# In Search of the Optimal Searching Sequence for VQ Encoding

Yuk-Hee Chan and Wan-chi Siu

***Abstract*—The codeword searching sequence is sometimes very vital to the efficiency of a vector quantization (VQ) encoding algorithm. In this paper, we evaluate some necessary criteria for the derivation of an optimal searching sequence and derive the optimal searching sequence based on such criteria.**

## I. Introduction

Two common strategies have been used to reduce the complexity inherent in a vector quantization (VQ) encoding algorithms. One resorts to simpler but suboptimal variants and sacrifices quality such as the tree searched VQ [1]. The other remains with the original VQ and devises fast algorithms such as the partial distance search (PDS). This second category of algorithms is more flexible since they are codebook-independent [2]–[8], but the searching sequence of the codewords is very vital to the efficiency of the algorithms.

Consider the case that one has to represent a given D-dimensional input vector $\vec{x} = (x_1, x_2, \cdots, x_D)$ with a particular codeword selected from a codebook containing $N$ codewords, namely, $\mathcal{C} = \{\vec{y}_i \mid i = 1, 2, \cdots, N\}$. The selection is based on the minimum Euclidean distance criterion, where the Euclidean distance measure is defined as $d = \sum_{j=1}^{D}(y_{i,j} - x_j)^2$. To effect an efficient encoding algorithm, one can define an appropriate vector-to-scalar mapping and sort the codewords in the codebook according to their mapping values. Then, for any input vector $\vec{x}$, one can first evaluate its mapping value and start the search for the minimum distortion codeword with the codeword having the closest mapping value and proceed to the next nearer codewords. Obviously, if the mapping used is efficient, it is most likely that one can arrive at the minimum after only a few steps, which can save a lot of computational effort since those codewords without being searched could be disqualified easily through various tests [2]–[8] and be rejected.

In this paper, we evaluate some necessary conditions for the derivation of an optimal searching sequence for VQ encoding and derive the optimal searching sequence based on the minimum distortion criterion.

## II. Algorithm

Ideally, an optimal mapping $f$ should satisfy the following criterion

$$\|\vec{v} - \vec{v}_o\|^2 \propto |f(\vec{v}) - f(\vec{v}_o)| \tag{1}$$

where $\vec{v}_o$ is the input vector and $\vec{v}$ is a particular codeword. However, this cannot be achieved if the mapping $f$ is not bijective. We proposed a simple algorithm to dynamically define a searching sequence for encoding a particular input vector [9]. In this approach, more than one mapping is defined to determine the searching sequence. The concept can be generalized here. In fact, the basic idea is to confine $\vec{v}_d$, the closest codeword to vector $\vec{v}_o$, into a smaller set by using the fact that

$$\vec{v}_d \in \Re_k \subseteq \Re_{k-1} \subseteq \cdots \subseteq \Re_1, \quad \text{if} \quad \vec{v}_d \in \Re_k \tag{2}$$

where $f_i(\vec{v})$ is any suitable mapping and

$$\begin{aligned}\Re_k \equiv \{\vec{v} : |f_1(\vec{v}) - f_1(\vec{v}_o)| < \epsilon_1\} \cap \\ \{\vec{v} : |f_2(\vec{v}) - f_2(\vec{v}_o)| < \epsilon_2\} \cap \cdots \cap \\ \{\vec{v} : |f_k(\vec{v}) - f_k(\vec{v}_o)| < \epsilon_k\}.\end{aligned} \tag{3}$$

Here, $\epsilon_i$ are some predefined values. In that case, one can search less codeword to get $\vec{v}_d$. The derivation in [9] is not optimal in terms of the convergence of the searching sequence since the complexity is most concerned during the derivation of the algorithm.

Assume that the mappings we are looking for are in the form of $f(\vec{v}) = \sum_{i=1}^{D} s_i v_i^2$, where $s_i$'s are scalar coefficients. Then, in order to minimize the size of the set $\Re_k$, the following two criteria should be satisfied: (i) mappings are uncorrelated with each other and (ii) the deviation of mapping values between two different mappings is easily measurable. In formulation, we have

a) $\mathbf{E}(f_i(\vec{v})f_j(\vec{v})) = 0$ if $f_i \neq f_j$
b) $\mathbf{E}\left[(f_i(\vec{v}) - f_j(\vec{v}))^2\right]$ is maximized

where $\mathbf{E}(\cdot)$ is the expectation operator. If criterion (a) is satisfied, criterion (b) can be further simplified and satisfied by maximizing both $\mathbf{E}[(f_i(\vec{v}))^2]$ and $\mathbf{E}[(f_j(\vec{v}))^2]$.

Without losing the generality, we are going to define $D$ mappings here. Then, we have $\mathbf{m} = \mathbf{F}\mathbf{v}$, or, in matrix form,

$$\begin{pmatrix} m_1 \\ m_2 \\ \cdot \\ m_D \end{pmatrix} = \begin{pmatrix} f_{11} & f_{12} & \cdots & f_{1D} \\ f_{21} & f_{22} & \cdots & f_{2D} \\ \cdots & \cdots & \cdots & \cdots \\ f_{D1} & f_{D2} & \cdots & f_{DD} \end{pmatrix} \begin{pmatrix} v_1^2 \\ v_2^2 \\ \cdot \\ v_D^2 \end{pmatrix} \tag{4}$$

where $m_i$ and $f_{ij}$ are the output and the $j$th mapping coefficient of mapping $f_i$ respectively. To satisfy the two criteria, the covariance matrix of the mapping results should be a diagonal matrix in the form of

$$\mathbf{A_o} = \begin{pmatrix} a_1 & 0 & \cdots & 0 \\ 0 & a_2 & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & \cdots & a_D \end{pmatrix}, \quad \text{where } a_1 \geq a_2 \geq \cdots \geq a_D.$$

Paper approved by S. Dimolitsas, the Editor for Speech Processing of the IEEE Communications Society. Manuscript received August 8, 1994; revised March 5, 1995.

The authors are with the Department of Electronic Engineering, Hong Kong Polytechnic University, Hung Hom, Hong Kong.

IEEE Log Number 9414786.

As we have $\mathbf{A_o} = \mathbf{E}[\mathbf{mm^T}] = \mathbf{E}[\mathbf{Fvv^TF^T}] = \mathbf{FE}[\mathbf{vv^T}]\mathbf{F^T}$, the mapping matrix $\mathbf{F}$ and the matrix $\mathbf{E}[\mathbf{mm^T}]$ should then respectively be the eigenvector matrix and the eigenvalue matrix of the matrix $\mathbf{E}[\mathbf{vv^T}]$. It is not realistic to get $\mathbf{E}[\mathbf{vv^T}]$ in practical application, but we can approximate it with the codebook on hand. In particular, we have

$$a_{m,n} = \frac{1}{N}\sum_{i=1}^{N}\left(y_{i,m}^2 - \overline{y_m}\right)\left(y_{i,n}^2 - \overline{y_n}\right) \tag{5}$$

where $a_{m,n}$ is the $m$th element of the $n$th column of the matrix $\mathbf{E}[\mathbf{vv^T}]$ and

$$\overline{y_m} = \frac{1}{N}\sum_{i=1}^{N} y_{i,m}^2. \tag{6}$$

Since the matrix $\mathbf{E}[\mathbf{vv^T}]$ is a real and symmetric matrix, the eigenvectors are real and orthonormal. The definition of the desirable mappings is then very simple. Suppose we want to use $k$ mappings to define the set $\Re_k$. The $k$ eigenvectors associated with the largest eigenvalues should be selected to be the corresponding mapping coefficient vectors.

Prior to encoding, codewords are sorted with their mapping results for each mapping. During encoding, the corresponding mapping results of the input vector $\vec{x}$ are evaluated to define the set $\Re_k$ with $\epsilon_1 = \epsilon_2 = \cdots = \epsilon_k = \epsilon$.

The proposed searching sequence $\{S(i) : i = 0, 1, \cdots, N-1\}$ for a given input vector is then dynamically obtained by gradually increasing the tolerance $\epsilon$, which can be described with a typical C language format as follows:

$i = 0;$
$\epsilon = \epsilon_{\text{initial}};$
$\Re_{\text{old}} = \Phi;$
while $\{$number of elements in $\Re_{\text{old}} \neq N\}$
  $\{$
  $\epsilon += \Delta\epsilon;$
  Get $\Re_k$;
  For $(\vec{v} \in \Re_k \setminus \Re_{\text{old}})\ S(i{+}{+}) = \vec{v};$
  $\Re_{\text{old}} = \Re_k;$
  $\}$

For each increment of the tolerance $\epsilon$, there may be more than one element in the set $\Re_k \setminus \Re_{\text{old}}$. In that case, the relative order of these elements can be randomly assigned.

## III. Simulations

The performance of the proposed technique has been tested via simulation experiments on a set of 256 level gray-scale digital images of size 256 × 256 pixels. Four standard images "House", "Girl", "Couple" and "Germany" have been used as a training set to obtain codebooks with vector size $D = 16$ (blocks of 4 × 4 pixels). The performance is evaluated in terms of the number of codewords required to achieve a particular $PPSNR$ while encoding a number of test images, where $PPSNR$ is defined as

$$PPSNR = 10\log\left(\frac{256 \times 256}{\text{distortion of the encoded image}}\right). \tag{7}$$

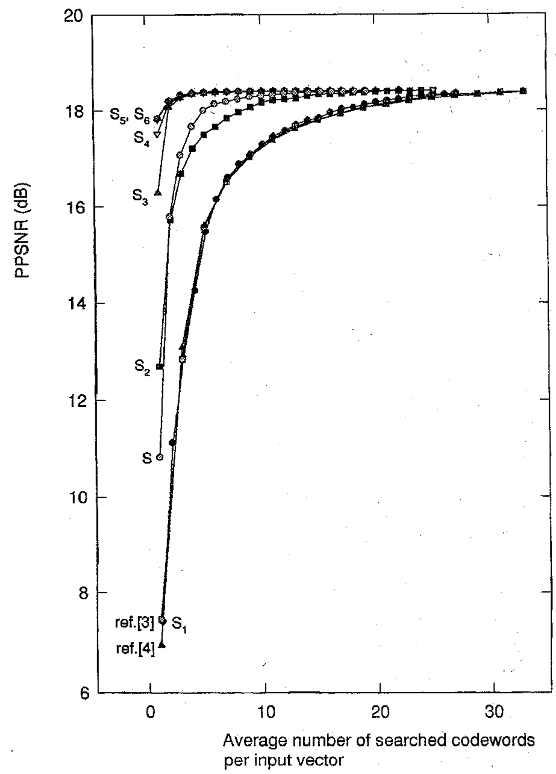


Fig. 1. Comparison of the convergence performance of various searching sequences. $S_n$ stands for the sequence generated with $n$ mappings using the proposed approach. ($PPSNR$ = 18.4158 dB when EFS is applied).

Some other searching sequences defined in [3], [4] are also evaluated for comparison. During the evaluation, we gradually increase the searching range for each input vector according to the searching sequence defined. These experiments are performed to evaluate the convergence performance of different searching sequences.

Fig. 1 shows the case when a codebook of size 256 is used. The result is obtained by encoding the test image "Tunnel" not belonging to the training set. Here, sequence $S_n$ denotes the searching sequence generated with $n$ mappings using the proposed approach. From this figure, one can see that the performance of the proposed searching sequence converges much faster on the optimal performance compared with those of other searching sequences [3], [4]. Typically, the performance of sequence $S_n$ is better than that of $S_m$, where $n > m$. Note that, when the searching sequence $S_n$, where $n > 2$, is applied, only about 2 out of 256 codewords are necessary to be searched to achieve an encoding performance comparable to that of the Exhaustive Full Search scheme (EFS). Sequence $S$ is actually a modified version of the sequence proposed in [9] and is shown here for reference purpose. (Note the difference between their sequence-generation algorithms.) In terms of the convergence performance, sequence $S_3$ is much better than sequence $S$. However, though sequence $S$ is also generated with 3 mappings, its generation overheads is much less than that of sequence $S_3$ or even that of sequence $S_2$. Hence, for real time processing, sometimes it is more desirable to exploit this sequence instead. Table I shows the

TABLE I
COMPUTATIONAL OVERHEADS REQUIRED FOR LOCATING THE STARTING ELEMENTS OF DIFFERENT SEARCHING SEQUENCES FOR A GIVEN INPUT VECTOR

| | Number of Multiplications | Number of Additions | Number of Comparisons |
|---|---|---|---|
| Sequence S | D+1 | 2D-1 | $3\lceil \log_2 N \rceil$ |
| Sequence $S_k$ | 2kD | k(D-1) | $k\lceil \log_2 N \rceil$ |
| Sequence in Ref.[3] | D | D-1 | $\lceil \log_2 N \rceil$ |
| Sequence in Ref.[4] | D-1 | 0 | $\lceil \log_2 N \rceil$ |

computational overheads required for locating the starting elements of different searching sequences for a given input vector. Similar results can be obtained in using different codebooks of various sizes to encode various test images.

## IV. CONCLUSION

In this paper, an algorithm is proposed to adaptively determine the codeword searching sequence for any given input vector. Some necessary criteria for the derivation of an optimal searching sequence are investigated and the optimal searching sequence based on these criteria is also given and evaluated. These searching sequences would by no means be optimal in minimizing their generation overheads but could obtain the desirable codeword of the input vector very effectively. In such a case, they can be used together with any other category 2 algorithms [2]–[8] to improve the encoding efficiency in vector quantization.

## REFERENCES


[1] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression.* Norwell, MA: Kluwer, 1990.

[2] K. K. Paliwal and V. Ramasubramanian, "Effect of ordering the codebook on the efficiency of the partial distance search algorithm for vector quantization," *IEEE Trans. Commun.*, vol. 37, no. 5, pp. 538–540, May 1989.

[3] A. P. Wilton and G. F. Carpenter, "Fast search methods for vector lookup in vector quantization," *Electron. Lett.*, vol. 28, no. 25, pp. 2311–2312, Dec 1992.

[4] R. L. Joshi and P. G. Poonacha, "A new MMSE encoding algorithm for vector quantization," in *Proc. IEEE ICASSP '91*, pp. 645–648.

[5] C. D. Bei and R. M. Gary, "An improvement of the minimum distortion encoding algorithm for vector quantization," *IEEE Trans. Commun.*, vol. COM-33, pp. 1132–1133, 1985.

[6] M. R. Soleymani and S. D. Morgera, "A fast MMSE encoding technique for vector quantization," *IEEE Trans. Commun.*, vol. 37, no. 6, pp. 656–659, June 1989.

[7] S. W. Ra and J. K. Kim, "Fast weight-ordered search algorithm for image vector quantization," *Electron. Lett.*, vol. 27, no. 22, pp. 2081–2083, Oct. 1991.

[8] G. Poggi, "Fast algorithm for full-search VQ encoding," *Electron. Lett.*, vol. 29, no. 12, pp. 1141–1142, June 1993.

[9] Y. H. Chan and W. C. Siu, "The determination of the searching sequence for VQ encoding," in *Proc. IEEE ISCAS'94*, pp. 3.129–3.132.