Fast codeword search algorithm for ECVQ using hyperplane decision rule

FAST CODEWORD SEARCH ALGORITHM FOR ECVQ USING HYPERPLANE DECISION RULE

K. Imamura, A. Swilem, H. Hashimoto
Kanazawa University, Kanazawa, Japan

ABSTRACT

Vector quantization is the process of encoding vector data as an index to a dictionary or codebook of representative vectors. One of the most serious problems for vector quantization is the high computational complexity involved in searching for the closest codeword through the codebook. Entropy-constrained vector quantization (ECVQ) codebook design based on empirical data involves an expensive training phase in which Lagrangian cost measure has to be minimized over the set of codebook vectors. In this paper, we describe a new method allowing significant acceleration in codebook design process. This method has feature of using a suitable hyperplane to partition the codebook and image data. Experimental results are presented on image block data. These results show that our method performs better than previously known methods.

1. INTRODUCTION

A standard vector quantization (VQ) [1] is an efficient compression technique for which many variants [2] are known. It is defined as a mapping Q from a k-dimensional Euclidean space R^k to a finite set $Y = \{y_1, y_2, ..., y_N\}$ 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) [1]. This codebook is used in both the encoder and the decoder. The encoding phase is equivalent to finding 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) [3] uses 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 it is 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 $P(y_i)$ of the codeword y_i is approximated empirically using the training set. We define the cost function of the codeword y_i for encoding the vector x as the Lagrangian function,

$$J(x, y_i) = d(x, y_i) + \lambda R(y_i)\,, \qquad (1)$$

where d 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.

The computational cost of finding the best suitable codeword in the codebook design and encoding imposes practical limits on the codebook size N. When N becomes larger, the computational complexity problem for full codebook search occurs. To avoid such an exhaustive search through the codebook, many fast algorithms [4]-[7] for standard VQ and [8]-[10] for ECVQ have been proposed. These algorithms reduce the computational complexity by performing some simple tests before computing the distortion between the training vector and each codeword, and then rejecting those codewords that fail in the tests. The better known acceleration methods for nearest neighbor search for ECVQ are the double annulus method [8] and Cardinal method [9]. And the algorithm speeding up the nearest neighbor search based on the hyperplane partitioning technique for standard VQ has been presented in [11].

This paper introduces a new algorithm to reduce the time complexity of the codebook search using a hyperplane partitioning rule for ECVQ. The idea of the proposed algorithm relies on the separation of the codebook and the training vectors into two parts, and searching in only one part according to the vector feature. Two methods, based on double annulus method [8] and Cardinal method [9], are developed by the proposed algorithm. The efficiency of the developed methods is compared with the basic double annulus method and Cardinal method.

The paper is organized as follows. Section 2 reviews the double annulus method and Cardinal method for ECVQ. Section 3 describes the hyperplane decision method. Section 4 presents the fast algorithm that uses the hyperplane partitioning rule for ECVQ. Experimental results are shown in section 5, and concluding remarks are given in section 6.

2. FAST ALGORITHMS FOR ECVQ

2.1 Double Annulus Method

Johnson et al. [8] 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 the 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:

$$\| y_j \| + \lambda R(y_j) < \| x \| + J(x, y_i)\,, \qquad (2)$$

and

$$\| y_j \| - \lambda R(y_j) > \| x \| - J(x, y_i)\,, \qquad (3)$$

where $\| y_j \|$ is the Euclidean distance of y_j from the origin, and $R(y_j)$ is the codeword length of y_j. Thus, by the inequalities in (2) and (3), only the codewords within the annulus defined by $\| x \| + J(x, y_i)$ and $\| x \| - J(x, y_i)$ will be searched.

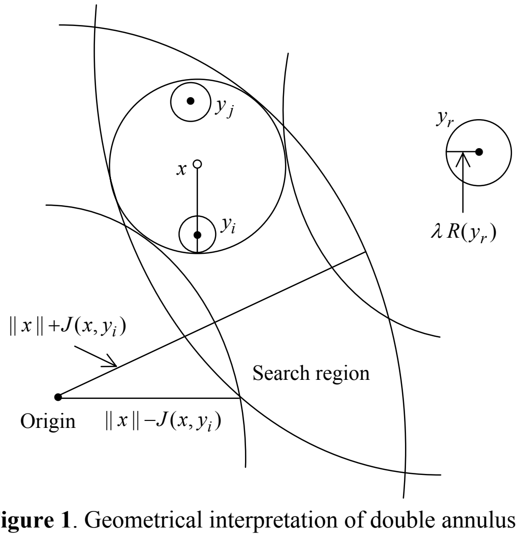

Figure 1. Geometrical interpretation of double annulus method in 2-dimensional case.

0-7803-7762-1/03/$17.00 ©2003 IEEE

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:

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

and

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

The inequalities in (2), (3), (4) and (5) constrain distortion calculations to those codewords completely contained in the search region shown in Fig. 1.

2.2 Cardinal Method

Cardinal [9] introduced the most acceleration method for GLA on ECVQ using two elimination rules. In the first elimination rule, an unit vector $u = (1,1,...,1)/\sqrt{k}$, where k is the vector dimension, on the central line is used as a reference line as shown in Fig. 2. For a given input vector x and the 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:

$$u\,y_j^T + \lambda R(y_j) < u\,x^T + J(x, y_i)\,, \qquad (6)$$

and

$$u\,y_j^T - \lambda R(y_j) > u\,x^T - J(x, y_i)\,. \qquad (7)$$

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 this method 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 $L_1 : u\,z^T = u\,x^T + J(x, y_i)$ and $L_2 : u\,z^T = u\,x^T - J(x, y_i)$.

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:

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

and

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

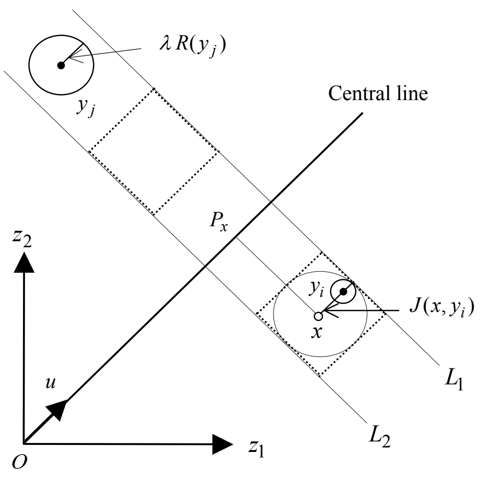

Figure 2. Geometrical interpretation of Cardinal method in 2-dimensional case.

By using the constrains 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. Every codeword whose sphere is not contained in this region is eliminated. Cardinal method is considered as the generalization of Lee and Chen method [6] to ECVQ.

3. HYPERPLANE DECISION METHOD

The search region in Fig. 2 is reduced to the two dotted squares. The same situation occurs in the Lee and Chen method that uses the distortion as the squared Euclidean distance. Lee and Chen introduced an elimination method [6], which uses the mean and the variance of the vector for two tests to reject the codewords. In the mean test, an unit vector u on the central line is used. For a given input vector x with mean value m_x, and a current best codeword y_i with distance $d_{\min} = d(x, y_i)$, any codeword that is closer to x than y_i has to be located inside the hypersphere centered at x with radius $d_{\min}$. Two boundary points $L_{\max} = (m_{\max}, m_{\max}, ..., m_{\max})$ and $L_{\min} = (m_{\min}, m_{\min}, ..., m_{\min})$ can be obtained by projecting the hypersphere on the central line, where

$$m_{\max} = m_x + d_{\min}/\sqrt{k}\,, \qquad (10)$$

and

$$m_{\min} = m_x - d_{\min}/\sqrt{k}\,. \qquad (11)$$

Thus, by (10) and (11) only the codewords that are bounded by the two hyperplanes $S_1 : u\,z^T = m_x + d_{\min}/\sqrt{k}$ and $S_2 : u\,z^T = m_x - d_{\min}/\sqrt{k}$ will be searched.

In the variance test, the squared root of variance of the vector x, v_x, is used as the distance $d(x, L_x)$ between x and its projection point L_x on the central line. The closest codeword y_j with squared root of variance v_{y_j} will satisfy the following inequality:

$$(v_x - v_{y_j})^2 < d_{\min}^2\,. \qquad (12)$$

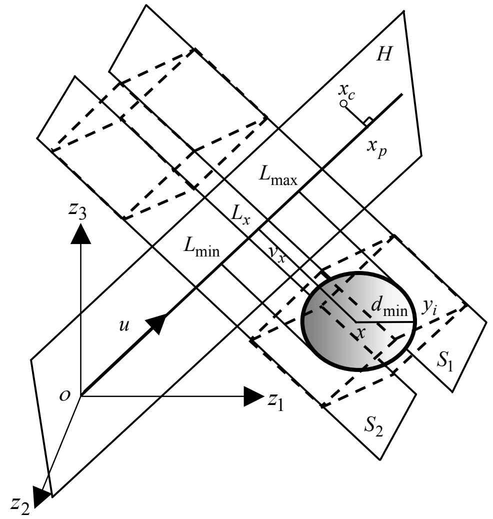

Figure 3. Geometrical interpretation of hyperplane decision method in 3-dimensional case.

However, if the two search areas that are shown by the two dotted cubes in Fig. 3 are separated, the search area will be reduced to one dotted cube only, and the computation complexity may be reduced to around half. Fig. 3 does not show search areas in 3-dimensional case correctly, but it is used to depict the basic idea of search areas for easily understanding, as the extension of 2-dimensional case in Fig. 2. We introduced a technique for separating the codebook and searching in one side area according to the input vector feature for Lee and Chen method in [11]. This method succeed to reduce the time complexity with almost same performance of Lee and Chen method.

The nearest codeword for an input vector belongs to one of two search areas shown in Fig. 3. If this relation is known before the codeword searching, the search area can be reduced. Although a perfect identification of the search area for all input vectors is difficult, a probable and reasonable separation is possible when the codebook size is relatively large and codewords distribution in the signal space is smooth.

The chosen hyperplane H, which separates the codebook into two parts, contains the origin o, the centroid of the input vectors $x_c = (x_{c1}, x_{c2}, ..., x_{ck})$ and the projection point of the centroid on the central line $x_p = (x_{p1}, x_{p2}, ..., x_{pk})$, where

$$x_{p1} = x_{p2} = ... = x_{pk} = 1/\sqrt{k}\sum_{i=1}^{k} x_{ci} .$$

This hyperplane is expressed as

$$H : h\, z^T = 0 , \qquad (13)$$

where h is the normal vector to the hyperplane H. The hyperplane H is used as a decision function that discriminates to which half-space a given vector x belongs using the following conditions:

- If $h x^T < 0$, then x belongs to the lower half-space separated by H.
- If $h x^T \geq 0$, then x belongs to the upper half-space.

H is not adequate for this method to be applied directly to Cardinal method for ECVQ, because the training vectors and codewords are densely distributed close to the central line. In this case, the hypersphere centered at the input vector x with radius $J(x, y_i)$ may cross the central line, then the best codeword searching will fail because of larger possibility for the closest codeword to be in the other half space.

4. FAST ALGORITHM FOR ECVQ USING HYPERPLANE PARTITIONING RULE

We will change the hyperplane H described in the last section to be perpendicular to the central line as shown in Fig. 4. This hyperplane containing the centroid of the training vectors, x_c, on it can be expressed as

$$G : u\, z^T = u\, x_c^T = 1/\sqrt{k}\ \sum_{i=1}^{k} x_{ci} = \sqrt{k}\, m_{x_c} = M_c . \qquad (14)$$

The hyperplane G discriminates to which half-space a given vector x belongs by the following conditions:

- If $u x^T = 1/\sqrt{k}\sum_{i=1}^{k} x_i < M_c$, (15)
  then x belongs to the lower half-space.
- If $u x^T = 1/\sqrt{k}\ \sum_{i=1}^{k} x_i \geq M_c$, (16)
  then x belongs to the upper half-space.

From the distribution of the training vectors and codewords, only a small number of vectors will be near to the chosen hyperplane, then the possibility of the hypersphere centered at the input vector crossing over this hyperplane is reduced. As a result, failure in best codeword searching becomes to be less.

Now we depict the proposed algorithm that uses the hyperplane G to separate the training vectors and the codebook. The proposed algorithm divides the training vectors into two sub-groups T_{lw} and T_{up}, which contain the vectors that satisfy (15) and (16), respectively. Also it divides the codebook into two sub-codebooks Y_{lw} and Y_{up} by the same equations. The training vectors in the sub-group T_{lw} will be searched in the sub-codebook Y_{lw}, and the training vectors in the sub-group T_{up} will be searched in the sub-codebook Y_{up}. Hence, the proposed algorithm can reduce the search area and speed up the search process.

The application of the proposed algorithm to Cardinal method may be easily understood with the geometrical interpretation for 3-dimensional case in Fig. 4. This figure is the extension of 2-dimensional case in Fig. 2 and includes the proposed hyperplane G. The hyperplane G divides the signal space into two half-spaces, and each half-space has its own training vectors and codewords.

Actually, the time needed by the search in each half-space 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 the 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, if it is 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 [12].

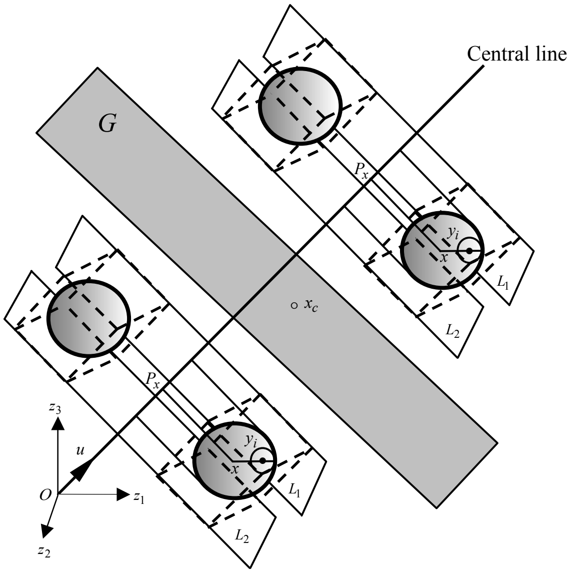

Figure 4. Geometrical interpretation of the proposed method in 3-dimensional case.

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 is divided into 4×4 blocks, so the training set contains 16384 blocks. The tested methods are Cardinal method, double annulus method, and the proposed method applied to the double annulus method (AI) and Cardinal method (AII). The PSNR of the proposed method (AII) and its comparison to Cardinal method are shown in Figs. 5 and 6, respectively, for different codebook sizes, N, at various values of λ = 0.5, 2, 4 and 8 with Lena image. We want to insist on the fact that the search based on our method in AI is strictly equivalent to AII search. This means the outputs are exactly the same. The proposed method has almost the same performance of Cardinal method at higher codebook size. For example, the degradation of our method is only 0.073 dB more than Cardinal method at codebook size 256 with λ = 0.5, and this value decreases by increasing the codebook size. There is small degradation for smaller codebook size, for example, our method has 0.141 dB less than Cardinal method at codebook size 32 with λ = 0.5. This is because the proposed method is not equivalent to Cardinal 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. Figs. 7 and 8 present the time execution for the four methods with various codebook sizes at λ = 0.5 for Lena and Baboon, respectively. The timings were made on Pentium III (866 MHZ). We can see that our new methods AI and AII significantly accelerate the codebook design for double annulus method and Cardinal method. Compared to the double annulus method, AI reduces the time by 20% to 43.6% for Lena and 12.7% to 31.7% for Baboon. And compared to the Cardinal method, AII reduces the time by 16.7% to 37.9% for Lena and 10.8% to 27.9% for Baboon. As we know, the total number of distortion calculations is a dominant figure of the computational complexity. AI reduces this number by 23.2% to 44.5% for Lena and 14.3% to 32.7% for Baboon than the double annulus method. Also, AII reduces this number by 17.5% to 38.2% for Lena and 11.7% to 28.4% for Baboon than Cardinal method. From the last results, AI and AII are faster at a large codebook than a small codebook.

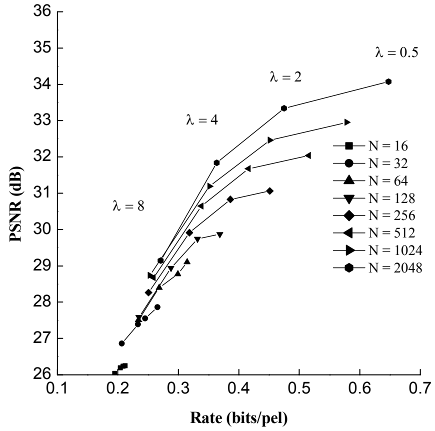

Figure 5. PSNR of the proposed method (AII) for Lena.

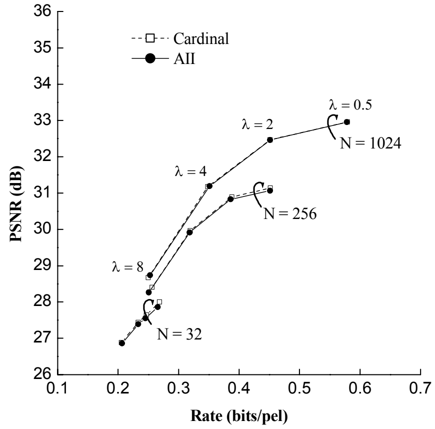

Figure 6. Comparison of PSNR for Lena.

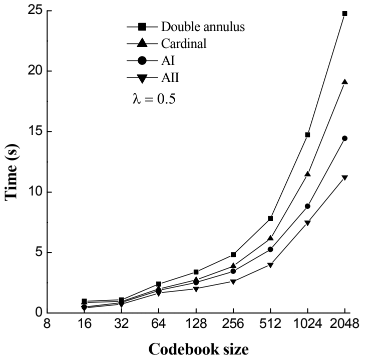

Figure 7. Comparison of time at λ =0.5 for Lena.

6. CONCLUSIONS

In this paper, we have presented a new algorithm of accelerating the codebook design for ECVQ. The proposed algorithm uses a hyperplane decision technique for separating the training vectors and the codebook into two groups, and employs searching in one group according to the vector feature. By applying this algorithm to double annulus method and Cardinal method, two methods AI and AII are developed, respectively. Compared to double annulus method, the obtained results show that AI allows acceleration range from 12.7% to 43.6% and reduces the total number of distortion calculations by 14.3% to 44.5%. And compared to the Cardinal method, AII allows acceleration range from 10.8% to 37.9% and reduces the total number of distortion calculations by 11.7% to 38.2%. Furthermore, the performance of the proposed methods are quite close to the performance of Cardinal method.

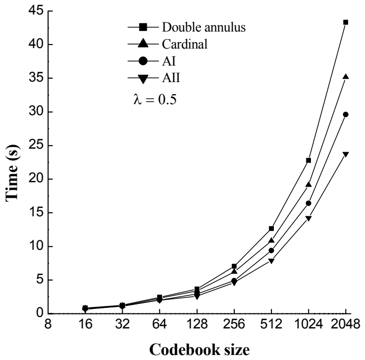

Figure 8. Comparison of time at λ = 0.5 for Baboon.

7. REFERENCES

[1] Y. Linde, A. Buzo and R. M. Gray, "An algorithm for vector quantizer design," IEEE Trans. Commun., vol. COM-28, Jan. 1980, pp. 84-95.

[2] Special issue on vector quantization, IEEE Trans. Image Process., vol. 5, no. 2, Feb. 1996.

[3] P. A. Chou, T. Lookabough and R. M. Gray, "Entropy-constrained vector quantization," IEEE Trans. Acoust., Speech, Signal Process., vol. 37, Jan. 1989, pp. 31-42.

[4] M. D. Orchard, "A fast nearest-neighbor search algorithm," in Proceedings of ICASSP 91:1991 International Conference on Acoust., Speech and Signal Process., vol. 4, April 1991, pp. 2297-2300.

[5] L. Guan and M. Kamel, "Equal-average hyperplane partitioning method for vector quantization of image data," Pattern Recognition Lett., vol. 13, no. 10, Oct. 1992, pp. 693-699.

[6] 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, June 1994, pp. 143-148.

[7] 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, Feb. 2000, pp. 59-62.

[8] M. H. Johnson, R. Ladner and E. A. Riskin, "Fast nearest neighbor search for ECVQ and other modified distortion measures," in Proceedings of ICIP-96, vol. 3, Sept. 1996, pp. 423-426.

[9] J. Cardinal, "Fast search for entropy-constrained VQ," in Proceedings of ICIAP-99: the 10th International Conference on Image Analysis and Process., Sept. 1999, pp. 1038-1042.

[10] M. H. Johnson, R. Ladner, and E. A. Riskin, "Fast nearest neighbor search of entropy-constrained vector quantization," IEEE Trans. Image Process., vol. 9, Aug. 2000, pp. 1435-1437.

[11] A. Swilem, K. Imamura and H. Hashimoto, "A fast search algorithm for vector quantization using hyperplane decision rule," J. of the Institute of Image Information and Television Engineers, vol. 56, no. 9, Sept. 2002, pp. 1513-1517.

[12] J. Z. C. Lai and C. C. Lue, "Fast search algorithms for VQ codebook generation," J. of Vis. Commun. and Image representation, vol. 7, no. 2, June 1996, pp. 163-168.