PAPER

A Novel Variable-Rate Classified Vector Quantizer Design Algorithm for Image Coding

Wen-Jyi HWANG†, Yue-Shen TU†, *and* Yeong-Cherng LU†, *Nonmembers*

SUMMARY This paper presents a novel classified vector quantizer (CVQ) design algorithm which can control the rate and storage size for applications of image coding. In the algorithm, the classification of image blocks is based on the edge orientation of each block in the wavelet domain. The algorithm allocates the rate and storage size available to each class of the CVQ optimally so that the average distortion is minimized. To reduce the arithmetic complexity of the CVQ, we employ a partial distance codeword search algorithm in the wavelet domain. Simulation results show that the CVQ enjoys low average distortion, low encoding complexity, high visual perception quality, and is well-suited for very low bit rate image coding.

key words: *vector quantization, fast codeword search, wavelet transform*

1. Introduction

Vector quantizers (VQs) [1] have shown to be effective for image coding for their excellent rate-distortion performance over traditional scalar quantization schemes. However, the visual perception of a VQ can be poor due to the possible edge degradation of the VQ. This is because the image blocks containing edges usually constitute a small fraction of the image; hence, the training set that used for codebook design is populated with a small fraction of edge vectors. As a result, edges are in general poorly coded, appearing jagged in the reconstructed images.

To improve quality of the visual perception of the reconstructed images and to reduce the degree of edge degradation, a number of classified VQ (CVQ) techniques have been studied [1]–[4]. The CVQ techniques classify the input image blocks into different classes, and design one VQ for each class independently so that the edge information can be properly preserved. However, many CVQs have a common drawback that each class of these CVQs are constructed using the fixed-rate full-search VQs; hence, the constraints of average rate and the storage size can not be specified/controlled independently, and the computational complexity for the encoding of the CVQs can be high.

In this paper, we present a novel CVQ design algorithm for the image coding which enjoys the advantages of low average distortion, high visual perception quality, and low computational complexity. Moreover, the average rate and the storage size required by the algorithm can be prespecified and controlled during the design process. In the algorithm, we first classify the input image blocks in accordance with their edge orientation. There are four classes of edge orientation considered in the CVQ: DC (i.e. no edge is presented in the block), horizontally oriented, vertically oriented, and diagonally oriented.

We construct a VQ for each class to reduce the possible edge degradation of the reconstructed images. The codewords of the VQs are stored in the wavelet domain. This is because the energy of codewords is concentrated on few elements in the wavelet domain. Given a storage size quota for each of the VQs, by truncating the insignificant elements having little energy, more number of codewords can be constructed without exceeding the storage constraints. Therefore, the average distortion for the VQs can be reduced. The number of insignificant elements of codewords are dependent on their edge orientations. Hence, the dimension of the VQs for different classes are different. To control the average rate and storage size of the CVQ, we allow these quantities to be specified before the design. Then, we optimally allocate the rate and storage size to the VQ for each class to minimize the average distortion of the CVQ using the dynamic programming technique. The VQ of each class is then constructed subject to the constraints of the allocated rate and storage size.

To reduce the arithmetic complexity of the CVQ, we utilize a partial distance search (PDS) [5] search technique which performs the fast codeword search for the variable-rate VQ encoding in the wavelet domain. The fast search algorithm is able to reduce the computation time for the encoding without sacrificing the storage size and the performance of the CVQ. Simulation results show that the CVQ algorithm is well-suited for designing a very low bit rate and high dimension VQ with low average distortion, high visual perception quality and fast encoding time.

2. The CVQ Design Algorithm

Figure 1 shows the basic structure of our CVQ. During the encoding process of the CVQ, a source vector **x** is first classified in accordance with its edge orientation

Manuscript received November 7, 1997.
Manuscript revised January 13, 1998.
†The authors are with the Department of Electrical Engineering, Chung Yuan Christian University, Chungli, 32023, Taiwan, R.O.C.

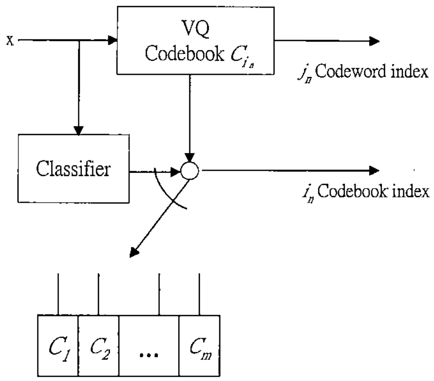


Fig. 1 The general structure of a classified VQ.

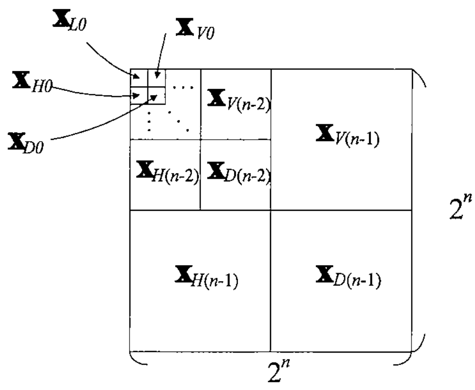


Fig. 2 The result of a n-stage DWT of a $2^n \times 2^n$ vector $\mathbf{x}$.

into one of the four classes: DC, horizontally oriented, vertically oriented and diagonally oriented. Then $\mathbf{x}$ is quantized by the VQ of the selected class. To ensure proper reproduction at the decoder output, both the class and codeword indices for $\mathbf{x}$ should be transmitted to the decoder. Note that only one VQ is constructed for each class. The VQ for different classes might not have the same vector dimension, storage size and average rate.

The design of the CVQ involves the design of edge-oriented classifier, the construction of the VQ for each class subject to the rate and storage size constraints, and the fast encoding technique for reducing the arithmetic complexity for the variable-rate CVQ. In the following subsections, we first describe the classifier. Then we present the optimal rate and storage allocation for the design of the CVQ, and the construction of the VQ for each class after the allocation. Finally, we discuss a fast codeword search technique which performs PDS in the wavelet domain to reduce the encoding complexity.

2.1 An Edge-Oriented Classifier for the CVQ

Since the edge classification is performed in the wavelet domain, in this section, we first briefly introduces some basic facts of the wavelet transform. Let $\mathbf{X}$ be the n-stage discrete wavelet transform (DWT) of a $2^n \times 2^n$ image block $\mathbf{x}$. Then, as shown in Fig. 2, $\mathbf{X}$ is also a $2^n \times 2^n$ block containing subblocks $\mathbf{x}_{L0}$ and $\mathbf{x}_{Vk}, \mathbf{x}_{Hk}, \mathbf{x}_{Dk}, k = 0, \ldots, n-1$, each with dimension $2^k \times 2^k$. Note that, in the DWT, the subblocks $\mathbf{x}_{Lk}$ (lowpass subblocks), and $\mathbf{x}_{Vk}, \mathbf{x}_{Hk}, \mathbf{x}_{Dk}$ (vertical, horizontal and diagonal orientation selective subblocks), $k = 0, \ldots, n-1$, are obtained recursively from $\mathbf{x}_{L(k+1)}$ with $\mathbf{x}_{Ln} = \mathbf{x}$. The decomposition of $\mathbf{x}_{L(k+1)}$ into four subblocks $\mathbf{x}_{Lk}, \mathbf{x}_{Vk}, \mathbf{x}_{Hk}, \mathbf{x}_{Dk}$ can be carried out using a simple quadrature mirror filter (QMF) scheme as shown in [6].

To perform the classification for a given input image block $\mathbf{x}$, we first note that the subblocks $\mathbf{x}_{V0}$ and $\mathbf{x}_{H0}$ in the DWT of $\mathbf{x}$ are actually scalars. The magnitude of $|\mathbf{x}_{H0}|$ and $|\mathbf{x}_{V0}|$ can effectively reveals the edge orientation of $\mathbf{x}$. Based on these magnitudes, as shown in Fig. 3, we divide the $\mathbf{x}_{H0} - \mathbf{x}_{V0}$ plane into nine regions. The image blocks with $(\mathbf{x}_{H0}, \mathbf{x}_{V0})$ located in the region labelled "L," "H," "V," and "D" are the DC, horizontally oriented, vertically oriented and diagonally oriented blocks, respectively. For simplicity, we number the DC, horizontally oriented, vertically oriented, and diagonally oriented classes as class 1, class 2, class 3 and class 4, respectively. Figure 4 shows the location of $(\mathbf{x}_{H0}, \mathbf{x}_{V0})$ of image blocks of twelve 512×512 training images. The dimension of the image blocks is 8×8. From Fig. 4, it is observed that these locations are symmetric. Therefore, the regions shown in Fig. 3 can be symmetrically divided so that the corners of these regions $(a_1, b_1), (a_1, b_2), (a_2, b_1), (a_2, b_2)$ satisfy $a_2 = -a_1, b_2 = -b_1$.

2.2 Design of the VQ in Each Class of the CVQ

The VQ design involves the truncation of coefficients of codewords in the wavelet domain to reduce the codeword dimension, and the optimal storage and rate allocation for the VQ in each class to minimize the average distortion of the CVQ. The detail discussion of the design is given below:

2.2.1 Coefficient Truncation of Codewords in the Wavelet Domain

Let p_i be the probability that an input image block is classified to class i, and let r_i (average bpp) and m_i (number of vectors with dimension $2^n \times 2^n$) be the constraint of the average rate and the storage size for constructing the VQ for class i, respectively. Moreover, let R and M be the overall constraints of average rate and storage size of the CVQ, respectively. Then,

$$\sum_{i=1}^{4} p_i r_i \leq R - R_c, \tag{1}$$

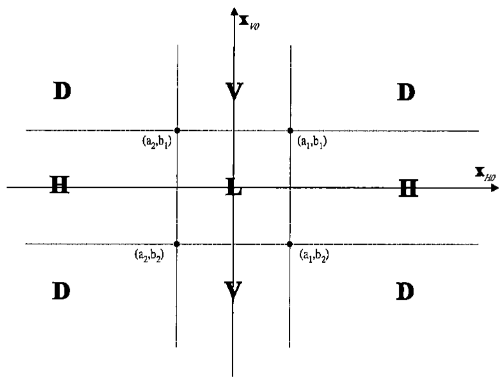


Fig. 3 The partition of $\mathbf{x}_{H0} - \mathbf{x}_{V0}$ plane for the edge classification, where L,H,V, and D denote the regions for DC blocks, horizontally oriented blocks, vertically oriented blocks, and diagonally oriented blocks, respectively.

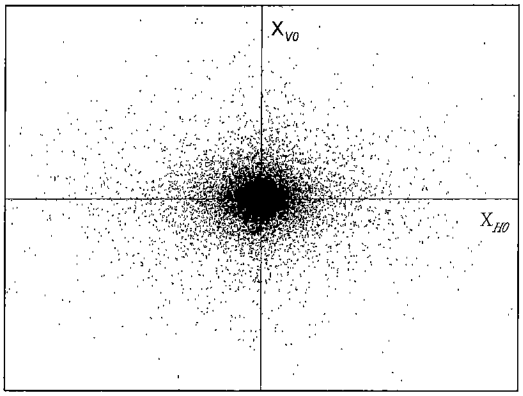


Fig. 4 The location of the wavelet coefficients $\mathbf{x}_{H0}$ and $\mathbf{x}_{V0}$ of 8×8 image blocks in the $\mathbf{x}_{H0} - \mathbf{x}_{V0}$ plane.

$$\sum_{i=1}^{4} m_i \leq M, \tag{2}$$

where R_c is the average rate for transmitting the class indices and $\sum_{i=1}^{4} p_i r_i$ is the average rate for transmitting the codeword indices. Here we assume both the class indices and codeword indices are entropy-encoded. Let Γ_i be the set of the image blocks classified to class i. Now, for Γ_i given, let $d_i(r_i, m_i)$ be the minimum distortion attainable by any VQ with constraints r_i and m_i. In other words,

$$d_i(r_i, m_i) \leq d_i(r_i, m_i, Q) \text{ for all } Q \tag{3}$$

where Q represents a VQ, and $d_i(r_i, m_i, Q)$ is the distortion of that VQ, over the set Γ_i, with constraints r_i and m_i. Then, given the allocation $\{(r_i, m_i), i = 1, \ldots, 4\}$ and the classifier, the minimum average distortion of the CVQ is $D = \sum_{i=1}^{4} p_i d_i(r_i, m_i)$. Hence, the optimal allocation $\{(r_i^*, m_i^*), i = 1, \cdots, 4\}$ minimizing D is given by

$$\{(r_i^*, m_i^*), i = 1, \cdots, 4\} =$$

$$\mathop{\arg\min}_{\{(r_i,m_i),i=1,\cdots,4\}\in T} \sum_{i=1}^{4} p_i d_i(r_i, m_i) , \tag{4}$$

where

$$T = \Bigg\{ \{(r_i, m_i), i = 1, \cdots, 4\} : \sum_{i=1}^{4} m_i \leq M, \ \sum_{i=1}^{4} p_i r_i + R_c \leq R \Bigg\}. \tag{5}$$

To proceed with the optimization, we need to know $d_i(r_i, m_i)$, the lowest distortion attainable by any VQ for class i with rate r_i and storage size m_i, that is, the distortion of the optimum VQ for class i, denoted by Q_i^*. We also need Q_i^* to complete the design of the overall CVQ. However, no known method exists for finding Q_i^* and the corresponding distortion $d_i(r_i, m_i)$. In [7], the entropy-constrained vector quantizer (ECVQ) [8] design algorithm is used to approximate Q_i^* and $d_i(r_i, m_i)$. The ECVQ algorithm is simple to implement and performs well subject to the average rate and alphabet size constraints. However, the algorithm might not be able to fully utilize the storage size available to minimize the average distortion of the VQ. Therefore, here we present a novel technique which modify the ECVQ algorithm for the approximation of Q_i^*.

In the technique, the ECVQ algorithm is still used for the VQ design. After the design of VQ, all the codewords are then saved in the wavelet domain. By using the unitary DWT, the square distances between sourcewords and codewords before and after the transform are identical. Therefore saving the codewords in the wavelet domain will not increase the average distortion of the VQ. In addition, we note that the energy of codewords is concentrated on only few coefficients in the wavelet domain. Hence, by truncating the insignificant coefficients having little energy in the wavelet domain, a significant amount of storage size can be saved with little increase in average distortion. Equivalently, given a constraint of storage size, by removing the insignificant coefficients, more number of codewords can be constructed without exceeding the storage constraint. Therefore, the average distortion for the VQ can be reduced. To illustrate these facts in more detail, we first start with the approximation of $d_i(r_i, m_i)$ using the simple ECVQ algorithm.

Given the constraints r_i and m_i, let B be the resulting codebook after ECVQ design. Given an input image block $\mathbf{x}$, let $p_{j/i}$ be the probability that after $\mathbf{x}$ is classified to the class i, the j-th codeword in B, $\mathbf{y}^j$, is selected as the reproduction codeword for $\mathbf{x}$. Define the modified distortion measure between $\mathbf{x}$ and $\mathbf{y}^j$ as

$$\rho(\mathbf{x}, \mathbf{y}^j) = D(\mathbf{x}, \mathbf{y}^j) - s \log p_{j/i}, \tag{6}$$

where s is the Lagrange multiplier obtained when constructing the ECVQ for class i, and $D(\mathbf{u}, \mathbf{v}) = \sum_t (u_t - v_t)^2$ is the squared distance between $\mathbf{u}$ and $\mathbf{v}$. Let $\mathbf{y}^q$

be the reproduction codeword of $\mathbf{x}$. For an optimal encoder for a variable-rate VQ, q should be such that [8]

$$q = \arg\min_{\{j:\mathbf{y}^j \in B\}} \rho(\mathbf{x}, \mathbf{y}^j). \tag{7}$$

The average distortion of the ECVQ therefore is $E\{d(\mathbf{x}, \mathbf{y}^q)\}$, where $\mathbf{y}^q$ satisfies Eq. (7). In [7], $E\{d(\mathbf{x}, \mathbf{y}^q)\}$ is used for the approximation of $d_i(r_i, m_i)$. Let $\mathbf{Y}^j$ be the DWT of $\mathbf{y}^j$. For an unitary DWT, it can be shown that $D(\mathbf{x}, \mathbf{y}^j) = D(\mathbf{X}, \mathbf{Y}^j)$. Hence, $\rho(\mathbf{x}, \mathbf{y}^j) = \rho(\mathbf{X}, \mathbf{Y}^j)$ where $\rho(\mathbf{X}, \mathbf{Y}^j) = D(\mathbf{X}, \mathbf{Y}^j) - s \log p_{j/i}$. That is, storing codewords in the wavelet domain will not increase the average distortion of the ECVQ.

To see how the truncation of insignificant coefficients in the wavelet domain can improve the approximation of $d_i(r_i, m_i)$, we let $\hat{\mathbf{Y}}^j$ be the truncated version of $\mathbf{Y}^j$. Since the insignificant coefficients in the wavelet domain for different classes might also be different, we consider each class separately for the truncation. For a sourceword $\mathbf{x}$ in class 1, the subblocks $\mathbf{x}_{H(n-1)}, \mathbf{x}_{V(n-1)}$ and $\mathbf{x}_{D(n-1)}$ in general contain little energy. Hence, for each of the codewords in class 1, these subblocks can be truncated. The dimension of the codewords after truncation, N_1, therefore is only $N_1 = 2^{n-1} \times 2^{n-1}$. In class 2, little energy is contained in the subblocks $\mathbf{x}_{V(n-1)}$ and $\mathbf{x}_{D(n-1)}$ of a sourceword $\mathbf{x}$. Hence, we can truncate these insignificant subblocks for the codewords in class 2. The dimension of codewords after truncation therefore is $N_2 = 2^n \times 2^{n-1}$. Similarly, in class 3, the subblocks $\mathbf{x}_{H(n-1)}$ and $\mathbf{x}_{D(n-1)}$ of a sourceword $\mathbf{x}$ are insignificant subblocks containing little energy. Hence, these subblocks for the codewords in class 3 can be truncated. The dimension of codewords after truncation, N_3, is $2^{n-1} \times 2^n$. Finally, in class 4, the energy is distributed more uniformly in the wavelet domain than other classes. Hence, we do not truncate any subblock in the wavelet domain, and $N_4 = 2^n \times 2^n$.

Let $\mathbf{x}$ be the input image block classified to the class i. To encode $\mathbf{x}$ using the truncated codewords, we first note that all the input image blocks for the encoding have the same dimension $2^n \times 2^n$. Hence, after the classification, the image blocks belonging to each class have to be truncated in the wavelet domain in the same manner as the codewords for that class so that both truncated codewords and image blocks in the same class have the same dimension. In addition, although all the image blocks for encoding are squares with dimension $2^n \times 2^n$, the size of images can be rectangular with dimension $2^{k_1} \times 2^{k_2}$, where k_1 and k_2 are not necessarily equal, but are multiples of n.

Let $\mathbf{X}$ be the DWT of $\mathbf{x}$ and $\hat{\mathbf{X}}$ be the truncated version of $\mathbf{X}$. Define

$$\rho(\hat{\mathbf{X}}, \hat{\mathbf{Y}}^j) = D(\hat{\mathbf{X}}, \hat{\mathbf{Y}}^j) - s \log p_{j/i}. \tag{8}$$

The encoding of input image blocks given the truncated codewords is based on $\rho(\hat{\mathbf{X}}, \hat{\mathbf{Y}}^j)$. Let $\hat{\mathbf{Y}}^q$ be the selected truncated codeword for representing $\mathbf{x}$, then

$$q = \arg\min_{\{j:\mathbf{y}^j \in B\}} \rho(\hat{\mathbf{X}}, \hat{\mathbf{Y}}^j). \tag{9}$$

Since the VQ only stores the truncated codewords in the wavelet domain, after the decoding, the selected truncated codeword in the spatial domain has to be obtained. Let $\hat{\mathbf{y}}^q$ be the inverse DWT of $\hat{\mathbf{Y}}^q$. Since $\hat{\mathbf{Y}}^q$ is the truncated version of $\mathbf{Y}^q$, if the dimension of $\hat{\mathbf{Y}}^q$ is not $2^n \times 2^n$, we can simply fill the truncated coefficients with zero when taking the inverse transform. The average distortion of the VQ based on the truncated codewords is therefore $E\{d(\mathbf{x}, \hat{\mathbf{y}}^q)\}$, where q satisfies Eq. (9).

The truncated insignificant coefficients contain little energy. Hence, first we have $D(\hat{\mathbf{X}}, \hat{\mathbf{Y}}^j) \approx D(\mathbf{X}, \mathbf{Y}^j)$, and therefore $\rho(\mathbf{x}, \mathbf{y}^j) \approx \rho(\hat{\mathbf{X}}, \hat{\mathbf{Y}}^j)$. Secondly, since $d(\mathbf{y}^j, \hat{\mathbf{y}}^j) = d(\mathbf{Y}^j, \hat{\mathbf{Y}}^j)$, it follows that $d(\mathbf{y}^j, \hat{\mathbf{y}}^j)$ is equal to the total energy of the truncated insignificant coefficients, and therefore is small. Based on these facts, we conclude that the q's obtained from Eqs. (7) and (9) in general are the same, and the difference between $\mathbf{y}^q$ and $\hat{\mathbf{y}}^q$ is small. Therefore, with the same number of codewords, the VQ based on the truncated codewords has approximately the same average distortion with that of the VQ based on codewords without truncation. In addition, since the storage size for saving the truncated codewords is less than the storage size for saving the untruncated codewords, given a storage size constraint, the VQ based on the truncated codewords can have more codewords in the codebook and consequently achieves lower average distortion. Therefore, using the VQ with truncated codewords can have a better approximation of $d_i(r_i, m_i)$.

To approximate $d_1(r_1, m_1)$, since the dimension N_1 is only $2^{n-1} \times 2^{n-1}$, we can first use ECVQ to design a variable-rate VQ subject to the constraint of average r_1 bpp and $4m_1$ codewords with dimension $2^n \times 2^n$. Since the dimension of codewords after the truncation in the wavelet domain is only one quarter of the dimension before the truncation, although we use $4m_1$ codewords for the design of the ECVQ, the resulting storage size after the truncation does not exceed the storage constraint. Therefore, $d_1(r_1, m_1)$ can be approximated by the ECVQ with average rate r_1 and number of truncated codewords $4m_i$ with dimension N_1. In class 2, The dimension of codewords after truncation is $N_2 = 2^n \times 2^{n-1}$. To approximate $d_2(r_2, m_2)$, we can use ECVQ to design a variable-rate VQ subject to the constraint of average r_2 bpp and $2m_2$ codewords with dimension $2^n \times 2^n$. After the design, we then truncate the insignificant subblocks of codewords in the wavelet domain to obtain the approximation of Q_2^* and $d_2(r_2, m_2)$. Similarly, in class 3, the dimension N_3 is $2^{n-1} \times 2^n$. Consequently, the approximation of $d_3(r_3, m_3)$ can be obtained in a similar fashion as that of $d_2(r_2, m_2)$ except that the subblocks to be truncated are different.

Finally, in class 4, $N_4 = 2^n \times 2^n$ and no truncation is necessary. Hence, we use the ECVQ algorithm subject to the rate r_4 and storage size m_4 codewords with dimension $2^n \times 2^n$ for the approximation of $d_4(r_4, m_4)$.

2.2.2 Optimal Rate and Storage Allocation

After the tables containing $d_i(r_i, m_i), i = 1, \ldots, 4$, for a set of discrete r_i and m_i are obtained, we now consider the optimal rate and storage allocation problem given in Eq. (4). Since the rates and storage sizes take only positive values and are stored as discrete values in tables, and the objective function in Eq. (4) is separable, one can iteratively use the dynamic integer programming technique [9] to find the optimal solution. That is, instead of considering both allocation problems simultaneously, one can solve them iteratively, where each iteration consists of two steps. In step one, the rate allocation is fixed to the values determined in the previous iteration, and the optimal storage allocation which minimizes the distortion is found using the dynamic programming technique. In step two, the storage values obtained in the previous step are fixed, and the optimal rate allocation which minimizes the distortion is found using the dynamic programming technique. The iterations are continued until the overall distortion ceases to decrease appreciably, at which point the sets of storage and rate allocation values are considered to be the solution of the optimization problem. This iterative optimization with respect to two sets of variables yields a *partial optimal solution* (*POS*) [10] to the optimization problem.

After the optimal rate and storage allocation is accomplished, for each class i, the VQ associated with that class is constructed using the ECVQ algorithm with rate r_i and storage size m_i (that is, $4m_1$ codewords for class1, $2m_2$ codewords for class 2, $2m_3$ codewords for class 3, and m_4 codewords for class 4). The codewords after design are stored in the wavelet domain with insignificant coefficients truncated. This completes the design of VQ for each class of CVQ.

2.3 Fast Codeword Search Algorithm for the Encoding of CVQ

During the encoding process of the CVQ, after the classification, from Eq. (9), the exhaustive search process is required. Therefore, without the fast codeword search algorithm presented in this subsection, the arithmetic complexity of the CVQ can be high.

The objective of using the fast search algorithm for the variable-rate CVQ is to reduce the computation time for finding the codeword $\mathbf{y}^q$ satisfying Eq. (9) among the codewords of the selected class after the edge classification. The algorithm is an extension of the fast algorithm presented in [11] for fixed-rate VQs which perform the PDS in the wavelet domain. The algorithm enjoys the following advantages. Before the fast search, the DWT of the codewords have been obtained and saved in the codebook. Moreover, since the DWT of the input image block for encoding has been obtained already after the classification, no extra computation overhead is required to perform the fast search for the CVQ. In addition, in the DWT of a codeword, most of the energy of the codeword will be concentrated in low-pass subblocks. Hence, when the PDS is started at the locations which contain more energy in the wavelet domain, the computation time for performing the codeword search can be significantly reduced.

To perform the PDS, as shown in the Fig. 5, we index the elements of vectors $\hat{\mathbf{X}}$ and $\hat{\mathbf{Y}}^j$ in the zig-zag order. Let X_t and Y_t^j be the t-th element of $\hat{\mathbf{X}}$ and $\hat{\mathbf{Y}}^j$, respectively. Moreover, let $D^f(\hat{\mathbf{X}}, \hat{\mathbf{Y}}^j) = \sum_{t=1}^{f}(X_t - Y_t^j)^2$ be the partial distance between $\hat{\mathbf{X}}$ and $\hat{\mathbf{Y}}^j$. Therefore,

$$D(\hat{\mathbf{X}}, \hat{\mathbf{Y}}^j) > D^f(\hat{\mathbf{X}}, \hat{\mathbf{Y}}^j). \tag{10}$$

Suppose now $\mathbf{x}$ is classified into the class i. Let $\rho^f(\hat{\mathbf{X}}, \hat{\mathbf{Y}}^j) = D^f(\hat{\mathbf{X}}, \hat{\mathbf{Y}}^j) - s \log p_{i/j}$. To perform the fast codeword search among the truncated codewords $\hat{\mathbf{Y}}^1, \ldots, \hat{\mathbf{Y}}^{n_i}$, where n_i is the number of codewords for class i after the VQ design, we initialize the *current closest codeword* to be $\hat{\mathbf{Y}}^h$, where $h = \arg\min_j \rho^1(\hat{\mathbf{X}}, \hat{\mathbf{Y}}^j)$, and *the current minimum distortion* ρ_{min} to be $\rho(\hat{\mathbf{X}}, \hat{\mathbf{Y}}^h)$. For each truncated codeword $\hat{\mathbf{Y}}^j$ to be searched, we compute $\rho^1(\hat{\mathbf{X}}, \hat{\mathbf{Y}}^j)$. Suppose $\rho^1(\hat{\mathbf{X}}, \hat{\mathbf{Y}}^j) > \rho_{min}$, it follows from Eqs. (8), (10) that $\rho(\hat{\mathbf{X}}, \hat{\mathbf{Y}}^j) > \rho_{min}$. Hence, $\hat{\mathbf{Y}}^j$ is not the clos-

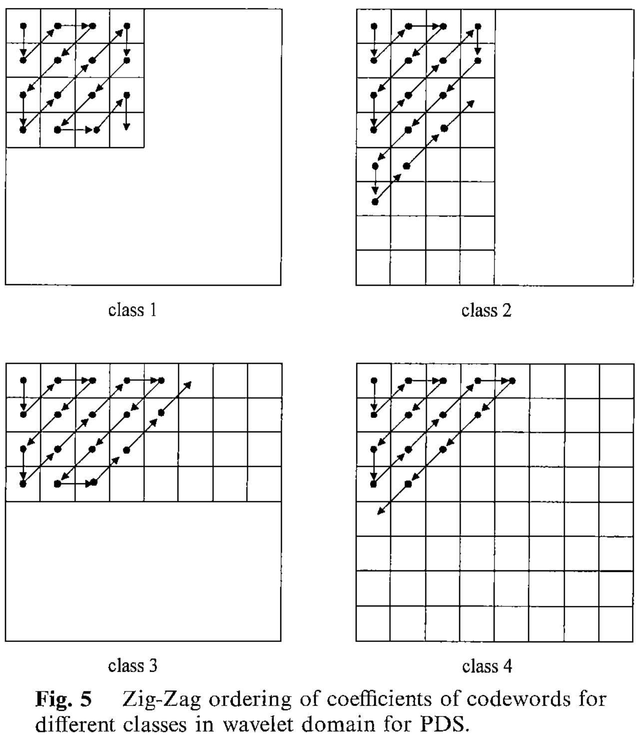


Fig. 5 Zig-Zag ordering of coefficients of codewords for different classes in wavelet domain for PDS.

est codeword to $\hat{\mathbf{X}}$ and can be rejected. Suppose $\rho^1(\hat{\mathbf{X}}, \hat{\mathbf{Y}}^j) < \rho_{min}$, then we perform the following PDS process. Starting from $f = 2$, for each value of f, $f = 2, \ldots, N_i$, we first evaluate $\rho^f(\hat{\mathbf{X}}, \hat{\mathbf{Y}}^j)$. Suppose $\rho^f(\hat{\mathbf{X}}, \hat{\mathbf{Y}}^j) > \rho_{min}$, then from Eqs. (8), (10), it follows that $\rho(\hat{\mathbf{X}}, \hat{\mathbf{Y}}^j) > \rho_{min}$ and $\hat{\mathbf{Y}}^j$ can be rejected. Otherwise, we go to the next value of f and repeat the same process. This PDS process is continued until $\hat{\mathbf{Y}}^j$ is rejected or f reaches N_i. If $f = N_i$, then we compare $\rho(\hat{\mathbf{X}}, \hat{\mathbf{Y}}^j)$ with ρ_{min}. If $\rho(\hat{\mathbf{X}}, \hat{\mathbf{Y}}^j) < \rho_{min}$, then the *current minimum distortion* ρ_{min} is replaced by $\rho(\hat{\mathbf{X}}, \hat{\mathbf{Y}}^j)$ and the *current closest codeword* to $\hat{\mathbf{X}}$ is set to $\hat{\mathbf{Y}}^j$. After all the codewords are searched, the final *current closest codeword* is then the actual closest codeword to $\hat{\mathbf{X}}$, and the ρ_{min} is the corresponding modified distance.

3. Simulation Results

In this section, we present the simulation results of CVQ for image coding. The training data for the design of CVQ are twenty four 512×512 images. The dimension of image blocks is 8×8. The wavelet used in the classification and in the fast search algorithm is the simple Haar wavelet [6] (i.e., $h(n) = \frac{1}{\sqrt{2}}, n = 0, 1$) so that no multiplication is required for the wavelet transform. Hence, the corresponding computation overhead can be small. In the design of classifier, we note that proper selection of (a_1, b_1) can improve the performance of the CVQ. This is because the selection of suitable values of (a_1, b_1) can help to effectively partition $\mathbf{x}_{H0} - \mathbf{x}_{V0}$ plane as shown in Fig. 3 so that image blocks locating in the same region have similar energy distribution in the wavelet domain. Consequently, it is easier for us to find a good codebook for each class and the performance of the CVQ can be improved. Selection of (a_1, b_1) depends on the wavelet used for classification, and in general is accomplished experimentally. From our experiments, for Haar wavelet used in the paper, the parameters $a_1 = b_1 = 60$ performs relatively well than other values for classification and coding. Therefore, we let $a_1 = 60$ and $b_1 = 60$ for the partition of $\mathbf{x}_{H0} - \mathbf{x}_{V0}$ space. The probability for each class is equal to the number of training vectors fall into that class divided by total number of training vectors. Based on the partition, the probability for each class is $p_1 = 0.61$, $p_2 = 0.11$, $p_3 = 0.14$ and $p_4 = 0.14$.

Table 1 shows the result of rate and storage allocation for each class subject to the total average rate constraint of $R = 0.190$ and storage size constraint of $M = 17000$. From the table, it is observed that, although a large portion of image blocks belong to the DC class, the average rate and storage size allocated to the DC class are less than those allocated to the other classes. This is because the DC class contains only low-pass image blocks. To quantize low-pass image blocks with a low average distortion, a small amount of rate and storage size is necessary. However, for the other classes containing high-pass image blocks, more amount of average rate and storage size are required for the VQs to obtain a low average distortion. We also note that this result is quite consistent with the result of bit allocation shown in [2].

Table 1 Rate and storage allocation for each class.

	class 1	class 2	class 3	class 4
Rate r_i	0.142	0.183	0.189	0.202
Storage size m_i	1250	2750	3500	9500

The performance measure of the VQ is the PSNR, which is equal to 255^2 divided by the average distortion (mean squared-error) of the quantized image. Table 2 shows the PSNR of various existing coding techniques and the CVQ subject to the constraints of average rate 0.190, and storage size 17000, respectively. The PSNRs for the algorithms shown in Table 2 are measured on the "Lena" image which is outside the training data. Form the table, we observe that at lower average rates, the CVQ enjoys higher PSNR than other techniques. This is because the CVQ is able to efficiently control and use the storage size available to achieve a higher PSNR; whereas, other techniques might not have full freedom to adjust the storage size of the coding system to reduce the average distortion. The original and reconstructed "Lena" images encoded by our CVQ are shown in Fig. 6. To further show the effectiveness of our algorithm, in addition to the "Lena" image, we also compress two other test images "Pepper" and "F-16" (outside the training data) using the same CVQ in Table 2, and compare the coding results with those of the same test images encoded by JPEG. Table 3 shows the PSNRs for these test images coded by both CVQ and JPEG. From the table, it can be observed that, as com-

Table 2 Performance of various coding techniques for the image "Lena."

Technique	Rate	PSNR
WAVELET VQ [12]	0.21	29.11
EZW [13]	0.25	33.17
DCT-CVQ [2]	0.32	31.41
Trellis Coded DCT [14]	0.25	32.49
Trellis Coded Wavelet [15]	0.27	34.01
SA-W-VQ [16]	0.25	33.97
SPIHT [17]	0.20	33.25
CVQ (storage constraint $M = 17000$)	0.19	34.13

Table 3 Performance of our algorithm and JPEG for various test images. The average rate of the CVQ is 0.19 bpp.

	CVQ ($M = 17000$)	JPEG [18]	
	PSNR	rate	PSNR
Lena	34.13	0.40	33.42
Pepper	34.77	0.41	34.24
F-16	34.57	0.54	33.73

(a) (b)

Fig. 6 The original test image "Lena" and its reconstruction after quantized by our variable-rate CVQ subject to rate constraint 0.19 and storage size 17000. (a) original image. (b) reconstructed image.

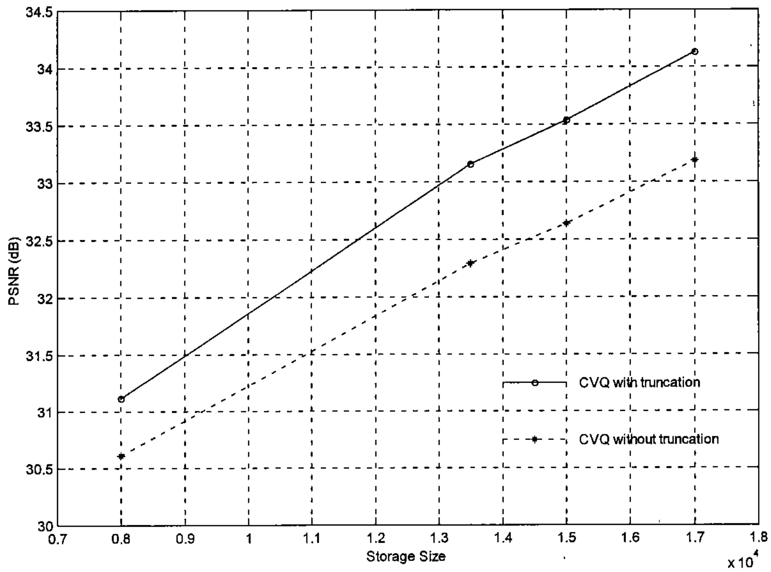


Fig. 7 PSNRs of our CVQ algorithms with and without truncating codeword coefficients in the wavelet domain for various storage size constraints.

pared with JPEG, our algorithm also achieves higher PSNR at lower average rate. In addition, the generality of the CVQ algorithm is demonstrated from the high performance our algorithm has for different test images in Table 3.

Figure 7 shows the PSNR of our CVQ algorithm for various storage size constraints. The average rate constraint is 0.19. The PSNR is again measured on the test image "Lena." From the figure, we observe that the PSNR of the CVQ is improved by increasing the storage size constraint of the CVQ. For the comparison purpose, we also implement the CVQ algorithm without truncating the codeword coefficients in the wavelet domain [19], but with the same classifier, and rate and storage allocation algorithm. To ensure meaningful comparison between the two techniques, since the storage size of the CVQ in this paper is specified in terms of the number of vectors with dimension $2^n \times 2^n$, the storage size of the CVQ without truncating is also specified in the same way. We can find from the figure that, subject to the same average rate and storage size constraints, the CVQ which truncates insignificant coefficients of codewords in the wavelet domain perform better than the CVQ which does not. This is not surprising because, subject to the same storage size constraint, the CVQs truncating insignificant coefficients can include more codewords in the codebook for the encoding.

The arithmetic complexity of a VQ in general is defined as *average number of distance calculations per image block*. Since, without truncating insignificant co-

Table 4 Arithmetic complexities of various codeword search schemes for the CVQ.

Exhaustive Search	6828.17
PDS[5]	301.51
PDS + ordering[20]	199.50
PDS + wavelet	75.42

efficients of codewords in the transform domain, one distance calculation requires $2^n \times 2^n$ multiplications, we therefore define the arithmetic complexity of the CVQ presented in this paper as $\bar{T} = \frac{T_1}{2^n \times 2^n}$, where T_1 is the average number of multiplications per image block (sourceword).

Table 4 shows the arithmetic complexities of the CVQ for encoding the test image "Lena." The storage size constraint is 17000. From the table, it is observed that the fast codeword search using the PDS in the wavelet domain requires only approximately 1.1% of the arithmetic complexity of the traditional exhaustive search schemes (that is, performing the full-search in the original domain). Moreover, our fast search technique also enjoys lower arithmetic complexities as compared with other fast search methods.

Based on these simulation results, we conclude that, by combining the CVQ design technique with the fast PDS in the wavelet domain for the codeword search, we are able to construct a VQ with high compression ratio, high PSNR, high visual quality and low arithmetic complexity.

4. Conclusion

A novel variable-rate CVQ design algorithm is presented for image coding. The algorithm can achieve a higher PSNR at lower bit rate than other existing coding techniques. When performing the PDS in the wavelet domain for the encoding, the algorithm can have very low computational complexity. In addition, the algorithm allows the average rate and storage size of the CVQ to be pre-specified before the design. Simulation results show that the algorithm is well-suited for the very low bit rate image coding where high PSNR, high visual quality, and fast encoding time are required.

References

[1] A. Gersho and R.M. Gray, Vector Quantization and Signal Compression, Kluwer, 1992.

[2] J.W. Kim and S.U. Lee, "A transform domain classified vector quantizer for image coding," IEEE Trans. Circuits and Systems for Video Technology, vol.2, pp.3–14, March 1992.

[3] A. Kubrick and T. Ellis, "Classified vector quantization of images," IEE Proceedings, vol.137, Pt.I, no.6, pp.379–386, Dec. 1990.

[4] B. Ramamurthi and A. Gersho, "Classified vector quantization of images," IEEE Trans. Commun., vol.34, pp.1105–1115, Nov. 1986.

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

[6] M. Vetterli and J. Kovacevic, "Wavelets and Subband Coding," Prentice Hall, New Jersey, 1995.

[7] W.J. Hwang and H. Derin, "Multistage storage- and entropy-constrained tree-structured vector quantization," IEEE Trans. Signal Processing, vol.44, pp.1801–1810, 1996.

[8] P.A. Chou, T. Lookabaugh, and R.M. Gray, "Entropy-constrained vector quantization," IEEE Trans. Acoustics, Speech, and Signal Processing, vol.37, pp.31–42, Jan. 1989.

[9] F.S. Hillier and G.J. Lieberman, "Introduction to Operation Research," 4th Edition, Holden-Day, 1986.

[10] R.E. Wendell and A.P. Hurter, Jr., "Minimization of a non-separable objective function subject to disjoint constraint," Oper. Res., 24, pp.643–657, 1976.

[11] W.J. Hwang, S.S. Jeng, and B.Y. Chen, "Fast codeword search algorithm using wavelet transform and partial distance search techniques," Electronic Letters, pp.365–366, vol.33, Feb. 1997.

[12] M. Antonini, M. Barland, P. Mathieu, and I. Daubechies, "Image coding using wavelet transform," IEEE Trans. Image Processing, vol.1, pp.205–220, April 1992.

[13] J.M. Shapiro, "Embedded image coding using zerotrees of wavelet coefficients," IEEE Trans. Signal Processing, pp.3445–3462, vol.41, Dec. 1993.

[14] M.W. Marcellin, P. Sriram, and K.L. Tong, "Transform coding of monochrome and color images using trellis coded quantization," IEEE Trans. Circuits Systems for Video Technology, vol.3, pp.270–276, Aug. 1993.

[15] P. Sriram and M.W. Marcelin, "Image coding using wavelet transforms and entropy-constrained trellis coded quantization," Proc. '93 ICASSP Conf., pp.V554-V557, April 1993.

[16] E.A.B. da Silva, D.G. Sampson, and M. Ghanbari, "A successive approximation vector quantizer for wavelet transform image coding," IEEE Trans. Image Processing, vol.6, pp.299–310, Feb. 1996.

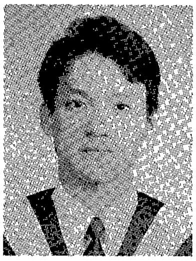
[17] A. Said and W.A. Pearlman, "A new, fast and efficient image codec based on set partitioning in hierarchical trees," IEEE Trans. Circuits and Systems for Video Technology, pp.243–250, June 1996.

[18] W.B. Pennebaker and J.L. Mitchell, "JPEG Still Image Data Compression Standard," Van Nostrand Reinhold, 1992.

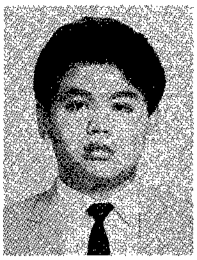
[19] W.J. Hwang, Y.S. Tu, and Y.C. Lu, "Storage- and entropy-constrained classified vector quantization," IEEE Trans. Consumer Electronics, pp.1252–1261, vol.43, Nov. 1997.

[20] 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, pp.538–540, 1989.

Wen-Jyi Hwang received the Dipl. Electronics Eng. from National Taipei Institute of Technology, Taiwan, in 1987, and the M.S. and Ph.D. degrees in Electrical and Computer Engineering from the University of Massachusetts at Amherst, USA, in 1990 and 1993, respectively. Since 1993, he has been with the Department of Electrical Engineering at Chung Yuan Christian University, Taiwan, as an Associate Professor. His research interests include signal compression, image processing, and digital communication systems.



Yue-Shen Tu obtained his B.S. and M.S. degrees from Chung Yuan Christian University in 1995 and 1997, respectively. His research interests include signal compression and image processing.



Yeong-Cherng Lu was born in Taipei, Taiwan, in 1970. He obtained his B.S. degree from Chung Yuan Christian University 1n 1996. Now he is currently working toward M.S. degree in the Department of Electrical Engineering, Chung Yuan Christian University. His research interests include signal compression, image processing, and fast motion estimation