論文 94-19-4-11

벡터 양자화의 고속 부호화 알고리즘

正會員 蔡 鍾 吉* 正會員 黃 燦 植*

Fast VQ Encoding Algorithm

Jong Kil Chae*, Chan Sik Hwang* *Regular Members*

要 約

벡터 양자화의 부호화에서 입력벡터에 가장 잘 정합되는 코드벡터를 탐색하는 과정에서 발생하는 계산의 복잡도는 코드북의 크기에 비례하여 지수적으로 증가하고 실질적으로 응용을 제한한다. 본 논문에서는 정합 가능성이 없는 코드백터에 대한 왜곡의 계산을 제거하기 위한 조건의 시작 벡터로서 참조 벡터를 사용하는 단순, 고속의 효율적인 벡터 양자화의 부호화 알고리즘을 제안하였다. 이는 입력벡터에 정합 가능성을 갖는 참조 벡터를 선택하고 코드 벡터에 대한 왜곡의 계산을 제거하기 위한 조건을 결합하는 것이다. 제한된 방법은 전탐색 벡터 양자화에 비하여 단지 10~15%의 수학적 연산을 필요로 한다. 그리고 덧셈과 비교 연산의 수는 크게 줄어 들지 않지만 곱셈은 벡터 양자화의 여러 고속부호화 방법의 70~80%까지 들었다.

ABSTRACT

A problem associated with vector quantization(VQ) is the computational complexity incurred in searching for a codevector with the closest to a given input vector, where the complexity increases exponentionally with proportion to codebook size and then limits practical application. In this paper, a simple and fast, but efficient, VQ encoding algorithm is presented using a reference codevector as start codevector of premature exit condition, which eliminates distance claculation of unlikely codevectors. The algorithm is to find reference codevector having the possibility to be the nearest vector to input vector first and then to incorporate premature exit condition. The proposed algorithm needs only 10~15% of mathematical operations compared with the conventional full search VQ. Although the number of additions and comparsions of the proposed algorithm is not reduced greatly, the number of multiplication is reduced up to 70~80% compared with other fast VQ encoding methods.

*慶北大學校 電子工學科
Dept. of Electronics, Kyungpook Nat'l Univ.
論文番號 : 93228
接受日字 : 1993年 12月 1日

I. Introduction

Data compression using VQ has received a great interest because of tis superior rate-distortion performance over scalar quantization and simple table look-up decoding[1, 2]. In the VQ, a signal is first divided into subsequent blocks of data with size k. The block of data is then scanned to form a k-dimensional vectro. A codebook of representative vectors chosen as training sequence. The process of VQ is searching for a codevectro that is the closest matching to input vector for a given codebook and representing the codevectro using the index achieving data compression over input vector. The signal is restored by a simple table look-up method using the index as the address to a table containing the codevectors at the decoder[1, 2].

It is limited to apply VQ to pratical voice coding, image coding and pattern recognition because the computation time for searching a codevector having minimum distortion and the memory space of codebook increase exponentially with the vector dimension and rate. To quantize a given input vector, it is required to find its distance from $N(=2^{KR})$ codevectors, and then compare the distance to find the closest codevector. Therefore, quantization of each input vector requires 2^{KR} distance computations with k multiplications, k subtractions and k-1 additions in the case of squared error distortion measure, and 2^{KR}-1 comparisons, where k is vector dimension and R is bits /sample [3-9].

Technique to circumvent this problem can be classified into two groups. One is, simpler yet, suboptimal variant of the full search VQ and then sacrifies performance, including the tree searched VQ, multistage VQ and lattice VQ etc. [1, 2]. The other devises fast searching algorithm with maintaining the same performance of the full search VQ [3, 9]. Bei and Gray [3] developed a partial distance search(PDS) algorithm which allows early termination of the distortion calculation by incorporation a premature exit condition in the search poocedure. Paliwal and Ramasubramanian [4] redeced computation time of PDS algorithm by sorting the codebook according to the order of intervector transition probability of the codevectors. Ra and Kim [5] introduced another premature exit condition which is based on the distance between mean of input vector components and mean of codevector components. The triangular inequality elimination(TIE) is utilized to reject distortion calculation of several unlikely codevectors by Orchard [6]. Huang, et al. [7] developed an algorithm combining the TIE and distance in the hypersphere to eliminate codevectors more than those by Orchard [6]. Soleymani and Morgera [8] developed an abortion condition which is based on the absolute error associated with the element of codevector. These methods maintain the same fidelity of the full search VQ [3-9].

In this paper, a fast codebook search algorithm is proposed, which is to find reference codevector having the possiblility to be the nearest vector to input vector first and then to incorporate premature exit condition by Soleymani's [8] to eliminate distance calculation of unlikely codevector. The reference codevector is used as start codevector for premature exit condition by Soleymani's [8].

As a result, the proposed algorithm can save up to 85-90% of computation time compared with the full search VQ. Since the algorithm is able to reduce multiplications, it can search minimum distortion codevector much faster than any other recently proposed methods.

II. Fast VQ encoding algorithm

VQ is a procedure by which vectors of k-dimension resulted from a given data source are approximated by codevectors from a set of codevectors commonly called a codebook. Let

$\mathbb{Y}=\{\mathbb{y}_i, i=1, \cdots . N\}$ be codebook of size N, where $\mathbb{y}_i=(y_{i1}, y_{i2} \cdots , y_{ik})$ is a k-dimensional codevector in space $\mathbb{R}^k$. For a given input vector $\mathbb{x}_i=(x_1, x_2 \cdots , x_k)$ in $\mathbb{R}^k$ space, vector quantization $Q(\mathbb{x})$ is applied in $\mathbb{R}^k$ space and defined as

$$Q(\mathbb{x})=\arg\min d(\mathbb{x}, \mathbb{y}_i) \quad (1)$$

where $d(\mathbb{x}, \mathbb{y}_i)$ is distance between input vector $\mathbb{x}$ and codevector $\mathbb{y}_i$ and arg denotes the function that means the variable minimizing $d(\mathbb{x}, \mathbb{y}_i)$. In order to find $Q(\mathbb{x})$, the basic VQ encoding uses and exhaustive full search method that calculates distances between a input vector and codevector in the codebook. In the algorithm proposed here, an simple, but efficient, test is performed before computing the distortion for each code vector in the same way of Soleymani's [8]. The test is composed of comparing absolute error associated with each element of codevector $\mathbb{y}_j$ as defined in Eq. (2), $|x_n-y_{jn}|$, to the square root of squared error distortion between input vector $\mathbb{x}$ and codevector $\mathbb{y}_i$ with minimum distortion to input vector thus far, where x_n and y_{jn} denote the n th component of input vector $\mathbb{x}$ and codevector $\mathbb{y}_j$ respectively.

$$|x_n-y_{in}| \leqq [d(\mathbb{x}, \mathbb{y}_i)]^{1/2} \quad n=1, 2, \cdots , k \quad (2)$$

If codevector $\mathbb{y}_j$ satisfying Eq. (2) is found, codevector $\mathbb{y}_j$ is changed to $\mathbb{y}_i$ which is minimum distortion codevector found thus far. The test is equivalent to first checking if a given codevector is inside a hypercube enscribing the hypersphere, centered around the input vector $\mathbb{x}$, with radius equal to the squared root of minimum distortion found thus far. This has been used in a different way in the minimax method of Cheng and Gersho [9].

If we use codevector having the possibility of minimum distortion to input vector as referene codevector $\mathbb{y}_i$, the number of codevectors satisfying Eq. (2) will not be large and then a drastic reduction of mathematical operation can be obtained as shown Fig. 1 of dimensional case.

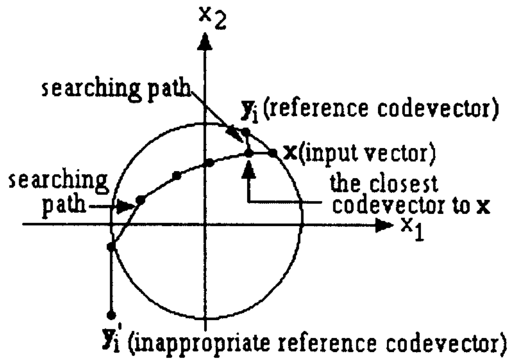

Fig 1. Searching path according to reference codevector

We now introdece a method to choose reference codevector $\mathbb{y}_i$ having the possibility to be minimum distortion vector to input vector. From the fact that power of codevector with the closest matching to input vector is nearly equal to that of input vector, we choose M codevectors with $d(\mathbb{y}_i, \mathbb{O})$ which are nearly equal to $d(\mathbb{x}, \mathbb{O})$ first where $\mathbb{O}$ is origin vector having 0 for every component of vector. If codevector with minimum difference between $d(\mathbb{y}_i, \mathbb{O})$ and $d(\mathbb{x}, \mathbb{O})$ is chosen as reference codevector, it is frequently opposite to input vector $\mathbb{x}$ for origin vector $\mathbb{O}$. Then, we propose mini-max method(MMM) or maximum normalized cross correlation method (MMCM) defined as Eq. (3) or (4), and apply it to choose a reference codevector $\mathbb{y}_i$.

$$\mathbb{y}_i=\underset{i=1,\ldots,M}{\mathrm{Min}}\left[\underset{n=1,\ldots,k}{\mathrm{Max}}(|x_n-y_{in}|)\right] \quad (3)$$

$$\mathbb{y}_i=\mathrm{Min}\left[\sum_{n=1}^{k}[x_n-y_{in}]/d(\mathbb{x}, \mathbb{O})\, d(\mathbb{y}, \mathbb{O})\right] \quad (4)$$

where M is the number of codevectors with $d(\mathbb{y}, \mathbb{O})$ nearly equal to $d(\mathbb{x}, \mathbb{O})$ and k is the vector dimension.

We now provide an outline of the proposed algoritm :

1) For a given codebook Y={$\mathbb{y}_i$ i=1, ⋯ , N}, determine d($\mathbb{y}$, $\mathbb{O}$) for each codevector.
2) Sort d($\mathbb{y}_i$, $\mathbb{O}$)'s by the ascending or descending order and store the sorted d($\mathbb{y}_i$, $\mathbb{O}$)'s in the momory.
3) Compute d($\mathbb{x}$, $\mathbb{O}$) of input vector $\mathbb{x}$ and origin vector $\mathbb{O}$.
4) Choose M chdevectors with d($\mathbb{y}_i$, $\mathbb{O}$) nearly equal to d($\mathbb{x}$, $\mathbb{O}$)
5) Choose $\mathbb{y}_i$ from the chosem M codevectors wity the MMM or the MNCCM that is defined in Eq. (3) or (4)
6) Calculate d($\mathbb{x}$, $\mathbb{y}_i$)
7) Check Eq. (2), if it is satisfied for every component of codevector $\mathbb{y}_j$, change $\mathbb{y}_j$ to $\mathbb{y}_i$ and return to step 6). Otherwise the current $\mathbb{y}_i$ is the closest codevector to input vector $\mathbb{x}$.

Eliminating the need for computing of distortions of the rejected codevectors by Eq. (2), this results in a drastic reduction in the number of multiplications. The increase in the number of addition and comparisons is moderate because of finding the reference codevector and testing by Eq. (2) For those codevectors which fail the test by Eq. (2), we can sav k multiplications, k-1 additions, and the remanining subtractions. Denoting the average number of rejected codevectors by N', the average number of ruquired multiplications perpel is

$$N_m = N - N' + O_m \tag{5}$$

And the average number of additions per pel is

$$N_a = (1 - 1/k)(N - N) + O_a \tag{6}$$

The average number of subtractions per pel is

$$N_s = (N - N') + N' \times k'/k + O_s \tag{7}$$

where k′ is the average number of components checked before a codevector is rejected. Since we perform one comparison after each subtraction and one after each k multiplications, the average number of comparisons per pel is

$$N_c = N_s + \frac{N_m}{k} + O_c \tag{8}$$

The overhead operations to choose reference codevector $\mathbb{y}_i$ is O_m, O_a, O_s and O_c which is incurred in the choice of M codevector from entire codebook and in finding the reference codevector by Eq. (3) or (4). The square root introduced in Eq. (2) is not a frequent operation that does not have a significant effect on the overall complexity.

Ⅲ. Simulation results

To evaluate the proposed algorithm two different codebook are used, each one has 512 and 1024 codevectors. The codebook is generated by the well known LBG algorithm [1] and vector dimension is 16 of 4X4 pixels in spatial domain. Training set data consists of 8 standard images of size 512X512 with 8 bit resolution. Test image is 'Lenna' of size 512X512 with 8 bit resolution.

The table 1, 2 show the average number of mathematical operations required per pel for various fast VQ encoding algorithms. The opreations include all the overhead computation required to search the minimum distortion codevector to input vector. For an appropiate choice of reference codevector having the possibility to be minimum distortion vector to input vector, M is 5 in our simulation.

Table 1 and 2 of spatial domain VQ are evaluated and discussed. Huang's [7] algorithm not including the PDS is worse than Soleyamani's [8] regardless of codebook size. Huang's algorithm including the PDS is similar to or worse than Soleymani's according to codebook size. But the proposed algoritm can reduce the number of multiplications up to 70~80% compared with Soleymani's and Huang's using PDS regardless of the MMM and the MNVVM. Addition and comparison of the proposed algorithm is similar to of Soleymani's and Huang's using the PDS because of finding the reference codevector and testing to eliminate distance calculation.

Table 1. Computational complexity required per pel of VQ, codebook size=512

Algorithm		mult.	add (+&-)	comare
Bei[3]		67.1	102.2	66.1
Soleymani [8]		41.8	168.4	131.8
Huang[7]		59.5	191.5	144.3
Huang[7] &PDS[3]		41.1	154.7	125.9
PROPOSE	MMM	30.8	167.5	139.7
	MNCCM	35.8	170.4	138.4

Table 2. Computational complexity required per pel of VQ, codebook size=1024

Algorithm		mult.	add (+&-)	comare
Bei[3]		119.3	174.6	119.1
Soleymani [8]		58.5	269.6	218.4
Huang[7]		114.3	375.1	286.3
Huang[7] &PDS[3]		76.6	299.6	248.6
PROPOSE	MMM	46.9	298.2	255.9
	MNCCM	51.8	300.8	254.1

Summarizing the results, the proposed algorithm can always reduce the number of multiplication up to 70~80% compared with Soleymani's regardless of the MMM and the the MNCCM. And addition and comparison of the proposed algorithm are generally similar to hose of Soleymani's. The MMM is better than the MNCCM because most operations for the MMM are addition and comparison, those for the MNCCM are addition and multiplication.

Ⅳ. Conclusion

A simple and fast, but efficient, VQ encoding algorithm is porposed, which eliminates distance calculation of unlikely codevectors. In the porposed algorithm, reference codevector having the possibility to be the nearest vector to input vector is found first and then applied to premature exit condition. Eliminating the need to compute the distortion for almost all codevectors, the algorithm can reduce the number of multiplications up to 70~80% compared with other fast VQ encoding methods. But the increase in the number of addition and comparisons is moderate because of finding the reference codevector and testing to eliminate distance calculation. The algorithm requries 10~15% of mathematical operations compared with the full search VQ. The performance of proposed algorithm is somewhat dependent on images and actual codevectors in the same way what that of other fast VQ encoding algorithm.

References

1. A. Gersho and R. M. Gray, *Vector quantization and signal processing*, Kluwer Academc publishers, Norwell, MA. 1991
2. H. Abut, *Vector quantization*, IEEE press, Newyork, NY. 1990
3. C. D. Bei and R. M. Gray, "An improvement of the minimum distortion encoding algorithm for vector quantization", *IEEE* trans. on commun vol. COM 33, No. 10, pp. 1132 1133, Oct. 1985
4. K. K. Paliwal and V. Ramasubramanian, Effect of ordering the codebook on the efficiency of the partial distance search algorithm for vector quantization , *IEEE trans. on commun.*

vol. COM-37, No. 5, pp. 538~540, May. 1989
5. S. W.Ra and J. K. Kim "Fast weight-ordered search algorithm for image vector quantization", *IEE Electronics Letters*, pp 2081~2083, Oct. 1991
6. M. Y. Orchard, "A fast nearest neighbor search algorithm", *Proc. IEEE Int. Conf., Acoust., Speech, Signal, Processing*, vol. 4, pp. 2297~2300, May. 1991
7. C. M. Huang, Q. Bi, G. S. Stiles and R. W. Haris, "Fast full search equivalent encoding algoritms for image compression using vector quantization", *IEEE trans. on Image Processing*, Vol 1, No. 3, pp. 413~416, Jul. 1992
8. M. R. Soleymani and S. D. Morgera, "An efficient nearest neighbor search method", *IEEE trans. on commun.* vol. COM-35, No. 6, pp. 677~679, Jun. 1985
9. D. Y. Cheng, A. Gersho, B. Ramamurthi and Y. Soham, "Fast search algorithms for vector quantization and pattern matching", *Proc. IEEE Int. Conf., Acoust., Speech, Signal, Procssing*, vol. 1, pp. 9.11.1~9.11.4, Mar. 1985

蔡 鍾 吉(Jong Kil Chae) 정회원
경북대학교 전자공학과 박사과정 수료
제18권 11호 참조.

黃 燦 植(Chan Sik Hwang) 정회원
경북대학교 전자공학과 교수
제18권 5호 참조.