.2	90.2	97.9	99.9	100.0	100.0

the two cases of quantizer mismatches, SSVQ shows good performance behavior without severe degradations. Giving attention only to the SSVQ, one can see that only 64($= L_P$) out of 256 ($=L_S$) codeword searches are sufficient to have the performance of ESVQ. SSVQ with 1/8 to 1/16 search complexity can produce the efficient search capability with a small performance degradations. Search accuracies in Table 2 well support SNR performances in Table 1. It can be seen that SSVQ of 1/4 search complexity can search optimum codewords at almost 100 % accuracy. Comparing the test and coded images for the case of SMPTE1 design and SMPTE2 coding, in Fig.4 the degradation is hardly visible when L_P =32. This result indirectly shows that block mean or vector component mean can be a very good selection featue of partial codebooks. However, the possibility of selecting erroneous partial codebooks causes a small performance degradation.

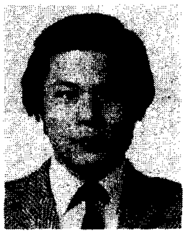
IV. CONCLUSION

In this work, we have proposed a fast codebook search algorithm for vector quantization in image coding. Utilizing the block mean as a selection criterion of a smaller codebook, partial codebook, which is sliding over and selected from a super codebook, sliding search VQ (SSVQ) algorithm has been proposed and shown to reduce the search complexity to 1/16 - 1/8 with a slight performance degradation. The algorithm has also been shown to have smaller search complexity than tree search VQ.

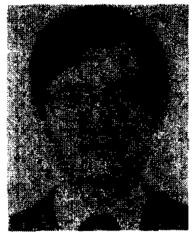
REFERENCES

1. Gray, R.M., 'Vector quantization', IEEE ASSP Magazine, Vol.1, No.2, pp.4-29, April 1984.
2. Soleymani, M.R. and Morgera, S.D., 'An efficient nearest search method', IEEE Trans., Vol.COM-35, pp.677-679, June 1987.
3. Linde, Y., Buzo, A., and Gray, R.M., 'An algorithm for vector quantizer design', IEEE Trans., COM-28, pp.84-95, January 1980.

www.dbpia.co.kr



高 鍾 錫(Jong Seog KOH) 正會員
1959年 8月 7日生
1982年 2月：高麗大學校 電子工學科 卒業(工學學士)
1984年 2月：韓國科學技術院 電氣및 電子工學科 卒業(工學碩士)
1984年 3月～現在：韓國科學技術院 電氣및 電子工學科 博士過程



金 在 均(Jae Kyoon KIM) 正會員
1938年 9月17日生
1958. 4～1962. 2：韓國航空大學 應用電子科(工學士)
1962. 3～1967. 2：서울大學校 大學院 電子工學科(工學碩士)
1967. 9～1971. 8：美國남가주大學校大學院 電氣工學科(工學博士)
1962. 4～1966. 7：空軍服務 中尉
1967. 9～1972. 3：美國 남가주 大學校(研究院)
1972. 4～1973. 3：美國 NASA GSFC(研究院)
1973. 4～現在：韓國科學技術院院 教授
1987. ～現在：本 學會 副會長
1984. 1～1985. 6：科學技術處 電氣·電子조정관

www.dbpia.co.kr