

Short Paper

A Training Approach for Efficient VQ Codeword Search*

J. S. PAN

*Department of Electronic Engineering
National Kaohsiung Institute of Technology
Kaohsiung, Taiwan 807, R.O.C.
E-mail: jspan@cc.nkit.edu.tw*

In this paper, an efficient approximate VQ codeword search algorithm is proposed. This algorithm is based on a modification of the Chebyshev metric (or Manhattan metric). Applying this new algorithm to VQ codeword search and comparing it with the minimax method, it is found that more than 36% and 5% multiplications can be saved for 8 and 1024 codewords, respectively. In terms of the total number of mathematical operations, a few mathematical operations can be saved without inducing any extra distortion. Experimental results confirm this new algorithm.

Keywords: VQ, Euclidean metric, Manhattan metric, Chebyshev metric, codeword search

1. INTRODUCTION

Vector Quantization (VQ) [1,2] has been widely used in various applications involving VQ-based encoding and VQ-based recognition. The response time of encoding and recognition is a very important factor to be considered in real-time applications. Unfortunately, a full search algorithm is applied in VQ encoding and recognition, and this is a time consuming process when the codebook size is large. A vector quantizer of rate γ bits/sample and dimension k is a mapping from a k -dimensional vector space into some finite subset $C = \{C_j; j = 1, \dots, N\}$, where $N = 2^{k\gamma}$. The subset C is called a codebook, and its elements C_j are called codewords, codevectors, reproducing vectors, prototypes, or design samples. A distortion measure $D(X, C_j)$ is a non-negative dissimilarity measure between vector X and codewords C_j . This distortion is used to measure how close the input vector X is to these codewords C_j . The nearest codeword is selected in order to encode the input vector X . Therefore, encoding each input vector requires N distortion computations and $N-1$ comparisons.

The codeword search problem in vector quantization is to assign one codeword to the test vector in which the distortion between this codeword and the test vector is the smallest among all the codewords. Given one codeword C_i and the test vector X in the k -dimensional space, the distortion of the squared Euclidean metric can be expressed as follows:

Received December 19, 1997; revised May 6, 1998; accepted June 11, 1998.

Communicated by Soo-Chang Pei.

* This work was supported by the National Science Council, Taiwan, R.O.C. under Grant No. NCS-87-2213-E-151-007.

$$D(X, C_i) = \sum_{i=1}^k (x^i - c_i^i)^2,$$

where $C_i = \{c_i^1, c_i^2, \dots, c_i^k\}$

and $X = \{x^1, x^2, \dots, x^k\}$.

Each distortion calculation requires k multiplications and $2k-1$ additions. Therefore, it is necessary to perform $k2^{kr}$ multiplications, $(2k-1)2^{kr}$ additions, and $2^{kr} - 1$ comparisons to encode each input vector. The computation complexity depends on the codebook size and dimensions. A large codebook size and higher dimension are needed to obtain high performance in VQ encoding and recognition systems, thus resulting in increased computation load during codeword search.

2. BRIEF HISTORY OF CODEWORD SEARCH

Since codeword search is a serious problem in real time application of vector quantization, many algorithms have been proposed to improve the speed of codeword search [3-13]. The partial distortion search (PDS) algorithm [3] is a simple and efficient codeword search algorithm which allows early termination of the distortion calculation between a training vector and a codeword by introducing a premature exit condition in the search process. Given the current minimum distortion:

$$D(X, C_i) = D_{\min}, \quad (1)$$

$$\text{if } \sum_{i=1}^s (x^i - c_j^i)^2 \geq D_{\min}, \quad (2)$$

$$\text{then } D(X, C_j) \geq D(X, C_i), \quad (3)$$

where $s \leq k$.

The efficiency of PDS derives from elimination of an unfinished distortion computation if its partial accumulated distortion is larger than the current minimum distortion. This will reduce computation to $(k-s)$ multiplications and $2(k-s)$ additions at the expense of s comparisons.

The hypercube approach is a well known premature method [4] which is efficient if the difference for any coefficient is generally larger than the difference of the other coefficients, such as the first coefficient of cepstrum coefficients. Assume Eq. 1 already exists:

$$\text{if } |x^j - c_j^i| \geq \sqrt{D_{\min}}, \quad (4)$$

then C_j will not be the nearest neighbour to X . No multiplication operation is required to test the hypercube approach.

The minimax method [5] is to take the codeword with the minimum value of maximum dimension-distortion as the tentative match and then use the hypercube approach and partial distortion search (PDS). The absolute error inequality criterion [6], the mathematical relationship between the city block metric (or L_1) and the Euclidean metric (or L_2), is used to reduce the computational complexity of codeword search. The bound for the Minkowski metric [7], which is also derived to speed up codeword search, has been proved to be a generalized form of PDS, the hypercube approach, absolute error inequality (AEI) criterion, and improved absolute error inequality (IAEI) criterion [8].

The computation time for the approximating and eliminating search algorithm (AESA) [9] is approximately constant for codeword search under a large codebook size. The triangular inequality elimination (TIE) and high correlation characteristics between adjacent speech frames are combined to get an efficient codeword search approach [10]. Taking advantage of the fact that the nearest codeword is usually in the neighborhood of the minimum squared mean distance, the mean-distance-ordered partial codebook search (MPS) algorithm [11] was proposed to speed up codeword search. The generalized form of the MPS algorithm was derived and a new approach was presented to further improve search efficiency [12]. The principal component transform and the geometrical relations between the input data vector and codewords are utilized to eliminate impossible codeword match [13]. In this paper, a totally different approach to efficient codeword search is proposed. The idea behind this approach is to modify the Manhattan metric (or Chebyshev metric) [14] so as to match the Euclidean distortion measure using a suitable training procedure so that the number of multiplication operations can be drastically reduced.

3. CODEWORD ELIMINATION METHOD

Multiplication operations are far more expensive compared with comparison and addition operations for general processors [15]. In this paper, an efficient approximate search algorithm which can drastically reduce the number of multiplication operations is presented. This algorithm is based on a modification of the Chebyshev metric (or Manhattan metric). Assume that the training data and codewords are $X_p = \{x_p^1, x_p^2, \dots, x_p^k\}$ and $C_i = \{C_i^1, C_i^2, \dots, C_i^k\}$, respectively, where $p = 1, 2, \dots, T$, $i = 1, 2, \dots, N$. T , k and N are the total number of training data vectors, the number of dimensions and the number of codewords, respectively. The squared Euclidean distortion between data vector X_p and codeword C_i can also be expressed as follows:

$$d(l, p) = \sum_{j=1}^k (x_p^j - c_i^j)^2. \tag{5}$$

The distortion of the Manhattan metric between data vector X_p and codeword C_i is

$$d_m(l, p) = \max_j |x_p^j - c_i^j|. \tag{6}$$

Obviously, it is multiplication free if the Manhattan metric is used to perform codeword search. Unfortunately, the squared Euclidean distortion measure is used in most codeword search applications. Intuitively, we expect to multiply a real coefficient by the Manhattan

metric, called the modified Manhattan metric, such that if the distortion for the data vector X_p and one codeword C_l using the Manhattan metric is larger than the distortion of the other codeword using the modified Manhattan metric, then C_l can be rejected as the codeword nearest to data vector X_p . The distortion between data vector X_p and codeword C_l using the modified Manhattan metric can be expressed as follows:

$$md_m(l, p) = rate \cdot d_m(l, p), \quad (7)$$

where $rate$ is a real coefficient.

Here, we need to find the parameter $rate$ by developing the training approach such that

$$\text{if } d_m(l, p) \geq md_m(j, p), \quad (8)$$

$$\text{then } d(l, p) \geq d(j, p). \quad (9)$$

The codeword with the minimum value of the maximum dimension-distortion is

$$n_p = \arg \min_i \max_j |x_p^j - c_i^j|, \quad (10)$$

and

$$d(n_p, p) = \sum_{j=1}^k (x_p^j - c_{n_p}^j)^2, \quad (11)$$

where n_p is the codeword nearest to the data vector X_p using the Manhattan metric. All the codewords C_i can be separated into two sets for every training data vector X_p :

$$\text{First set : } A_p = \{i \mid d(i, p) \geq d(n_p, p)\}. \quad (12)$$

$$\text{Second set : } B_p = \{i \mid d(i, p) < d(n_p, p)\}. \quad (13)$$

Next, we calculate the parameter $rate$ using Eqs. 14 and 15:

$$rate_p = \frac{\max_{l \in B_p} \max_j |c_l^j - x_p^j|}{\max_j |c_{n_p}^j - x_p^j|} \quad (14)$$

for each training data vector X_p .

Using the Manhattan metric instead of the squared Euclidean distortion measure, the distortion will increase if any codeword is located in the second set. Since the worst case of the multiplication coefficient is calculated using Eq.14, this guarantees that Eqs. 8 and 9 can be satisfied for data vector X_p . If all the training data vectors are considered, then

$$rate = \max_p rate_p + \delta, \quad (15)$$

where δ is a small value. If the training data is large enough, δ can be set to 0; otherwise, a small value should be assigned to δ so that no extra distortion can be induced for the test data. The parameter $rate$ can be computed from the training data. After the parameter $rate$ is obtained, a new codeword elimination criterion is developed as follows:

$$\text{if } \max_j |x_m^j - c_l^j| \geq \text{rate} \cdot \max_j |x_m^j - c_{n_p}^j|, \quad (16)$$

$$\text{then } \sum_{j=1}^k (x_m^j - c_l^j)^2 \geq \sum_{j=1}^k (x_m^j - c_{n_p}^j)^2, \quad (17)$$

$$\text{where } n_p = \arg \min_i \max_j |x_m^j - c_i^j|. \quad (18)$$

The efficiency of codeword search depends on the value of the parameter *rate*. The smaller the *rate* value is, the more efficient the algorithm is. In the extreme case, *rate* = 1, it is the Chebyshev metric or Manhattan metric. For this metric, the number of multiplications, comparisons and additions is 0, N(k-1)+(N-1) and Nk, respectively. The parameter *rate* can be reduced to a smaller value if the increase in distortion is small. If the training data set is large enough, this codeword elimination criterion is a powerful search approach, yet it does not induce any extra distortion. After obtaining the parameter *rate* from the training procedure, the efficient codeword search algorithm is as follows:

- Step 1:** Calculate the squared Euclidean distortion for data vector X_m and one codeword C_l .
- Step 2:** Use the hypercube approach to eliminate the codeword C_l .
- Step 3:** If the hypercube approach can not eliminate the codeword C_l , then use Eqs. 16 and 17 as the codeword elimination criterion.
- Step 4:** If the codeword C_l cannot be eliminated in step 2 and step 3, then use partial distortion search (PDS) to reduce the number of mathematical operations and update both the current minimum distortion and the maximum dimension distortion of the current nearest codeword (i.e., the distortion of the Manhattan metric). If all the codewords are tested, then terminate the search procedure; otherwise, go to step 2 to test the next codeword.

4. EXPERIMENTS AND CONCLUSIONS

The test materials for these experiments consisted of two hundred codewords recorded from two male speakers. The speech was sampled at a rate of 16 kHz, and 13-dimensional cepstrum coefficients were computed over 20 ms-wide frames with a 5 ms frame shift. A total of 20,030 analyzed frames used as the training data were recorded from one male speaker. The test data included 30,096 analyzed frames recorded from the other speaker. Codebooks of size 8, 256 and 1,024 codewords with a squared Euclidean distortion measure were used in these experiments.

Tables 1, 2 and 3 illustrate the performance of the conventional (or exhaustive full search) method, minimax method and this new efficient approximate search algorithm with different *rates*. The conventional method is denoted as cvt. The training rates were 1.874833, 2.028345 and 2.231797 for 8, 256 and 1024 codewords, respectively. Here δ was set to 0. The average distortion was the same as that of the conventional method if the training rates were used. If the training rates were used, more than 36% and 5% multiplication operations can be saved for 8 and 1024 codewords, respectively, by applying this new approach instead of the minimax method. In terms of the total number of mathematical operations, some operations can also be saved without inducing any extra distortion. Experimental results confirm the usefulness of this new approach.

Table 1. Performance comparison of the conventional method, minimax method and fast approximate algorithm for 8 codewords.

method	mul.	cmp.	add.	sum	average distortion
cvt	3,129,984	210,672	6,019,200	9,359,856	0.940086
minimax	502,231	3,376,691	3,587,363	7,466,285	0.940086
rate_1.0	0	3,099,888	3,129,984	6,229,872	0.959554
rate_1.1	97,216	3,332,328	3,189,985	6,619,529	0.945844
rate_1.2	142,396	3,345,996	3,230,290	6,718,682	0.941630
rate_1.3	178,404	3,355,566	3,262,337	6,796,307	0.940600
rate_1.4	209,539	3,367,516	3,289,937	6,861,985	0.940210
rate_1.5	237,772	3,367,516	3,314,977	6,920,265	0.940139
rate_1.6	262,278	3,371,061	3,336,813	6,970,152	0.940096
rate_1.7	282,651	3,373,216	3,355,062	7,010,929	0.940093
rate_1.8	301,284	3,374,809	3,371,852	7,047,945	0.940088
rate_1.9	316,990	3,375,730	3,386,098	7,078,818	0.940086
rate_2.0	330,482	3,376,189	3,398,439	7,105,110	0.940086
rate_2.1	341,306	3,376,384	3,408,397	7,126,087	0.940086
rate_2.2	352,263	3,376,497	3,418,520	7,147,280	0.940086
rate_2.3	361,709	3,376,520	3,427,279	7,165,508	0.940086
rate_2.4	370,395	3,376,539	3,435,332	7,182,266	0.940086
rate_2.5	378,380	3,376,542	3,442,745	7,197,667	0.940086

Table 2. Performance comparison of the conventional method, minimax method and fast approximate algorithm for 256 codewords.

method	mul.	cmp.	add.	sum	average distortion
cvt	100,159,488	7,674,480	192,614,400	300,448,368	0.346460
minimax	2,710,965	109,599,202	102,346,066	214,656,233	0.346460
rate_1.0	0	100,129,392	100,159,488	200,288,880	0.388426
rate_1.1	428,683	107,992,358	100,517,026	208,938,067	0.364018
rate_1.2	770,084	108,202,247	100,818,487	209,790,818	0.353663
rate_1.3	1,087,047	108,424,879	101,091,793	210,603,719	0.349195
rate_1.4	1,388,239	108,647,096	101,344,105	211,379,440	0.347424
rate_1.5	1,675,786	108,862,034	101,576,736	212,114,556	0.346771
rate_1.6	1,932,988	109,051,837	101,777,160	212,761,985	0.346557
rate_1.7	2,157,441	109,213,439	101,945,392	213,316,272	0.346482
rate_1.8	2,337,348	109,338,814	102,075,174	213,751,336	0.346468
rate_1.9	2,471,616	109,429,695	102,168,902	214,070,213	0.346461
rate_2.0	2,563,560	109,490,166	102,231,336	214,285,062	0.346460
rate_2.1	2,619,365	109,525,849	102,268,176	214,413,390	0.346460
rate_2.2	2,650,193	109,544,662	102,288,120	214,482,975	0.346460
rate_2.3	2,665,490	109,553,760	102,297,907	214,517,157	0.346460
rate_2.4	2,672,197	109,557,447	102,302,218	214,531,862	0.346460
rate_2.5	2,674,831	109,558,755	102,303,955	214,537,541	0.346460

Table 3. Performance comparison of the conventional method, minimax method and fast approximate algorithm for 1024 codewords.

method	mul.	cmp.	add.	sum	average distortion
cvt	400,637,952	30,788,208	770,457,600	1,201,883,760	0.271360
minimax	6,457,314	436,016,497	405,649,633	848,123,444	0.217360
rate_1.0	0	400,607,856	400,637,952	801,245,808	0.310621
rate_1.1	577,887	431,680,772	401,129,451	833,388,110	0.288017
rate_1.2	1,110,517	432,055,772	401,599,403	834,765,692	0.278229
rate_1.3	1,684,421	432,507,813	402,095,215	836,287,449	0.274065
rate_1.4	2,307,142	433,010,905	402,619,302	837,937,349	0.272397
rate_1.5	2,960,568	433,533,163	403,152,623	839,646,354	0.271760
rate_1.6	3,620,279	434,047,396	403,672,576	841,340,251	0.271497
rate_1.7	4,255,817	434,526,434	404,154,916	842,937,167	0.271393
rate_1.8	4,827,987	434,942,906	404,573,078	844,343,971	0.271366
rate_1.9	5,301,655	435,275,810	404,907,048	845,484,513	0.271360
rate_2.0	5,654,318	435,515,429	405,147,304	846,317,051	0.271360
rate_2.1	5,890,136	435,671,173	405,303,503	846,864,812	0.271360
rate_2.2	6,031,703	435,762,281	405,394,910	847,188,894	0.271360
rate_2.3	6,106,298	435,808,957	405,441,729	847,356,984	0.271360
rate_2.4	6,139,254	435,829,116	405,461,953	847,430,323	0.271360
rate_2.5	6,152,092	435,836,743	405,469,710	847,458,545	0.271360

REFERENCES

1. R. M. Gray, "Vector quantization," *IEEE Acoustics, Speech and Signal Processing Magazine*, Vol. 1, No. 2, 1984, pp. 4-29.
2. A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, 1992.
3. C. Bei and R. M. Gray, "An improvement of the minimum distortion encoding algorithm for vector quantization," *IEEE Transactions on Communications*, Vol. 33, No. 10, 1985, pp. 1132-1133.
4. K. T. Lo and W. K. Cham, "Subcodebook searching algorithm for efficient VQ encoding of images," *IEE Proceedings I*, Vol. 140, No. 5, 1993, pp. 327-330.
5. D. Y. Cheng, A. Gersho, B. Ramamurth and Y. Shoham "Fast search algorithms for vector quantization and pattern matching," *IEEE International Conference on Acoustics, Speech and Signal Processing*, 1984, pp. 9.II.1-9.II.4.
6. M. R. Soleymani and S. D. Morgera, "A high-speed algorithm for vector quantization," *IEEE International Conference on Acoustics, Speech and Signal Processing*, 1987, pp. 1946-1948.
7. J. S. Pan, F. R. McInnes and M. A. Jack, "Bound for Mindowski metric or quadratic metric applied to VQ codeword search," *IEE Proceedings-Vision, Image and Signal Processing*, Vol. 143, No. 1, 1996, pp. 67-71.
8. J. S. Pan, F. R. McInnes and M. A. Jack, "Fast clustering algorithms for vector quantization," *Pattern Recognition*, Vol. 29, No. 3, 1996, pp. 511-518.

9. E. Vidal, "An algorithm for finding nearest neighbours in (approximately) constant average time," *Pattern Recognition Letters*, Vol. 4, No. 3, 1986, pp. 145-157.
10. S. H. Chen and J. S. Pan, "Fast search algorithm for VQ-based recognition of isolated word," *IEE Proceedings I*, Vol. 136, No. 6, 1989, pp. 391-396.
11. S. -W Ra and J. K. Kim, "A fast mean-distance-ordered partial codebook search algorithm for image vector quantization," *IEEE Transaction on Circuits and Systems-II: Analog and Digital Signal Processing*, Vol. 40, No. 9, 1993, pp. 576-579.
12. J. S Pan and K. C. Huang, "A new vector quantization image coding algorithm based on the extension of the bound for Minkowski metric," *Pattern Recognition*, Vol. 31, No. 11, 1998, pp 1757-1760.
13. S. C. Tai, C. C. Lai and Y. C. Lin, "Two fast nearest neighbor searching algorithms for image vector quantization," *IEEE Transactions on Communications*, Vol. 44, No. 12, 1996, pp. 1623-1628.
14. J. R. Deller, Jr. J. G. Proakis and J. H. L. Hansen, *Discrete-time Processing of Speech Signals*, Macmillan Publishing Company, 1993
15. S. H. Leibson, "EDN-microprocessor directory," *Electrical Design News*, November 1993, pp. 148-148.

J. S. Pan (潘正祥) received the B.S. degree in Electronic Engineering from National Taiwan University of Science and Technology, Taiwan, in 1986, the M.S. degree in Communication Engineering from National Chiao Tung University, Taiwan, in 1988, and the Ph.D. degree in Electrical Engineering from the University of Edinburgh, U.K., in 1996. Currently, he is an associate professor in the Department of Electronic Engineering, National Kaohsiung Institute of Technology, Taiwan. His current research interests include pattern recognition, speech coding and image processing. Dr. Pan is a member of Pattern Recognition Society.