

Fast Encoding Algorithm for Vector Quantization

K.SOMASUNDARAM

Department of Computer Science and Applications
Gandhigram Rural University
Gandhigram – 624 302
Tamilnadu, India

S.VIMALA

Department of Computer Science
Mother Teresa Women's University
Kodaikanal – 624 102
Tamilnadu, India

Abstract

In this paper, we present a new and fast encoding algorithm (FEA) for vector quantization. The magnitude (sum of the components of a vector) feature of the vectors is used in this algorithm to improve the efficiency of searching. Sorting of the magnitude values enhances the searching. As the values are sorted, the searching can be terminated in advance to reduce the time needed to locate the representative code vector. For a codebook of size M (M generally being 128/256/512/1024), M distortion calculations are performed. But in the proposed method, only 11 distortion computations are done irrespective of the size of the codebook. The time taken to locate the representative codevector is significantly reduced from 0.77 seconds to 0.07 seconds on an average. The experiments were carried over with codebooks of sizes 128, 256, 512 and 1024 with the standard images Lena, Boats, Cameraman and Bridge.

Keywords: training vector; code vector; codebook; distortion; computational load.

I. Introduction

Vector Quantization is an efficient technique for data compression and has been successfully used for image compression because of its excellent rate-distortion performance and relatively simple structure. VQ finds its extensive use in various applications involving VQ-based encoding and VQ-based recognition [1], [2]. VQ has applications in different areas: protein classification, secondary structure computation [3], speech recognition, face detection, pattern recognition, real-time video based event detection and anomaly intrusion detection system, etc. [4]. With its relatively simple structure and computational complexity, VQ has received great attention in the last decade. The response time of encoding and recognition is a very important factor to be considered for real-time applications.

VQ Comprises of three stages: Codebook generation, Image encoding and Image Decoding. In VQ, for image compression, the input image is divided into several rectangular blocks which form vectors of fixed dimension. These blocks are called training vectors and the collection of training vectors form the training set of size N . N is computed using the equation (1).

$$N = m \times m / k \quad (1)$$

where k is the dimension of the vector and $m \times m$ is the number of pixels in the input image.

Generation of a codebook of size M ($M < N$) is of key importance in VQ. The quality of the reconstructed images mainly depends on the initial codebook generated.

Image encoding problem involves codeword searching to find the representative codevector for each training vector. Given one codeword $C_i = (c_{i1}, c_{i2}, c_{i3}, \dots, c_{ik})$ and the test vector $X = (x_1, x_2, x_3, \dots, x_k)$, the squared Euclidean distortion measure can be computed using the equation (1) to find the closest codevector.

$$d(X, C_i) = \sum_{j=1}^k (X_j - C_{ij})^2 \quad (2)$$

In the above equation, each distortion calculation requires k multiplications, $2k-1$ additions. For an exhaustive full search, encoding each input vector requires M distortion computations and M comparisons. Hence kM multiplications, $(2k-1)M$ additions and M comparisons are performed to encode each input vector.

One of the major issues of VQ encoding is to reduce the computational complexity from searching the best matched codeword in order to shorten the search time. Many efficient algorithms have been developed to reduce the computation load. The partial distortion search (PDS) is a simple and efficient codeword searching algorithm [6] that allows early termination of the distortion calculation between a training vector and the codeword by introducing a premature exit condition in the searching process, Nearest neighbor search algorithm based on the orthonormal transform [7], Partial distortion elimination (PDE) [8], Triangular inequality elimination (TIE) criterion used in [9]-[11] to reject a large number of unlikely codewords. The Equal Nearest Neighbor Search (ENNS) algorithm [12] uses the mean value of an input vector to reject impossible codewords. The improvement of ENNS algorithm, equal-average equal-variance nearest neighbor search (EENNS) uses the variance as well as the mean of a training vector to reject certain codewords [13]. The improved EENNS (IEENNS) method develops a new inequality between the features mean and variance, and the distance [14]. The improved ENNS algorithm (IENNS) is based on the inequality derived from the IAEI criterion [15].

In this paper, Fast Encoding Algorithm (FEA) is proposed to eliminate unnecessary distance computations while searching for the best-matched codevector.

To perform the experiment, the codebook was generated using ordered codebook generation technique (OCG) [16]. In this technique, the training vectors are arranged in ascending order of the sum of the components of the training vectors X_i using the equation (3).

$$S(X_i) = \sum_{j=1}^k X_{ij} \quad (3)$$

where $1 \leq i \leq N$ and $1 \leq j \leq k$

From the sorted list, the training vectors at every n^{th} position are selected to form the codebook of desired size. The value of n is computed using the equation (5). The codevectors are also in sorted order based on the sum of the components of the codevectors. Sum of the elements of the codevectors Y_i is calculated using the equation (4).

$$S(Y_i) = \sum_{j=1}^k Y_{ij} \quad (4)$$

where $1 \leq i \leq M$ and $1 \leq j \leq k$

$$n = N/M \quad (5)$$

where N is the size of the training set and M is the size of the codebook.

The remaining part of the paper is organized as follows: In section2, the proposed method (FEA) is explained. In section3, the results and discussion are given, and the conclusion is given in section4.

2. Proposed method (FEA)

As the codebook is generated using the ordered training set, the codevectors in the codebook are also in the sorted order. During each iteration, the sum of the components of the training vector X is calculated and is compared against the codevectors one by one. The comparison is done till the magnitude of the training vector is less than the magnitude of the codevector. Once the magnitude of the training vector is greater than that of the codevector, the comparison is terminated. This codeword is marked as the current point j . Now the codewords at the range of

positions from $j-5$ to $j+5$ alone are taken for computing the Euclidean distance. The other codevectors are eliminated. This reduces the time significantly. Now, the minimum distance between these 11 codewords and the training vector are computed to identify the codeword that is closer to the training vector using the equation (2). The index of the closest vector is then stored for the corresponding training vector.

For a codebook of size M (M generally being 128/256/512/1024), M distortion calculations are performed in exhaustive search method. But in the proposed method, irrespective of the size of the codebook, only 11 distortion computations are done. But there will be slight reduction in the PSNR value.

2.1. Algorithm for Initial Codebook Generation

- Step1:** Input the image of size $m \times m$ pixels.
- Step2:** Generate N training vectors using equation (1). Set the size of codebook as M .
- Step3:** Calculate the sum of the elements of the training vectors using the equation (3).
- Step4:** Rearrange the training vectors in ascending order based on the sum values using the equation.
- Step5:** Compute n using the equation (5)
- Step6:** Select the training vectors at every n^{th} position to form the codebook of desired size.

2.2. Algorithm for FEA method

- Step1:** Input the training set of size N along with the ordered codebook.
- Step2:** Compute the sum values S_x and S_{Y_j} using the equations (3) and (4)
- Step3:** Compare the value of S_x with the S_{Y_j} of all the codewords one by one
- Step4:** Terminate the comparison when the $S_x > S_{Y_j}$
- Step5:** Set the range of codevectors to be Y_{j-5} to Y_{j+5}
- Step6:** Compute the Euclidean distances between the training vector and the range of codevectors from position $j-5$ to $j+5$.
- Step7:** Identify the codevector whose Euclidean distance from the training vector is the minimum.
- Step8:** Encode the training vector with the index of the codevector identified in Step7.
- Step9:** The steps 2 thru. 8 are repeated for all the training vectors.

3. Results and discussion

Experiments were conducted on standard images Lena, Cameraman, Boats and Bridge of size 256×256 pixels. The algorithms are implemented using Matlab 7.0 on Windows Operating System. The hardware used is the Intel Core 2 E7400@ Duo 2.8. G.Htz. processor with 2 GB RAM.

The Euclidean distance computation involves kM multiplications, $(2k-1)M$ additions and M comparisons for a codebook of size. Hence for the size of 128, the ES method takes 16×128 multiplications, 31×128 additions and 128 comparisons = 6144 operations. For the same codebook size, the FEA method performs only 16×11 multiplications, 31×11 additions and 11 comparisons = 528 operations. Similar calculations are done for codebooks of sizes 256, 512 and 1024 and are given in Table-1. Irrespective of the codebook size, the FEA method performs only 11 computations against the codevectors at positions from $j-5$ to $j+5$, where j is the position at which the sum of the components of a training vector is less than that of the codevector.

Table-1: No. of computations required to encode the image with respect to exhaustive search (ES) algorithm and the proposed method (FEA).

Codebook Size	ES	FEA (Proposed)	Reduction %
128	6144	528	91.41%
256	12288	528	95.70%
512	24576	528	97.85%
1024	49152	528	98.93%

The reduction percentage is calculated using the equation (6).

$$\text{Reduction\%} = 100 - (\text{No. of operations by FEA} / \text{No. of operations by ES}) * 100 \quad (6)$$

Table-2: Time Taken To Encode The Image With Respect To Exhaustive Search (ES) Method And The Proposed Method (FEA).

Codebook Size	Lena		Camera		Boats		Bridge	
	ES	FEA	ES	FEA	ES	FEA	ES	FEA
128	0.2	0.06	0.2	0.06	0.2	0.06	0.2	0.06
256	0.4	0.08	0.41	0.06	0.39	0.06	0.41	0.08
512	0.83	0.08	0.83	0.08	0.84	0.08	0.83	0.06
1024	1.66	0.08	1.66	0.09	1.66	0.09	1.64	0.09

The time taken by the exhaustive search method and the proposed method to encode the training sets taken for the study is given in Table-2. The time varies depending on the size of the codebook in ES method. But the time remains more or less the same (0.06 to 0.09) for all the sizes of codebooks in FEA method. This is because, the number of operations remain the same for the proposed method for all codebook sizes. It is observed from Table-1 and Table-2, the number of operations and the time taken are doubled in ES method as the codebook size is doubled.

4. Conclusion

We have proposed a method FEA to minimize the time taken to encode the image by eliminating the unwanted distance computations. The number of operations (additions, multiplications and comparisons) required for encoding the image is reduced by 95.97% on an average for the codebooks of sizes 128, 256, 512 and 1024. The number of operations is doubled as the codebook size is doubled in ES, but remains the same for all the codebook sizes in FEA. The time taken by FEA is reduced by an average of 90.29% when compared to that of ES method with a slight degradation in PSNR value. Hence the FEA method may find place in applications, where time is a prime factor.

References

- [1] R.M.Gray, "Vector Quantization", IEEE ASSP Mag., pp. 4-29, Apr. 1984.
- [2] 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, 1980.
- [3] Juan J. Merelo Guervos, and et, al., "Protein Classification and Secondary Structure Computation" in Proceedings of the International Workshop on Artificial Neural Networks, Pages 415-421, 1991, Springer – Verlag.
- [4] C.Garcia and G.Tziritia, "Face Detection using Quantized Skin Color Regions Merging and Wavelet Packet Analysis", IEEE Trans. Multimedia, vol. 1, No. 3, pp. 264-277, September 1999.
- [5] Jeng-Shyang Pan, Zhe-Ming Lu, and Sheng-He Sun, "An Efficient Encoding Algorithm for Vector Quantization based on Subvector Technique", IEEE Transactions on Image Processing, Vol. 12, No. 3, pp. 265-270, March 2003.
- [6] C.D.Bei and R.M.Gray, "An Improvement of the Minimum Distortion Encoding Algorithm for Vector Quantization", IEEE Trans. Commun., Vol. COM-33, No. 10, pp. 1132-1133, 1985.
- [7] Z. Li, and Z.- M. Lu. : 'Fast codevector search scheme for 3D mesh model vector quantization', Electron. Lett., vol. 44, No. 2, pp. 104-105, Jan 2008.
- [8] C. Bei, R. M. Gray, "An improvement of the minimum distortion encoding algorithm for vector quantization", IEEE Trans. Commun., Vol. 33, no. 10, pp. 1132–1133, Oct 1985.
- [9] C.Garcia and G.Tziritia, "Face Detection using Quantized Skin Color Regions Merging and Wavelet Packet Analysis", IEEE Trans. Multimedia, vol. 1, No. 3, pp. 264-277, September 1999.
- [10] C.D.Bei and R.M.Gray, "An Improvement of the Minimum Distortion Encoding Algorithm for Vector Quantization", IEEE Trans. Commun., Vol. COM-33, No. 10, pp. 1132-1133, 1985.
- [11] Z. Li, and Z.- M. Lu. : 'Fast codevector search scheme for 3D mesh model vector quantization', Electron. Lett., vol. 44, No. 2, pp. 104-105, Jan 2008.
- [12] C. Bei, R. M. Gray, "An improvement of the minimum distortion encoding algorithm for vector quantization", IEEE Trans. Commun., Vol. 33, no. 10, pp. 1132–1133, Oct 1985.
- [13] C.H.Lee and L.H.Chen, "Fast closest codeword search algorithm for vector quantization", Prc. Inst. Elect. Eng., Vol. 141, No. 3, pp. 143-148, 1994.
- [14] S.J.Baek, B.K.Jeon, and K.M.Sung, "A fast encoding algorithm for vector quantization ", IEEE Signal Processing Lett., Vol. 4, pp. 325-327, Feb. 1997.
- [15] J.S.Pan, F.R.McInnes, and M.A.Jack, "Fast clustering algorithm for vector quantization", Pattern Recognition, Vol. 29, No. 3, pp. 511-518, 1996.
- [16] K.Somasundaram and S.Vimala, "Simple and Fast Ordered Codebook Generation For Vector Quantization," Proceedings of the National Conference, Gandhigram Rural Institute, ISBN 978-81-8424-574-5, Mar 2010.