Fast search algorithms for ECVQ using projection pyramids and variance of codewords

FAST SEARCH ALGORITHMS FOR ECVQ USING PROJECTION PYRAMIDS AND VARIANCE OF CODEWORDS

A. Swilem, K. Imamura, H. Hashimoto

Kanazawa University, Kanazawa, Japan

ABSTRACT

Vector quantization for image compression requires expensive time to find the closest codeword through the codebook. Codebook design based on empirical data for entropy–constrained vector quantization (ECVQ) involves a time consuming training phase in which a Lagrangian cost measure has to be minimized over the set of codebook vectors. In this paper, we propose two fast codebook generation methods for ECVQ. In the first one, we use an appropriate topological structure of input vectors and codewords to reject many codewords that are impossible to be candidates for the best codeword. In the second method, we use the variance test to increase the ability of the first algorithm to reject more codewords. These algorithms allow significant acceleration in the 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.

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 $C = \{y_0, y_1, ..., y_{N-1}\}$ of vectors in R^k called the codebook. A vector quantization process includes three phases: codebook design, encoding and decoding. The objective of codebook design is to construct a codebook C 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 $Q(x) = y_i \in C$ 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] employs a modified cost measure using both the effective distortion of the signal and the expected length of the transmitted code. We define the cost function for encoding the vector x by the codeword y_i as the Lagrangian function,

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

where λ is a constant called the Lagrange multiplier allowing to control the rate-distortion ratio and $R(y_i)$ is the length of the codeword y_i.

The computational cost of finding the best suitable codeword in the codebook design and encoding imposes practical limits on the codebook size *N* and the vector dimension *k*. When *N* and *k* become larger, the computational complexity problem occurs for full codebook search. This has motivated developing many fast nearest neighbor search algorithms [4]-[11]. Many algorithms concentrate on narrowing the area of the candidate codewords for which distortion must be calculated. For example, Lee and Chen [4] introduced one of such algorithms for standard VQ, which uses the mean and the variance of the vector to reduce the search area. Cardinal [6] generalized the technique in [4] to apply it to ECVQ. To constrain the search area, Cardinal method uses two elimination rules; the first rule utilizes the projection of the vector on a central line in the signal space, while the second rule employs the distance between the vector and its projection point on the central line.

Other algorithms exploit the topological structure of the codebook to avoid unnecessary codeword matching procedure. Lee and Chen [9] proposed a fast search algorithm for the codebook generation of standard VQ based on mean pyramids of codewords. Pan el al. [10] improved the encoding search process by adopting variance pyramids in addition to mean pyramids. Another technique using L$_2$-norm pyramids of codewords has been proposed in [11].

This paper proposes two methods to reduce the complexity in the codebook design for ECVQ. The proposed methods are developed by combing the idea of projection pyramid data structure with narrowing the search area by Cardinal method [6]. In the first method, we derive a multilevel inequality based on the projection pyramids of codewords. By employing this inequality, the codebook design process can be speeded up. In the second method, we add a variance test to the first algorithm to reject more codewords and speed up the codebook design process. These methods have the same coding quality as the full search method.

The paper is organized as follows. Section 2 describes the projection pyramid data structure. Section 3 introduces the proposed algorithms in detail. Experimental results are shown in section 4, and concluding remarks are given in section 5.

2. PROJECTION PYRAMID DATA STRUCTURE

Image pyramid data structure was originally developed for image coding by Burt and Adelson [12]. In this data structure, an image is represented hierarchically, with each level corresponding to a reduced-resolution approximation. Given an image Z^m of size $2^m \times 2^m$, its pyramid can be defined as a sequence of matrices $\{Z^0, ..., Z^{r-1}, Z^r, Z^{r+1}, ..., Z^m\}$, where an image Z^{r-1} in level $r-1$ has a size of $2^{r-1} \times 2^{r-1}$ and is a reduced-resolution version of Z^r. Note that Z^0 has only one pixel. A pyramid data structure can be formed by successively performing appropriate operations over 2×2 neighboring pixels in the next lower level. Therefore, the value of a pixel $z^{r-1}(i, j)$ in level $r-1$ can be obtained by

$$z^{r-1}(i, j) = f(\ z^r(2i, 2j), z^r(2i+1, 2j), z^r(2i, 2j+1), z^r(2i+1, 2j+1)\ ), \qquad (2)$$

0-7803-8251-X/04/$17.00 ©2004 IEEE

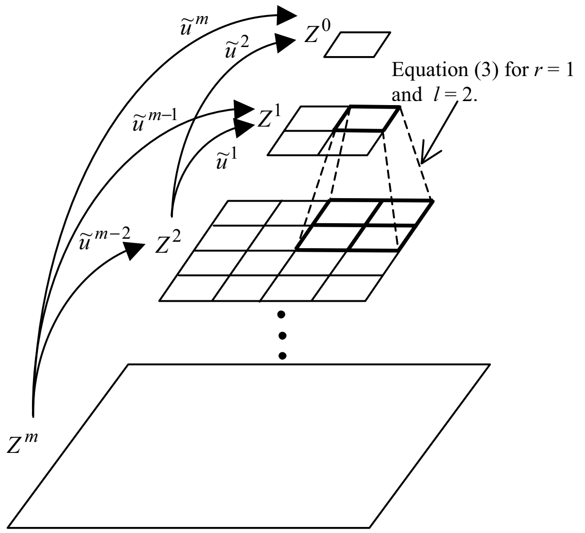

Fig. 1 The projection pyramid structure.

where f is an operation function.

There are many different types of image pyramids. The simplest pyramid data structure is the projection pyramid, which is formed by successively projecting the corresponding 2×2 neighboring pixels on a unit vector $\tilde{u}^1 = (1,1,1,1)/2$ in the 2×2-dimensional signal space. This kind of pyramid structure is equivalent to the double mean pyramid structure. In the projection pyramid, the bottom level is the original image itself, and the top level corresponds to the projection of the whole image on a unit vector $\tilde{u}^m = (1,1,...,1)/2^m$ in the $2^m \times 2^m$-dimensional space. Fig. 1 shows the projection pyramid structure. The projection pyramid pixel value $z^r(i,j)$ in level r is obtained from any lower level l as

$$
\begin{aligned}
z^r(i,j) &= \frac{1}{2^{l-r}} \sum_{p=0}^{2^{l-r}-1} \sum_{q=0}^{2^{l-r}-1} z^l(2^{l-r} i + p, 2^{l-r} j + q) \\
&= \tilde{u}^{l-r} \, (z^l(2^{l-r} i, 2^{l-r} j), z^l(2^{l-r} i + 1, 2^{l-r} j), ..., \\
&\qquad z^l(2^{l-r} i + 2^{l-r} - 1, 2^{l-r} j + 2^{l-r} - 1) \,)^T, \qquad (3)
\end{aligned}
$$

where $\tilde{u}^{l-r} = (1,1,...,1)/2^{l-r}$ is a unit vector in $2^{l-r} \times 2^{l-r}$-dimensional space and T denotes transpose.

3. FAST CODEBOOK DESIGN ALGORITHMS

In this section, we describe two high-speed closest-codeword search algorithms by deriving a robust inequality for ECVQ based on the projection pyramids of codewords. This inequality is used in the first proposed algorithm to reject many codewords and speed up the search process. In the second algorithm, we exploit the variance test for rejecting more codewords in the top level of the projection pyramids.

3.1. Projection Pyramid Algorithm

Before describing the method, we give some definitions and a lemma. Let $x = (x_0, x_1, ..., x_{k-1})$ be a k-dimensional vector, $y_i = (y_{i0}, y_{i1}, ..., y_{ik-1})$ be a codeword and $u = (1,1,...,1)/\sqrt{k}$ be a k-dimensional unit vector. Define the distortion $d(x, y_i)$ for representing the vector x by the codeword y_i as the Euclidean distance between x and y_i, that is,

$$d(x, y_i) = \sqrt{\sum_{j=0}^{k-1} (x_j - y_{ij})^2} \, . \qquad (4)$$

Lemma 1:

$$d(x, y_i) \;\; \geq \; \frac{1}{\sqrt{k}} \sum_{j=0}^{k-1} | \, x_j - y_{ij} \, | \, . \qquad (5)$$

Proof: For any real values $\alpha_0, \alpha_1, ..., \alpha_{k-1}$, we have

$$\Big(\sum_{j=0}^{k-1} |\alpha_j| \Big)^2 = (|\alpha_0| + |\alpha_1| + ... + |\alpha_{k-1}|)^2 = \sum_{j=0}^{k-1} \sum_{h=0}^{k-1} |\alpha_j| \, |\alpha_h| \, .$$

Also we have

$$
\begin{aligned}
\sum_{j=0}^{k-1} \sum_{h=0}^{k-1} (|\alpha_j| - |\alpha_h|)^2 &= \sum_{j=0}^{k-1} \sum_{h=0}^{k-1} (|\alpha_j|^2 - 2 |\alpha_j| |\alpha_h| + |\alpha_h|^2) \\
&= 2k \sum_{j=0}^{k-1} |\alpha_j|^2 - 2 \sum_{j=0}^{k-1} \sum_{h=0}^{k-1} |\alpha_j| |\alpha_h|,
\end{aligned}
$$

this implies

$$k \sum_{j=0}^{k-1} |\alpha_j|^2 - \sum_{j=0}^{k-1} \sum_{h=0}^{k-1} |\alpha_j| |\alpha_h| = \frac{1}{2} \sum_{j=0}^{k-1} \sum_{h=0}^{k-1} (|\alpha_j| - |\alpha_h|)^2 \geq 0,$$

hence, $\quad k \sum_{j=0}^{k-1} |\alpha_j|^2 \;\; \geq \; \Big(\sum_{j=0}^{k-1} |\alpha_j| \Big)^2$.

Let $\alpha_j = x_j - y_{ij}$ where $j = 0,1,...,k-1$, then

$$d(x, y_i) = \sqrt{\sum_{j=0}^{k-1} (x_j - y_{ij})^2} \;\; \geq \; \frac{1}{\sqrt{k}} \sum_{j=0}^{k-1} | \, x_j - y_{ij} \, | ,$$

this completes the proof. From lemma 1 we can obtain the following corollary:

Corollary 1:

$$d(x, y_i) \;\; \geq \;\; u \, (| \, x - y_i \, |)^T \, , \qquad (6)$$

where $| \, x - y_i \, |$ represents the vector whose components are absolute differences of two vectors components, $| x_j - y_{ij} |$ (where $j = 0,1,...,k-1$), and $u \, (| x - y_i |)^T$ is its projection on u. This corollary can be generalized to the Lagrangian distortion measure $J(x, y_i)$ as in the following corollary:

Corollary 2:

$$J(x, y_i) = d(x, y_i) + \lambda R(y_i) \;\; \geq \;\; u \, (| \, x - y_i \, |)^T + \lambda R(y_i) \, . \qquad (7)$$

We define another distortion,

$$\tilde{J}(x, y_i) = u \, (| x - y_i |)^T + \lambda R(y_i) \, , \qquad (8)$$

then $J(x, y_i) \;\; \geq \;\; \tilde{J}(x, y_i)$.

Let us assume that the current best codeword is y_b with the minimum distortion $J(x, y_b) = J_{\min}$. If the codeword y_i satisfies the inequality $\tilde{J}(x, y_i) \;\; \geq \; J_{\min}$, then $J(x, y_i) \; \geq J_{\min}$ is always guaranteed. Hence y_i cannot be closer to x than y_b and can be rejected.

By using the projection pyramid data structure described in section 2, we can extend Eq. (8) to a general case, where tighter decision boundaries for eliminating search operations are obtained in a multilevel. Let us assume that the vector dimension $k = 2^m \times 2^m$, and two projection pyramids for x and y_i are $\{X^0, X^1, ..., X^{r-1}, X^r, ..., X^m\}$ and $\{Y_i^0, Y_i^1, ..., Y_i^{r-1}, Y_i^r, ..., Y_i^m\}$, respectively. Then we can redefine Eq. (8) as,

$$
\begin{aligned}
\tilde{J}^m(x, y_i) &= u^m \, (| \, X^m - Y_i^m \, |)^T \; + \; \lambda R(y_i) \\
&= \frac{1}{2^m} \sum_{j=0}^{2^m-1} \sum_{h=0}^{2^m-1} | \, X^m(j,h) - Y_i^m(j,h) | \;\; + \; \lambda R(y_i), \qquad (9)
\end{aligned}
$$

where $u^m = (1,1,...,1)/2^m$ is a $2^m \times 2^m$-dimensional unit vector, and $X^m(j,h)$ and $Y_i^m(j,h)$ represent the values of the (j, h)-th pixels on X^m and Y_i^m, respectively. Thus, on the top level,

$$\tilde{J}^0(x, y_i) = | \, X^0 - Y_i^0 \, | + \lambda R(y_i) \, . \qquad (10)$$

From the last definition in Eq. (9) and corollary 2, we can easily get the following theorem.

Theorem 1:

$$\tilde{J}(x,y_i) \equiv \tilde{J}^m(x,y_i) \geq \tilde{J}^{m-1}(x,y_i) \geq \tilde{J}^{m-2}(x,y_i)$$
$$\geq \ldots \geq \tilde{J}^0(x,y_i)\,. \qquad (11)$$

Proof: Since

$$\tilde{J}^m(x,y_i) = \frac{1}{2^m}\sum_{j=0}^{2^m-1}\sum_{h=0}^{2^m-1} |X^m(j,h) - Y_i^m(j,h)| \ + \ \lambda R(y_i)$$
$$= \frac{1}{2^m}\sum_{j=0}^{2^{m-1}-1}\sum_{h=0}^{2^{m-1}-1} (|X^m(2j,2h) - Y_i^m(2j,2h)| \ +$$
$$|X^m(2j+1,2h) - Y_i^m(2j+1,2h)| \ +$$
$$|X^m(2j,2h+1) - Y_i^m(2j,2h+1)| \ +$$
$$|X^m(2j+1,2h+1) - Y_i^m(2j+1,2h+1)| \ ) + \lambda R(y_i),$$

and from Eq. (3) if $l = r + 1$, we get,

$$\tilde{J}^{m-1}(x,y_i) = \frac{1}{2^{m-1}}\sum_{j=0}^{2^{m-1}-1}\sum_{h=0}^{2^{m-1}-1}\frac{1}{2}|(X^m(2j,2h) - Y_i^m(2j,2h)) +$$
$$(X^m(2j+1,2h) - Y_i^m(2j+1,2h)) +$$
$$(X^m(2j,2h+1) - Y_i^m(2j,2h+1)) +$$
$$(X^m(2j+1,2h+1) - Y_i^m(2j+1,2h+1))\,| + \lambda R(y_i)$$
$$\leq \frac{1}{2^m}\sum_{j=0}^{2^{m-1}-1}\sum_{h=0}^{2^{m-1}-1} (|X^m(2j,2h) - Y_i^m(2j,2h)\,| \ +$$
$$|X^m(2j+1,2h) - Y_i^m(2j+1,2h)| \ +$$
$$|X^m(2j,2h+1) - Y_i^m(2j,2h+1)| \ +$$
$$|X^m(2j+1,2h+1) - Y_i^m(2j+1,2h+1)|\,) + \ \lambda R(y_i),$$

then $\tilde{J}^{m-1}(x,y_i) \leq \tilde{J}^m(x,y_i)$. This completes the proof.

With the above theorem in hand, we begin describing the proposed algorithm. For a training vector x with a current best codeword y_b and a current minimum distortion $J_{\min} = J(x,y_b)$, the algorithm starts from the top level of the projection pyramid for any other codeword y_i. It first calculates $\tilde{J}^0(x,y_i)$, then if $\tilde{J}^0(x,y_i) \geq J_{\min}$, thus codeword y_i will not be the closest one and can be rejected. Otherwise, $\tilde{J}^1(x,y_i)$ on the second level is calculated and checked. If $\tilde{J}^1(x,y_i) \geq J_{\min}$, from the similar reason as above, the codeword y_i can be rejected. If it is not rejected, the third level is tested. This process is repeated until y_i is rejected or the bottom level is reached. If the bottom level is reached, then the distortion $J(x,y_i)$ is calculated and checked. If $J(x,y_i) < J_{\min}$, the current minimum distortion $J_{\min}$ is replaced by $J(x,y_i)$ and the current closest codeword to x is set to be y_i.

It is emphasized that the first test in the top level is equivalent to the first elimination rule of Cardinal method [6] and this explains the reason of why we use the projection pyramid data structure.

3.2. Projection Pyramid With Variance Test Algorithm

Since the variance of a vector is a simple measure to detect whether a vector is homogeneous, we add a variance test to the projection pyramid algorithm at the top level test to reduce the search area. We define the variance values for the vector x and the codeword y_i as,

$$v_x^2 = \sum_{j=0}^{k-1}(x_j - m_x)^2, \ v_{y_i}^2 = \sum_{j=0}^{k-1}(y_{ij} - m_{y_i})^2 \ , \qquad (12)$$

where m_x and m_{y_i} are the mean values of x and y_i.

Lemma 2:

$$d(x,y_i) \geq |v_x - v_{y_i}|\,.$$

The proof of lemma 2 is given in [4]. Also, Lemma 2 can be extended to the Lagrangian distortion measure as in the following corollary:

Corollary 3:

$$d(x,y_i) + \lambda R(y_i) \geq |v_x - v_{y_i}| + \lambda R(y_i),$$

this means $J(x,y_i) \geq J_v(x,y_i)$,where

$$J_v = |v_x - v_{y_i}| + \lambda R(y_i)\,. \qquad (13)$$

Now we turn to describe the proposed algorithm. For a training vector x with a current minimum distortion $J_{\min}$, and any codeword y_i, the algorithm checks the top projection pyramid test. If $\tilde{J}^0(x,y_i) \geq J_{\min}$, y_i will be rejected. Otherwise, it calculates $J_v(x,y_i)$, then if $J_v(x,y_i) \geq J_{\min}$, thus codeword y_i will not be the closest codeword and can be rejected. Otherwise, it continues the projection pyramid procedure from the second level until finding the closest codeword y_i to x.

Actually, using the variance test at the top level is equivalent to the second elimination rule of Cardinal method [6].

4. 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, thereby each block becomes a 16-dimensional vector $(k = 16$ and $m = 2)$. The tested methods are Cardinal (CARD), which is known as the fastest existing method for ECVQ, the projection pyramid (PP) and the projection pyramid with variance test (PPV).

Figs. 2 and 3 present the execution time (in seconds) of the three methods with various codebook sizes at $\lambda = 0.5$ for Lena and Baboon, respectively. The timings were made on Pentium III (866 MHZ). We can see that both PP and PPV methods significantly accelerate the codebook design more than the CARD method. In case of Lena, PP and PPV reduce the time by average ratios 37.1% and 40.6%, respectively. But in case of Baboon, they reduce the execution time by average ratios 39.4% and 42.9%, respectively. It can be seen that as the codebook size increases, the efficiency of the proposed algorithms becomes better than the CARD method. This is an important merit of our algorithms, because design of a larger codebook size requires more intensive computation.

Figs. 4 and 5 compare the rejection ratio of the codewords in each iteration compared to the FS method at $N = 256$ and $\lambda = 0.5$ for both Lena and Baboon, respectively. The average rejection ratios are 95.3% for CARD, 96.5% for PP, and 96.9% for PPV in case of Lena, while they are 79.9%, 86.5% and 88.2% for baboon, respectively. We can see that the rejection ratio at iteration 1 is smaller than the ratios at other iterations. The main reason is that the tested methods employ a random best codeword for each training vector x at the first iteration, while in any other iterations, they use the best codeword obtained for x in the previous iteration.

Improvement of rejection ratio in the intermediate level for Baboon image is larger than that of Lena image. This difference can be explained from the statistics of the used test images in the following. For a given input vector x, the first elimination rule of the CARD method, which is also employed in the top level test of both the PP and the PPV methods, constrains the search area between two parallel hyperplanes normal to the central line[6]. The width of this search area is determined by the Lagrangian

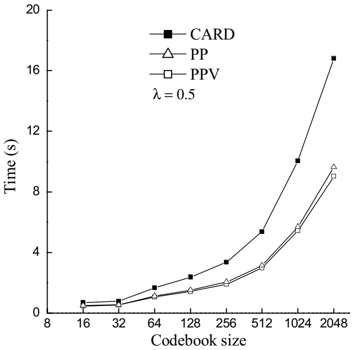

Fig. 2 Comparison of execution time at λ = 0.5 for Lena.

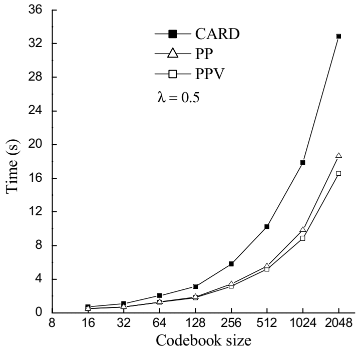

Fig. 3 Comparison of execution time at λ = 0.5 for Baboon.

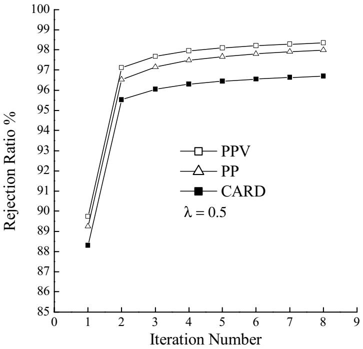

Fig. 4 Comparison of rejection ratio in each iteration at codebook size 256 and λ = 0.5 for Lena.

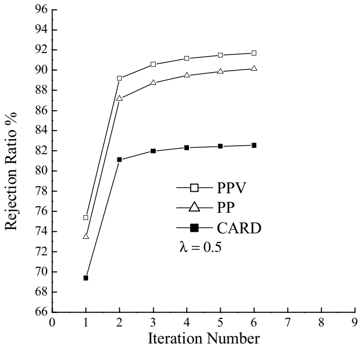

Fig. 5 Comparison of rejection ratio in each iteration at codebook size 256 and λ = 0.5 for Baboon.

distortion $J(x, y_b)$, where y_b is the current best codeword. This search area is affected by the distribution of codewords, which is also related to that of input vectors. For instance, when the codeword distribution is dense like in Lena image, $J(x, y_b)$ will be small and the width of the search area becomes narrow. Then, most of codewords will be rejected at the top level test, and there is less chance to reject more codewords in the other levels as in Fig. 4. While in Baboon image, the codeword distribution is not so dense as in Lena image, then $J(x, y_b)$ may be larger and more codewords are contained in the search area at the top level. Then, there is a chance to reject more codewords in the other levels of the proposed methods as in Fig. 5. As a result, the proposed projection pyramid algorithm is more effective for difficult images of which coding efficiency is low.

5. CONCLUSIONS

In this paper, we have presented two fast search algorithms for ECVQ. The first algorithm uses the projection pyramids of codewords to reject many unmatched codewords, thus drastically speeding up the search process in ECVQ codebook design. In the second algorithm, the variance test was added to the projection pyramid method to reject more codewords in the search process. Simulation results show that the complexity of the proposed algorithms is significantly reduced while the coding quality remains the same as of the exhaustive search algorithm.

6. REFERENCES

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

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

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

[4] 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.

[5] 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.

[6] 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.

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

[8] K. Imamura, A. Swilem, and H. Hashimoto, "Fast codeword search algorithm for ECVQ using hyperplane decision rule," Proc. ISCAS2003, vol. 2 , pp. 476-479, May 2003.

[9] C.-H. Lee and L.-H. Chen, "A fast search algorithm for vector quantization using mean pyramids of codewords," IEEE Trans. Commun., vol. 43, pp. 1697-1702, Feb. 1995.

[10] J. S. Pan, Z. M. Lu, and S. H. Sun, "Fast codeword search algorithm for image coding based on mean-variance pyramids of codewords," Electron. Lett., vol. 36, no. 3, pp. 210-211, Feb. 2000.

[11] B. C. Song and J. B. Ra, "A fast search algorithm for vector quantization using L_2-norm pyramid of codewords," IEEE Trans. Image Process., vol. 11, no. 1, pp. 10-15, Jan. 2002.

[12] P. J. Burt and E. Adelson, "The Laplacian pyramid as a compact image code," IEEE Trans. Commun., vol. COM-31, pp. 532-540, Apr. 1983.