2000년 한국정보처리학회 추계 학술발표논문집 제7권 제2호

서브밴드 영상의 VQ를 위한 고속 코드북 탐색 알고리즘

황재호*, 홍충선**, 이대영**
*극동정보대학 멀티미디어과
**경희대학교 전자정보학부
e-mail : mjrin@unitel.co.kr

Fast Codebook Search Algorithm for VQ of Subband Images

Jae-Ho Hwang*, Choong-Seon Hong**, Dae-Young Lee**
*Dept. of Multimedia, Keuk-Dong College
**Dept. of Electronic Engineering, Kyung-Hee University

Abstract

Two fast search algorithms are proposed for VQ encoding in subband/VQ coding schemes. These algorithms exploit the property of the transform domain that the large coefficients rarely exist in the decomposed subbands. And the exit condition of PDE algorithm can be satisfied by comparing the large values of the codeword with the corresponding ones of the input vector. The computational complexity can be reduced at the expense of memories without extra coding errors.

1. Introduction

Vector quantization (VQ) has been widely used for image coding applications. Given an input vector, finding its closest codeword in a codebook using an exhaustive search algorithm (ESA) is a time-consuming job. Many fast search algorithms have been proposed to reduce the complexity of a codebook search. One of the most well-known algorithms is the partial distance elimination (PDE) [1] which uses a premature exit condition instead of completing the calculation of the distortion between the input vector and each codeword. Several other methods [2][3] based on the triangle inequality elimination (TIE) reduce the search space by using the mathematical inequality at the expense of memories.

Meanwhile, in much literature, vector quantization has been employed for their subband image coding schemes [4]. In this letter, we propose two fast search algorithms that are appropriate for subband/VQ coding schemes. These algorithms are based on the fact that the premature exit condition in PDE can be satisfied in the early stages if the large distortion is generated as fast as possible during codeword matching. The probability of the large distortion will be increased if the large elements in each codeword are first matched with their corresponding elements in the input vector. This is achievable because just a few large coefficients exist within both the codewords and the input vectors in the transform domain. This is the key idea of our methods.

2. Algorithm I

Let $X^{m,n}=\{ x_j, j = 1, \ldots, K \}$ be an input vector in a subband $S_{m,n}$ (m : resolution level, n : orientation) where x_j is a transformed coefficient and K is a vector dimension. And let $C^{m,n} = \{ c_i, i = 1, \ldots, N \}$ be a sub-codebook of $S_{m,n}$ where N is a codebook size and $c_i = \{ c_{ij}, j = 1, \ldots, K \}$ is a codeword. Given the input vector $X^{m,n}$, the ESA searches for the codeword $c_i = Q(X^{m,n})$ which minimizes the squared Euclidean distance D($X^{m,n}$, c_i) over the whole sub-codebook $C^{m,n}$.

In multiresolution analysis, there exist three detail-bands of a horizontal (H), a vertical (V), and a diagonal (D) orientation for each resolution level and a smooth-band. Therefore, in the case of decomposition level M, M x 3 sub-codebooks (mainly vector) for detail-bands and a scalar codebook for a smooth-band are required for subband/VQ so that the total number of the needed sub-codebooks (called a multiresolution codebook) is M x 3 + 1.

The smooth-band image can be scalar-quantized with eight comparators and a scalar codebook by successive approximation [5]. The detail-band images are usually vector-quantized with their own vector codebooks. In the detail-bands, the energies (or magnitudes) of most coefficients are near zero or small. Conversely, the coefficients having large energy are very seldom. In codeword matching, the probability of the large distortion between the elements in each codeword and their corresponding elements in the input vector will be increased, if the elements which have large energy in the codeword are first selected. As a result, the premature exit condition in the PDE can be satisfied in the early stages.

In order to store the order information of the elements to be matched in each codeword, we employ a 2-dimensional reference table $T^{m,n} = \{ t_i, i = 1, \ldots, N \}$ for each sub-codebook $C^{m,n}$ where $t_i = \{ t_{ij}, j = 1, \ldots, K \}$. The element t_{ij} points the location of the element c_{ij} in the codeword c_i indexed in a descending order according to the energies of the elements. The energy of the jth element c_{ij} in the ith codeword c_i is defined as the squared value c_{ij}^2. For example, let's assume that K = 2 x 2, X = (2.1, 3.0, 0.8, -1.2), c_i = (1.3, -5.1, 0.3, 10.0), and $d_{\min}$ = 80.5. When the PDE is applied for codeword matching, the distortion until j = 3, $(2.1-1.3)^2 + (3.0-(-5.1))^2 + (0.8-0.3)^2$ = 66.5 does not exceed the current distortion $d_{\min}$ so that the calculation continues until j = 4. In the case of our algorithms, t_i becomes (4, 2, 1, 3) from preprocessing because of $c_{i4}^2 > c_{i2}^2 > c_{i1}^2 > c_{i3}^2$, so that the distortion in case of j = 4 is first calculated. Because the distortion, $(-1.2-10.0)^2$ = 125.44 exceeds $d_{\min}$, the rest of the calculation does not need to be performed. The 75 % ((4-1)/4) of the multiplication, the 85 % ((7-1)/7) of the addition, and the 75 % of the comparison are saved in comparison with the case of PDE. Even though this is the best case, statistically, large amount of resources will be saved.

Algorithm I (decomposition level: M) can be described as follows :

(1) *Preprocessing:* Re-order each sub-codebook in an ascending order in terms of the norm of the codeword $\|c_i\|$. From each re-ordered sub-codebook $\hat{C}^{m,n}$ (m = 1,

2 , ... , M, n = 1, 2, 3), obtain the first row of the reference table t_1 by indexing the codeword c_1 in a descending order according to the energy c_{1j}^2 (j = 1, 2, ... , K). Next, get t_2 from c_2, then t_3 from c_3, ..., t_N from c_N. From this procedure, each 2-D reference table $T^{m,n}$ (dimension : N x K) is generated.

(2) *On-line processing:*

(i) Set $i = 1, j = 1$, and $d_{\min} = \infty$.

(ii) Set $d_c = 0$.

(iii) Calculate the distortion d_{ij} between $x_{(t_{ij})}$ and $c_{i(t_{ij})}$. Set $d_c = d_c + d_{ij}$.

(iv) If j < K, then go to (v). Otherwise, if $d_c < d_{\min}$, then update $d_{\min} = d_c$, $Q(X^{m,n}) = i$, get next i. If i > N, go to (vi). Otherwise go to (ii).

(v) If $d_c > d_{\min}$, then get next i, and go to (ii). Otherwise, get next j, and go to (iii).

(vi) The codeword $c_i = Q(X^{m,n})$ for the input vector $X^{m,n}$ is now decided.

The above process continues until the end of the input vector in each subbband $S_{m,n}$ is reached.

3. Algorithm II

The disadvantage of Algorithm I is that a large amount of memory is required to store the 2-dimensional reference table (N x K). In each subband, the memory size for the table is N x K x $\log_2 K$ bits. Algorithm II, however, uses just a small portion of the table to save the memory space at the cost of using a little more resources.

In Algorithm II, if we use two items of each row in the table, the two largest elements in each codeword are matched with the reference table whose dimension is N x 2 and the rest of the codeword elements are matched in the order of the codeword itself. In this case, the memory space for each subband is N x 2 x $\log_2 K$ bits. If K is 16, then the memory space is reduced to 1/8 compared with Algorithm I.

4. Experiments

We performed some simulations to examine the efficiency of the proposed fast encoding algorithms. The well known LBG algorithm [6] is used for generating a multiresolution codebook with ten test images (512 x 512, 256 gray levels) as a training set. The performance evaluation of our algorithms is conducted with a 'Lena' image (512 x 512, 256 gray levels) outside the training set. All images are decomposed into three resolution levels with a 9/7-tab biorthonormal wavelet filter [7]. Our algorithms are evaluated with other methods by the average number of resources required per pixel.

In Table 1, the vector dimensions of all detail-bands are fixed to 4 x 4 and the coding errors are the same for all schemes (PSNR = 28.24 dB at 0.6 bpp). The proposed algorithms largely outperform PDE and TIE (Algorithm II in [2]. The experiment conducted in Table 2 is similar to Table 1, but the dimensions of level-3, level-2, and level-1 are 2 x 2, 4 x 4, and 8 x 8, respectively. This is a case in which high vector dimensions are often used for high resolution bands in subband/VQ coding schemes. In this case, the coding error is 30.47 dB at 0.4 bpp. In the tables, the dimension of the reference table for Algorithm II is N x 2.

5. Conclusion

We have presented the fast search algorithms for VQ encoding in subband/VQ schemes which are especially efficient to the high vector dimension. From the experimental results, we see that the proposed

Table 1. Average number of resources required per pixel for various fast algorithms (test image : 'Lena', N = 256, and K = 4 x 4 for all detail-bands)

Algorithm	x	+/–	compares	$\sqrt{\ }$
ESA	256.0	496.0	16.0	–
PDE	61.6	107.3	61.6	–
TIE	52.5	100.7	10.0	0.3
Algorithm I	31.6	47.2	31.6	–
Algorithm II	33.4	50.9	33.4	–

* The coding error is 28.24 dB at 0.6 bpp (the smooth-band is scalar quantized by 256 step sizes).

Table 2. Average number of resources required per pixel for various fast algorithms (test image : 'Lena', N = 256, and K = 2 x 2, 4 x 4, and 8 x 8 for level-3, level-2, and level-1, respectively)

Algorithm	x	+/–	compares	$\sqrt{\ }$
ESA	256.0	502.8	9.1	–
PDE	50.4	91.6	50.4	–
TIE	48.3	94:2	4.7	0.2
Algorithm I	25.8	42.5	25.8	–
Algorithm II	30.5	52.0	30.5	–

* The coding error is 30.47 dB at 0.4 bpp (the smooth-band is scalar quantized by 256 step sizes).

algorithms outperform PDE and TIE. In addition, the search process is much simpler than TIE. And our algorithms can be integrated with any TIE algorithms in order to further reduce the resources.

References

1 BEI, C.D. and GRAY, R.M.: 'An improvement of the minimum distortion encoding algorithm for vector quantization', IEEE Trans, 1985, COM-33, no. 10, pp. 1132-1133.

2 Huang, C. M., Bi, Q., Stiles, G. S., and Harris, R. W.: 'Fast full search equivalent encoding algorithms for image compression using vector quantization', IEEE Trans. Image Process., July 1992, vol. 1, no. 3, pp. 413-416.

3 LI, W. and SALARI, E.: 'A fast vector quantization encoding method for image compression', IEEE Trans. Cir. Sys. Video Tech., 1995, vol. 5, pp. 119-123.

4 COSMAN, P. C., GRAY, R. M. and ETTERLI, M.: 'Vector quantization of image subbands : A survey', IEEE Trans. Image Process., 1996, vol. 5, no. 2, pp. 202-225.

5 GERSHO, A. and GRAY, R. M.: 'Vector quantization and signal compression', Kluwer Academic Publishers, Boston, 1992, pp. 196-197.

6 LINDE, Y., BUZO, A., and GRAY, R. M.: 'An algorithm for vector quantizer design', IEEE, Trans. Commun., Jan. 1980, vol. COM-28, pp. 84-95.

7 ANTONINI, M., BARLAUD, M., MATHIEU, P. and DAUBECHIES, I.: 'Image coding using wavelet transform', IEEE Trans. Image Processing, Apr. 1992, vol. 1, no. 2, pp. 205-220.