A Fast Codebook Design Algorithm for ECVQ Based on Angular Constraint and Hyperplane Decision Rule(Image)

PAPER

A Fast Codebook Design Algorithm for ECVQ Based on Angular Constraint and Hyperplane Decision Rule

Ahmed SWILEM†, *Nonmember*, Kousuke IMAMURA††, *and* Hideo HASHIMOTO††a), *Members*

SUMMARY In this paper, we propose two fast codebook generation algorithms for entropy-constrained vector quantization. The first algorithm uses the angular constraint to reduce the search area and to accelerate the search process in the codebook design. It employs the projection angles of the vectors to a reference line. The second algorithm has feature of using a suitable hyperplane to partition the codebook and image data. These algorithms allow significant acceleration in codebook design process. Experimental results are presented on image block data. These results show that our new algorithms perform better than the previously known methods.
key words: *fast search algorithm, entropy-constrained vector quantization, hyperplane decision rule*

1. Introduction

Vector quantization (VQ) [1] has played an important role in numerous data compression systems. It is defined as a mapping Q from a k-dimensional Euclidean space R^k to a finite set $Y = \{y_1, y_2, \ldots, y_N\}$ of vectors in R^k called the codebook. Each representative vector y_i in the codebook is called a codeword. A complete description of vector quantization process includes three phases: codebook design, encoding and decoding. The objective of codebook design is to construct a codebook Y from a set of training vectors using clustering algorithms like the generalized Lloyd algorithm (GLA) [2]. This codebook is used in both the encoder and the decoder. The encoding phase is equivalent to find the vector $Q(x) = y_i \in Y$ minimizing the distortion $d(x, y_i)$ defined as the Euclidean distance between the vector x and y_i. The decoding phase is simply a table look-up procedure that uses the received index i to deduce the reproduction codeword y_i, and then uses y_i to represent the input vector x.

Entropy-constrained vector quantization (ECVQ), which incorporates an entropy constraint within the design procedure, is one of the fundamental extensions of the basic VQ concept so as to produce quantizers optimized for subsequent entropy coding. The pioneering work in this area was performed by Berger [3], Farvardin and Modestino [4], and Chou et at. [5]. Incorporation of an entropy constraint has been considered in a variety of quantization schemes, notably works of Pearlman and colleagues [6], [7] and others [8], [9].

ECVQ employs a modified cost measure using both the effective distortion of the signal and the expected length of the transmitted code. This length is not always equal to $\log_2 N$, where N is the codebook size, but dependent on the expected probability of the codeword. The codeword length $R(y_i)$ of the codeword y_i is usually taken as equal to the bound given by the entropy model, i.e. $R(y_i) = -\log_2 P(y_i)$, where the probability $P(y_i)$ is approximated empirically using the training set. We define the cost function for encoding the vector x by the codeword y_i as the Lagrangian function,

$$J(x, y_i) = d(x, y_i) + \lambda R(y_i), \tag{1}$$

where $d(x, y_i)$ is the Euclidean distance, and λ is a constant called the Lagrange multiplier allowing to control the rate-distortion ratio. Using of this cost measure implies that codewords introducing higher degradation may be chosen because of their short descriptions.

In many VQ applications, the computational cost of finding the nearest neighbor codeword in the codebook design and encoding imposes practical limits on the codebook size N. When N becomes larger, the computational complexity problem occurs for full codebook search. This has motivated the development of many fast nearest neighbor search algorithms. Algorithms to reduce search complexity ([10]–[19]) concentrate on narrowing the area of the candidate codewords for which distortion must be calculated. These techniques have their roots in pattern recognition techniques for the finding nearest neighbor [20], [21].

An algorithm for fast nearest neighbor search presented by Orchard [10] precomputes and stores the distance between each pair of codewords. Given an input vector x, the current best codeword y_i, and a candidate codeword y_j, if $d(x, y_j) \le d(x, y_i)$, then $d(y_j, y_i) \le 2d(x, y_i)$. Graphically, this constrains the search area within a sphere centered on the current best codeword, with a radius of twice the smallest distortion calculated so far. Huang et at. [11] introduced an additional constraint on codewords by sorting their distances from the origin. The distance between the current best codeword and the input vector constrains the search to codewords within an area about the origin, which is represented by an annulus in two dimensions. In three dimensions, this is the area between two concentric spheres. This constraint is known as the annular constraint. In the same paper, Huang et al. proposed a combination of the spherical and annular constraints with an efficient search method. Lee and Chen [12] introduced a projection method, which uses

Manuscript received February 24, 2003.
Manuscript revised October 17, 2003.
Final manuscript received December 2, 2003.
†The author is with the Graduate School of Natural Science & Technology, Kanazawa University, Kanazawa-shi, 920-8667 Japan.
††The authors are with the Faculty of Engineering, Kanazawa University, Kanazawa-shi, 920-8667 Japan.
a) E-mail: hasimoto@is.t.kanazawa-u.ac.jp

the mean and the variance of the vector as two constraints to reject the codewords. We described a lossy design method in [18], which employs a hyperplane decision rule to separate the search areas in Lee and Chen method and is considered as the extension of it. Johnson et at. [14] generalized the techniques in [10] and [11] to apply them to class of vector quantizers using Lagrangian distortion measure, in which a sum of the Euclidean distance and some constant assigned to each codeword is incorporated. Another technique added an additional endpoint for the annular constraint was proposed in the same paper. This constraint is known as the double annulus constraint. Also, Cardinal [15] presented an extension of the method in [12] to ECVQ. This method is considered as the better known acceleration method for nearest neighbor search for ECVQ.

In this paper, two new fast ECVQ design algorithms are proposed. The first one achieves equivalent performance to the full search ECVQ. It uses the annular constraint and another constraint called the angular constraint. This method uses the projection angles on a reference line in the space of input vectors and codewords. It searches a smaller number of codewords than the previous methods. The second method uses a hyperplane partitioning rule, which separates the codebook and the training vectors into two parts, and searches in only one part according to the vector feature. The searching in this method speeds up the codebook design process, but signal quality is sacrificed a little. The efficiency of the proposed new methods is compared with the previous methods.

The paper is organized as follows. Section 2 reviews the double annulus method and Cardinal method. Section 3 presents the angular constraint method. Section 4 describes the angular constraint method with the hyperplane decision rule. Experimental results are shown in Sect. 5, and concluding remarks are given in Sect. 6.

2. Fast Algorithms for ECVQ

2.1 Double Annulus Method

Johnson et at. [14] introduced an excellent method called the double annulus method for ECVQ using two annular constraints, and tried to search only those codewords lying in their overlapped area. The first annulus is centered at the origin that is the first reference point. For a given input vector x of distance $\|x\|$ from the origin and a current best codeword y_i with Lagrangian distortion $J(x, y_i)$, any closer codeword y_j to x than y_i in the sense of the Lagrangian cost measure will satisfy the following relationships:

$$\|y_j\| + \lambda R(y_j) < \|x\| + J(x, y_i), \tag{2}$$

and

$$\|y_j\| - \lambda R(y_j) > \|x\| - J(x, y_i), \tag{3}$$

where $\|y_j\|$ is the Euclidean distance of y_j from the origin, and $R(y_j)$ is the length of the codeword y_j. Thus, for any codeword y_j satisfying (2) and (3), the hypersphere centered at y_j with radius $\lambda R(y_j)$ must be fully contained in the annulus defined by $\|x\| + J(x, y_i)$ and $\|x\| - J(x, y_i)$.

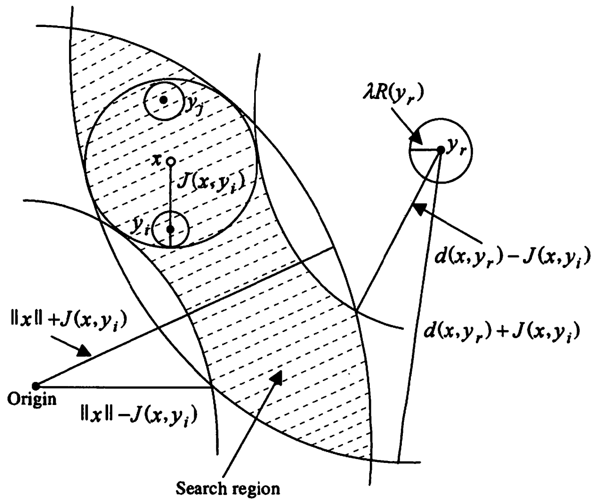


Fig. 1 Geometrical interpretation of double annulus method in 2-dimensional case.

The second annulus is centered at the farthest codeword from the origin, which is the second reference point, y_r. By using the distance to this codeword, the following inequalities can be defined:

$$d(y_j, y_r) + \lambda R(y_j) < d(x, y_r) + J(x, y_i), \tag{4}$$

and

$$d(y_j, y_r) - \lambda R(y_j) > d(x, y_r) - J(x, y_i). \tag{5}$$

The inequalities (2), (3), (4) and (5) constrain the distortion calculation to the codeword whose hypersphere is completely contained in the search region shown in Fig. 1.

2.2 Cardinal Method

Cardinal [15] introduced the most acceleration method for GLA on ECVQ using two elimination rules. In the first elimination rule, a unit vector $u = (1, 1, \ldots, 1)/\sqrt{k}$ on the central line is used as a reference line as shown in Fig. 2, where k is the vector dimension. For a given input vector x and a current best codeword y_i with Lagrangian distortion $J(x, y_i)$, any closer codeword y_j to x than y_i with length $R(y_j)$ will satisfy the following inequalities:

$$uy_j^T + \lambda R(y_j) < ux^T + J(x, y_i), \tag{6}$$

and

$$uy_j^T - \lambda R(y_j) > ux^T - J(x, y_i). \tag{7}$$

The rule in (6) and (7) is very similar to the rule in (2) and (3). While the rule in (2) and (3) uses the length of the vectors, the rule in (6) and (7) uses the projection of the vectors on u. The length of the vector is actually its distance to the origin o, but its projection on u may be seen as its parallel component to u. From the geometrical interpretation of

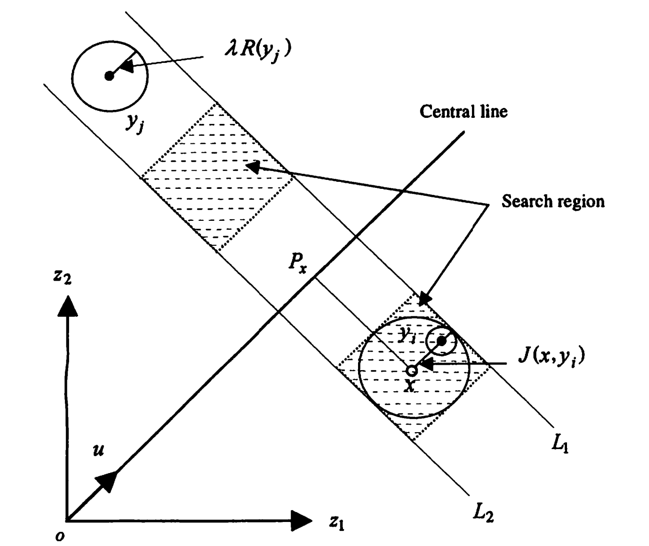


Fig. 2 Geometrical interpretation of Cardinal method in 2-dimensional case.

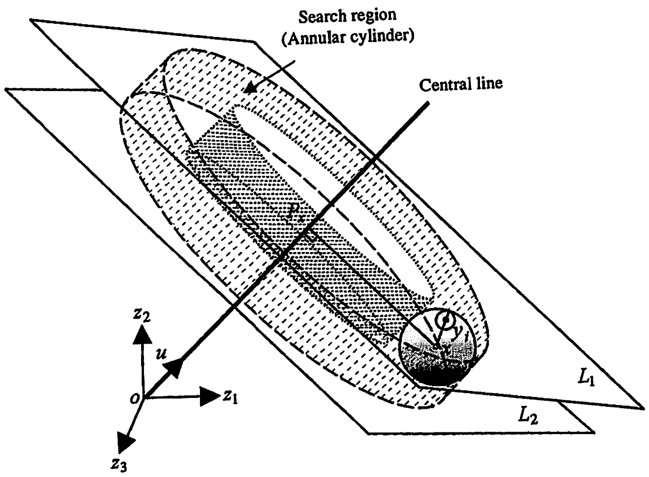


Fig. 3 Geometrical interpretation of Cardinal method in 3-dimensional case.

this method in 2-dimensional case in Fig. 2, for any codeword y_j satisfying (6) and (7), the hypersphere centered at y_j with radius $\lambda R(y_j)$ must be fully contained in the region between the two hyperplanes $L_1 : uz^T = ux^T + J(x, y_i)$ and $L_2 : uz^T = ux^T - J(x, y_i)$.

In the second elimination rule, the distance between the codeword and its projection point on the central line is used as follows: for a given input vector x with its projection point P_x on the central line, the closest codeword y_j with its projection point P_{y_j} will satisfy the following inequalities:

$$d(y_j, P_{y_j}) + \lambda R(y_j) < d(x, P_x) + J(x, y_i), \tag{8}$$

and

$$d(y_j, P_{y_j}) - \lambda R(y_j) > d(x, P_x) - J(x, y_i). \tag{9}$$

By using the constraints of the rule in (6) and (7) and the rule in (8) and (9), the search region will be reduced to the two dotted squares in Fig. 2. The search region in 3-dimensional case is expanded to an annular cylinder shown in Fig. 3. Every codeword whose sphere is not contained in this region is eliminated. Cardinal method is considered as the generalization of Lee and Chen method [12] to ECVQ.

3. Angular Constraint Method (Lossless Design)

The angular constraint was proposed in [22] for a nearest neighbor search technique of mean-shape-gain VQ, in which the best shape vector is searched on a hypersphere of radius 1 using the angle between the input shape vector and a reference direction. Our proposing angular constraint method is its extension, and a newly developed one applicable to ECVQ as well as standard VQ according to our limited knowledge.

In this section, a new method for codebook generation is proposed by using the angular constraint. As we saw in the last section, the annulus method constrains the search region by the two inequalities (2) and (3). For any codeword y_j satisfying (2) and (3), the hypersphere centered at y_j with radius $\lambda R(y_j)$ must be fully contained in the annulus region defined by $\|x\| + J(x, y_i)$ and $\|x\| - J(x, y_i)$. Additional another constraint in our method is as follows.

Let l be a reference line in the search space and it contains the unit vector $u = (1, 1, \ldots, 1)/\sqrt{k}$ on it. For any vector z, we define the angle between z and the reference vector u as:

$$\alpha_z = \cos^{-1} \frac{uz^T}{\|z\|}. \tag{10}$$

Because the values of all vector components are nonnegative, then the angle $\alpha \in [0, \frac{\pi}{4}]$. The angle α is called the projection angle to the reference line l. We define another angle between the input vector x and the tangent from the origin to the hypersphere centered at x with radius $J(x, y_i)$, where y_i is the current best codeword, as:

$$\theta_x = \sin^{-1} \frac{J(x, y_i)}{\|x\|}. \tag{11}$$

By the same way, we can define the angle between any codeword y_j and the tangent from the origin to the hypersphere centered at y_j with radius $\lambda R(y_j)$ as:

$$\theta_{y_j} = \sin^{-1} \frac{\lambda R(y_j)}{\|y_j\|}. \tag{12}$$

Figure 4 shows the geometrical interpretation of the angular constraint method in 2-dimensional case. For a given input vector x with its projection angle α_x to the reference line l and the closest codeword y_j with its projection angle α_{y_j}, the following inequalities should be satisfied:

$$\alpha_{y_j} + \theta_{y_j} < \alpha_x + \theta_x, \tag{13}$$

and

$$\alpha_{y_j} - \theta_{y_j} > \alpha_x - \theta_x. \tag{14}$$

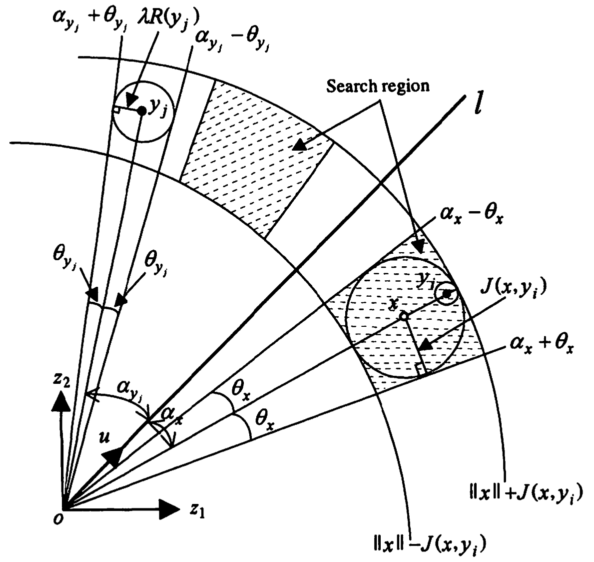


Fig. 4 Geometrical interpretation of angular constraint method in 2-dimensional case.

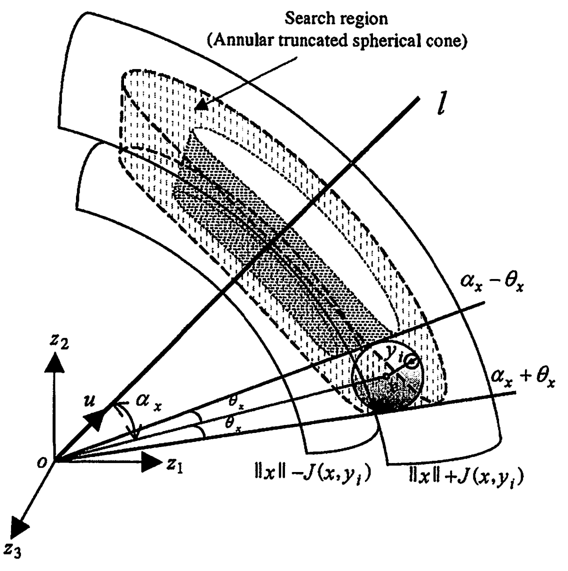


Fig. 5 Geometrical interpretation of angular constraint method in 3-dimensional case.

The inequalities (2), (3), (13) and (14) constrain the distortion calculation to the codeword whose hypersphere is completely contained in the search region shown in Fig. 4. The search region is expanded to an annular truncated spherical cone shown in Fig. 5 in which the geometrical interpretation of the angular constraint method in 3-dimensional case is represented. When $J(x, y_i) > \|x\|$, the angle θ_x can not be defined. This means that the hypersphere centered at x with radius $J(x, y_i)$ includes the origin in it. In this case, the search area is constrained into the hypersphere centered at the origin with radius $\|x\| + J(x, y_i)$ by the same way as in the annulus method. It is emphasized that the search based on our method is strictly equivalent to the full search and the obtained codebook is just the same to that by the full search.

Comparing the search region to that of Cardinal method in 2-dimensional case, the area of the dotted square in Fig. 2 is $4J^2(x, y_i)$. On the other hand, the search area of the above angular constraint method is $4(\sin\theta_x/\theta_x)^{-1}J^2(x, y_i)$ and slightly larger than that of Cardinal method. But, when θ_x is small, that means a good choice for the current best codeword and/or the input vector far from the origin from Eq. (11), both search areas are almost same. Moreover, considering that shapes of the search region are different and the search region of the angular constraint method is not fully contained in that of Cardinal method, codeword elimination for distortion calculation depends on codeword distribution. As a result, the angular constraint method may be superior to Cardinal method in some cases and inferior in other cases, but the difference of codeword elimination efficiency between both methods is expected to be small. Above consideration may be also valid for multidimensional cases.

Actually, the execution time needed in the search process is related to the computation of $J(x, y_i)$, the distortion associating with the best codeword y_i, so the choice of the first codeword to be tested is very serious issue of the search process. We can use the following idea: after applying the first iteration of the algorithm, the training vectors will be clustered with the initial codebook. Then the improved codebook will be generated by calculating the centroid of the training vectors of each cluster. However, for a training vector x grouped to index i in the previous iteration, $J(x, y_i)$ will be a small value even if y_i is a new codeword in the current iteration. At this stage, we should have a way to choose a better initial codeword y_i. This method was experimented with success in [13].

4. Angular Constraint with Hyperplane Decision Rule (Lossy Design)

Most nearest-neighbor search techniques employ searching the best codeword in the same search region for all training vectors. In this section, we introduce a technique using a hyperplane H to divide the signal space into two half-spaces according to the vector feature. This method has been tried with success in [19] for both Cardinal method and the double annulus method.

The chosen hyperplane H contains the centroid of the training vectors $x_c = (x_{c1}, x_{c2}, \ldots, x_{ck})$ and its projection point on the central line $x_p = (m_{x_c}, m_{x_c}, \ldots, m_{x_c})$, where m_{x_c} is the mean value of x_c. It is perpendicular to the central line as shown in Fig. 6 and can be expressed as:

$$H : uz^T = ux_c^T = \frac{1}{\sqrt{k}}\sum_{i=1}^{k} x_{c_i} = \sqrt{k}\, m_{x_c} = M_c. \tag{15}$$

This hyperplane H is used as a decision function that discriminates to which half-space a given vector x belongs by the following conditions:

$$\bullet \text{ If } \quad ux^T = \frac{1}{\sqrt{k}}\sum_{i=1}^{k} x_i < M_c, \tag{16}$$

then x belongs to the lower half-space.

- If $$ux^T = \frac{1}{\sqrt{k}} \sum_{i=1}^{k} x_i \geq M_c, \quad (17)$$

then x belongs to the upper half-space.

L. Guan and M. Kamel [23] studied the distribution of some images data and found that most images data vectors are located around the diagonal or the central line. Hence, only a small portion of vectors will be near to the chosen hyperplane H, then the possibility for the hypersphere centered at the input vector to cross over this hyperplane is small. As a result, failure in best codeword searching becomes to be less even if searching is performed in either half-space dependent on the input vector feature.

Now we depict the proposed method that uses the hyperplane H to separate both the training vectors and the codewords. The proposed method divides the training vectors into two sub-groups T_{lw} and T_{up}, and each sub-group contains the vectors satisfying (16) or (17), respectively. Also, it divides the codebook into two sub-codebooks Y_{lw}, and Y_{up} by the same equations. Searching for the training vectors in the sub-group T_{lw} is carried out in the sub-codebook Y_{lw} and for the training vectors in the sub-group T_{up} in the sub-codebook Y_{up}, by using the constraints in the inequalities (2), (3), (13) and (14). Hence, the proposed method can reduce the search area and speed up the search process.

The proposed method may be easily understood with the geometrical interpretation for 3-dimensional case in Fig. 6. This figure includes the proposed hyperplane H. The hyperplane H divides the signal space into two half-spaces, and each half-space includes its own training vectors and codewords.

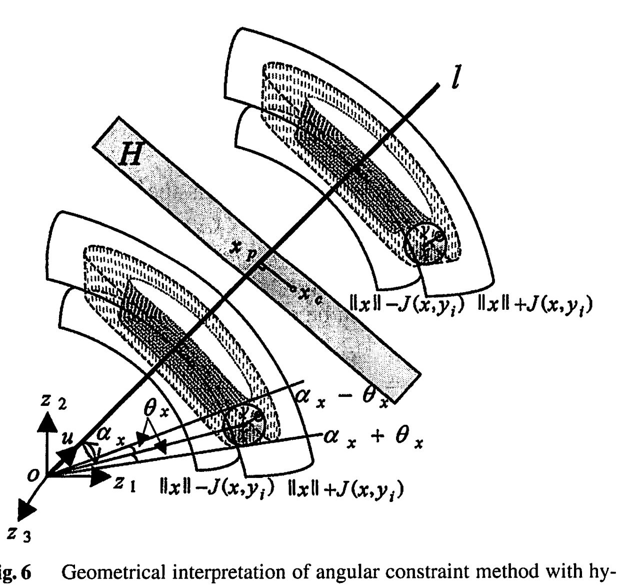


Fig. 6 Geometrical interpretation of angular constraint method with hyperplane decision rule in 3-dimensional case.

5. Experimental Results

Experiments were carried on vectors taken from the USC grayscale image set. We used two images, Lena and Baboon with size 512×512 and 256 gray levels. Each image was divided into 4×4 blocks, so the training set contains 16384 blocks. Both lossless design methods (the full search (FS), the double annulus (DA), Cardinal (CARD), the angular constraint (ANG)) and lossy methods (Cardinal with the hyperplane decision rule (CARDHP) in [19], the angular constraint with the hyperplane decision rule (ANGHP)) were tested.

Figures 7 and 8 show the PSNR of the ANG method and the ANGHP method, respectively, for different codebook sizes, N, at various values of $\lambda = 0.5, 2, 4$ and 8 with Lena image. And Fig. 9 shows the PSNR comparison between the ANG method and the ANGHP method for

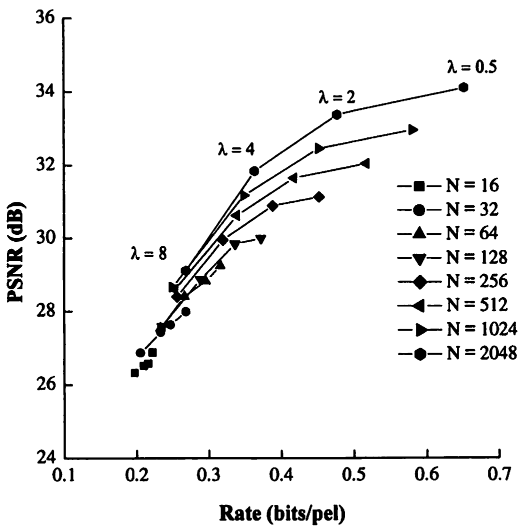


Fig. 7 PSNR of the ANG method for Lena.

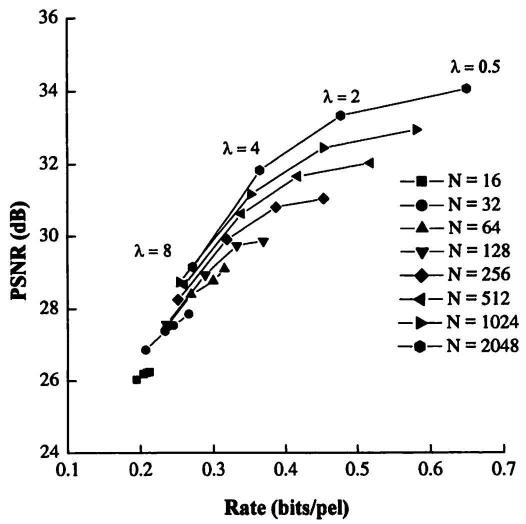


Fig. 8 PSNR of the ANGHP method for Lena.

N = 32, 256 and 1024 at various values of λ with Lena image. We want to insist on the fact that the ANG method has the same PSNR of the FS method (lossless design). Although the ANGHP method is a lossy design method, it has almost the same performance as the FS method at larger codebook size. For example, the performance of the ANGHP method is only 0.073 dB less than the FS method at codebook size 256 with λ= 0.5, and this value decreases by increasing the codebook size. There is a small degradation for smaller codebook size, for example, the ANGHP method has 0.141 dB less than the FS method at codebook size 32 with λ= 0.5. This is because the ANGHP method is not equivalent to the ANG method completely, and the best codeword happens to be in the other half-space and is missed to be searched out. However, there may be a small failure possibility in the case of large codebook size and smooth codebook distribution. Both the double annulus method with the hyperplane decision rule and the CARDHP method in [19] have the same PSNR of the ANGHP method.

Table 1 presents a comparison of execution time (in seconds) and the total number of distortion calculations (Ds) for codebook design at codebook size N= 256 and various values of λ for Lena and Baboon images. The timings were made on Pentium III (866 MHz). The ANG method significantly accelerates the codebook design compared to the FS method and the DA method in terms of the execution time and the total number of distortion calculations. It also has almost the same efficiency as the CARD method. Compared to the FS method, the average acceleration ratio of the ANG method is 7.2 for Lena and 3.5 for Baboon, and compared to the DA method, it is 1.2 for Lena and 1.1 for Baboon. The ANG method reduces the total number of distortion calculations for Lena by 95.6% and 25.4% less than the FS method and the DA method in average, respectively. Also, for Baboon it reduces this number by 81.4% and 5.4% less than the FS method and the DA method, respectively.

Figures 10 and 11 show comparisons of the execution time (in seconds) for Lena image with different codebook sizes at λ = 0.5 and 8, respectively. Compared to the DA method, the ANG method reduces the execution time by 12.7% to 23.2%. The execution time of the ANG method is quite close to that of the CARD method. The speed-up achieved by the ANGHP method ranges from 16.2% to

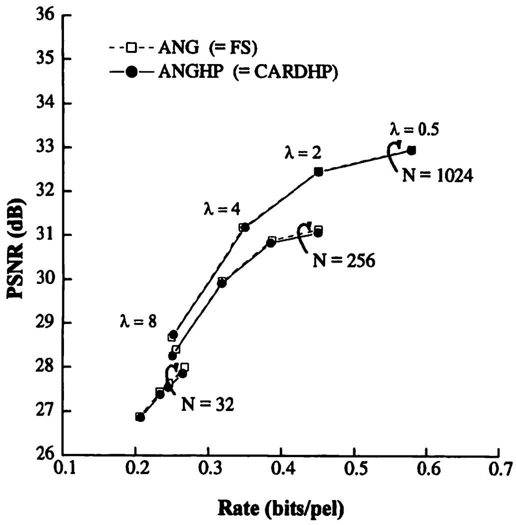


Fig. 9 Comparison of PSNR for Lena.

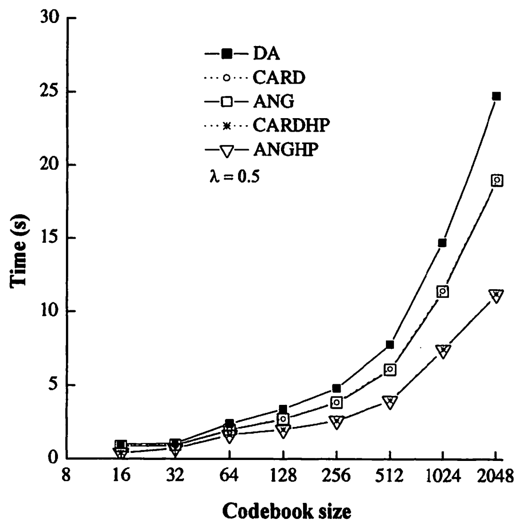


Fig. 10 Comparison of execution time at λ = 0.5 for Lena.

Table 1 Comparison of execution time and total number of distortion calculations (Ds) at codebook size 256.

		FS		DA		CARD		ANG	
Tested image	λ	Time (s)	Ds	Time (s)	Ds	Time (s)	Ds	Time (s)	Ds
Lena	0.5	27.586	33554432	4.820	2220518	3.867	1571992	3.845	1543178
	2	25.281	30867456	4.250	1773273	3.445	1303026	3.325	1276117
	4	18.344	22233088	3.109	1259261	2.609	987607	2.531	962216
	8	14.141	17235968	2.570	1008534	2.172	835976	2.107	811613
Baboon	0.5	20.656	25165824	7.047	5509069	6.195	5030076	6.231	5076475
	2	20.586	25165824	6.898	5384943	6.175	4978999	6.211	5038996
	4	23.727	28983296	7.516	5714961	6.805	5381459	6.878	5471404
	8	22.656	27656192	6.211	4288138	5.602	4067946	5.686	4162554

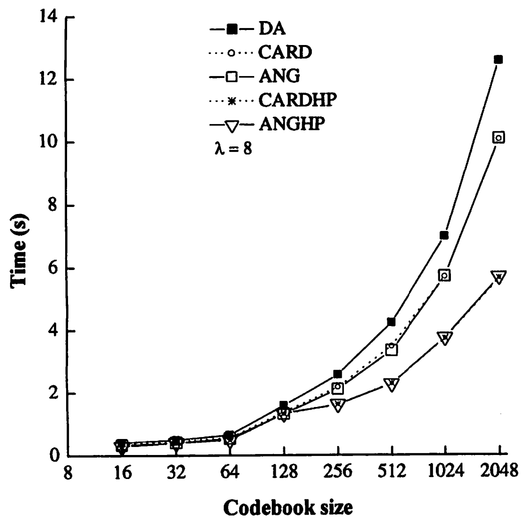


Fig. 11 Comparison of execution time at $\lambda = 8$ for Lena.

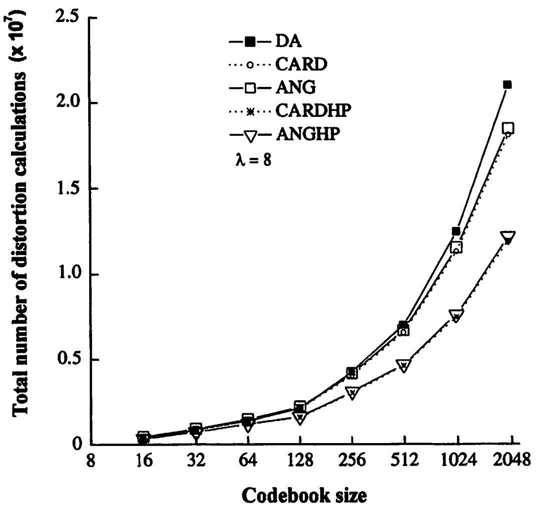


Fig. 13 Comparison of total number of distortion calculations at $\lambda = 8$ for Baboon.

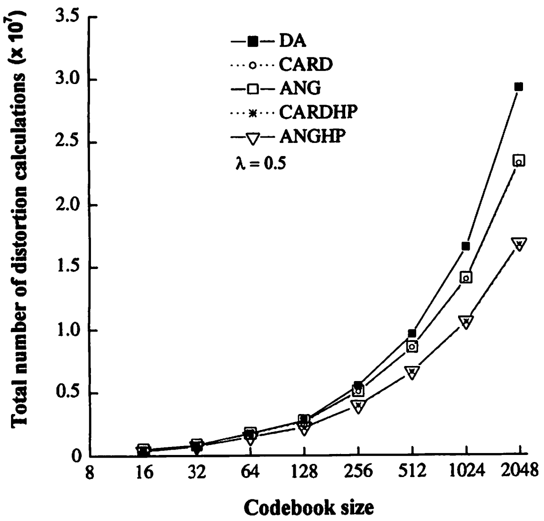


Fig. 12 Comparison of total number of distortion calculations at $\lambda = 0.5$ for Baboon.

42.9% compared to the ANG method. The two curves of the ANGHP method and the CARDHP method are nearly coincident. It can be seen that as the codebook size increases, the efficiency of the ANGHP becomes better than the ANG method. This is an important merit of the ANGHP method, because design of a larger codebook size requires more intensive computation.

Figures 12 and 13 present the total number of distortion calculations, which is a dominant figure of the computational complexity, for Baboon image with different codebook sizes at $\lambda = 0.5$ and 8, respectively. Compared to the DA method, the ANG method reduces the Ds by 2.9% to 20.1%. The total number of distortion calculations of the ANG method is almost the same as that of the CARD method. The Ds curve of the ANGHP method is also almost the same as that of the CARDHP method. Compared to the ANG method at $\lambda = 0.5$ and 8 with different codebook sizes, the ANGHP method reduces the Ds by 16.8% to 43.1% in the case of Lena image, and 12.4% to 27.9% in the case of Baboon image. From those results, only a small number of distortion calculations are carried out in the ANGHP method.

6. Conclusions

In this paper, we have proposed two new algorithms of accelerating the codebook design for ECVQ. The first algorithm (ANG) uses a new constraint called angular constraint. The ANG method employs the projection angles of the vectors to a reference line in the signal space, and achieves equivalent performance to full search ECVQ. It accelerates the codebook design process significantly compared with the full search method and the double annulus method, and has almost the same efficiency as Cardinal method.

The second algorithm uses a hyperplane decision technique for separating the training vectors and the codebook into two sub-groups, and carries on searching within one sub-group according to the vector feature. By applying this algorithm to the ANG method, the ANGHP method is developed. Using Lena and Baboon images with different codebook sizes at $\lambda = 0.5$, 2, 4 and 8 and compared with the ANG method, the ANGHP method attains acceleration range from 7.3% to 42.9% and reduces the total number of distortion calculations by 8.1% to 43.1%. Furthermore, the performance of the ANGHP method is quite close to that of the FS method.

References

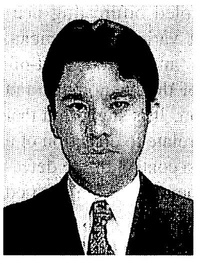
[1] A. Gersho and R.M. Gray, Vector quantization and signal compression, Kluwer, Boston, 1991.

[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, Jan. 1980.

[3] T. Berger, "Minimum entropy quantizers and permutation codes," IEEE Trans. Inf. Theory, vol.IT-28, no.2, pp.149–157, 1982.

[4] N. Farvardin and J. Modestino, "Optimum quantizer performance for a class of nongaussian memoryless sources," IEEE Trans. Inf. Theory, vol.IT-30, no.3, pp.485–497, 1984.
[5] P.A. Chou, T. Lookabough, and R.M. Gray, "Entropy-constrained vector quantization," IEEE Trans. Acoust. Speech Signal Process., vol.37, no.1, pp.31–42, Jan. 1989.
[6] R.P. Rao and W.A. Pearlman, "Alphabet-constrained vector quantization," IEEE Trans. Inf. Theory, vol.39, no.4, pp.1167–1179, 1993.
[7] D.P. de Garrido, W.A. Pearlman, and W.A. Finamore, "A clustering algorithm for entropy-constrained vector quantizer design with applications in coding image pyramids," IEEE Trans. Circuits Syst. Video Technol., vol.5, no.2, pp.83–95, April 1995.
[8] F. Kossentini, M.J.T. Smith, and C.F. Barnes, "Entropy constrained residual vector quantization," Proc. IEEE ICASSP, pp.598–601, April 1993.
[9] M. Lightstone and S.K. Mitra, "Entropy-constrained mean-gain-shape vector quantization for image compression," Proc. SPIE Symp. Visual Commun. Image Processing, Chicago, IL, vol.2308, pp.389–400, Sept. 1994.
[10] M.D. Orchard, "A fast nearest-neighbor search algorithm," Proc. ICASSP-91 International Conference on Acoust. Speech and Signal Process., vol.4, pp.2297–2300, April 1991.
[11] C.M. Huang, Q. Bi, G.S. Stiles, and ZR.W. Harris, "Fast full search equivalent encoding algorithms for image compression using vector quantization," IEEE Trans. Image Process., vol.1, no.3, pp.413–416, July 1992.
[12] C.-H. Lee and L.-H. Chen, "Fast closest codeword search algorithm for vector quantization," IEE Proc. Vis. Image Signal Process., vol.141, no.3, pp.143–148, June 1994.
[13] J.Z.C. Lai and C.C. Lue, "Fast search algorithms for VQ codebook generation," J. Vis. Commun. Image Represent., vol.7, no.2, pp.163–168, June 1996.
[14] M.H. Johnson, R. Ladner, and E.A. Riskin, "Fast nearest neighbor search for ECVQ and other modified distortion measures," Proc. ICIP-96, vol.3, pp.423–426, Sept. 1996.
[15] J. Cardinal, "Fast search for entropy-constrained VQ," Proc. ICIAP-99 the 10th International Conference on Image Analysis and Process., pp.1038–1042, Sept. 1999.
[16] K.-S. Wu and J.-C. Lin, "Fast VQ encoding by an efficient kick-out condition," IEEE Trans. Circuits Syst. Video Technol., vol.10, no.1, pp.59–62, Feb. 2000.
[17] M.H. Johnson, R. Ladner, and E.A. Riskin, "Fast nearest neighbor search of entropy-constrained vector quantization," IEEE Trans. Image Process., vol.9, no.8, pp.1435–1437, Aug. 2000.
[18] A. Swilem, K. Imamura, and H. Hashimoto, "A fast search algorithm for vector quantization using hyperplane decision rule," J. Institute of Image Information and Television Engineers, vol.56, no.9, pp.1513–1517, Sept. 2002.
[19] A. Swilem, K. Imamura, and H. Hashimoto, "A fast search algorithm for ECVQ based on hyperplane partitioning rule," Proc. IEEE INES 2003 7th International Conference on Intelligent Engineering Systems, vol.2, pp.704–709, March 2003.
[20] C.D. Feustel and L.G. Shapiro, "The nearest neighbor problem in an abstract metric space," Pattern Recognit. Lett., vol.1, pp.125–128, Dec. 1982.
[21] E. Vidal, "An algorithm for finding nearest neighbor in (approximately) constant average time," Pattern Recognit. Lett., vol.4, pp.145–157, July 1986.
[22] J. Cardinal, "Fast search techniques for product code vector quantizers," Proc. ECMCS-99 the 2nd EURASIP Conference on DSP for Multimedia Commun. and Services, June 1999.
[23] L. Guan and M. Kamel, "Equal-average hyperplane partitioning method for vector quantization of image data," Pattern Recognit. Lett., vol.13, no.10, pp.693–699, Oct. 1992.

Ahmed Swilem received B.S., M.S. degree in computer science from Minia University (Egypt), in 1992 and 1998, respectively. During 1998–2001, he was an assistant lecture in Computer Science Department, Minia University. He is now a doctoral student in Graduate School of Natural Science & Technology, Kanazawa University. His research interests include vector quantization, data compression and image processing.



Kousuke Imamura received the B.S., M.S. and Dr. Eng. degrees in Electrical Engineering and Computer Science in 1995, 1997 and 2000, respectively, all from Nagasaki University. He has been a research assistant of Information and Systems Engineering, Kanazawa University. His research interests are high efficiency image coding and image processing.

Hideo Hashimoto received the B.S., M.S. and Dr. Eng. degrees in Electronic Engineering in 1968, 1970 and 1975, respectively, all from Osaka University. He joined Electrical Communication Laboratories of Nippon Telegraph and Telephone Corporation (NTT) in 1975. Since 1993, he has been a professor of Information and System Engineering, Kanazawa University. His research interests are video coding, moving object segmentation and visual communication.