Optical Engineering 44(6), 067001 (June 2005)

New variable-rate finite state vector quantizer for image coding

Shiueng-Bien Yang
Leader University
Department of Computer Science
and Information Engineering
No. 188 Sec. 5 An-Chung Rd.
Tainan City, Taiwan
E-mail: ysb@mail.leader.edu.tw

Abstract. Although side-match vector quantization (SMVQ) and its variants have been proposed to reduce the bit rate of coding images, the coding quality is decreased by it. The reason is that when a block is encoded in SMVQ, the side-match method uses only two neighboring blocks, (no edge information is available in the other two neighboring blocks) to predict the state of the codebook. We propose a new variable-rate finite state vector quantizer (VRFSVQ) to encode the blocks with a variable rate according to both the coding quality and the bit rate. The VRFSVQ can preencode all the blocks, allowing each block to employ more than two neighboring blocks to yield the state codebook. Furthermore, whereas users of SMVQ must specify the size of the state codebook before applying the side-match method to yield the state codebook, in the VRFSVQ the users do not need to specify the size of the state codebook for each encoding block. Experimental results indicate that the VRFSVQ has a high peak SNR and low bit rate for image coding. [DOI: 10.1117/1.1928281]

Subject terms: finite state vector quantizer; image coding; smooth side-match method.

Paper 040182 received Mar. 31, 2004; revised manuscript received Aug. 23, 2004; accepted for publication Dec. 21, 2004; published online Jun. 7, 2005.

1 Introduction

Vector quantization (VQ)[1,2] is useful for data compression and coding, especially in image and speech coding. In VQ, a codebook, which consists of a set of codewords, must be found. Each codeword in the codebook is represented as a vector. When the input data, in the form of a vector, are encoded by VQ, they must be compared with all the codewords in the codebook to find the nearest codeword. Then, the nearest codeword index can be taken as the code of the input data. The input data are then reconstructed by a simple table-lookup technique, using the input data code.

Finite-state vector quantizers (FSVQs)[3–9,15] have been proposed for low-bit-rate image compression. The FSVQ provides an efficient result, since it exploits the correlation between neighboring vectors to reduce each index size by contracting the codebook into a smaller one, called the state codebook. For each input vector, the state codebook is generalized by the next-state function that is used to select the codewords from the supercodebook. Each input vector is encoded in its own state codebook.

The SMVQ was proposed in Ref. 10 and yields a kind of FSVQ. SMVQ uses the side-match method to obtain the state codebook for each input block. Many variants of SMVQ were recently proposed in Refs. 11–18. SMVQ tries to make the gray levels of the pixels across the boundaries of neighboring blocks as similar as possible. However, SMVQ may fail to select the codeword that best fits the current encoded block if the gray-level transition across the boundaries with the neighboring blocks is either increasing or decreasing. In Ref. 15, HTSMVQ, based on a three-sided side-match FSVQ, was proposed to improve the performance of SMVQ. Although many low-bit-rate still-image compression methods have been proposed recently, such as hybrid VQ and SPIHT,[19] which incorporate wavelet or subband techniques, HTSMVQ outperforms SPIHT and JPEG when the bit rate is higher than 0.2 bits per pixel (bpp). In Ref. 16, the smooth side-match vector quantizer (SSMVQ) was proposed and used to replace the side-match method to yield the state codebook, giving superior coding quality for the same bit rate. In Refs. 17 and 18, the smooth side-match method was applied to progressive image coding. However, two problems are common to all variants of SMVQ, as mentioned previously. First, each encoding block uses only two neighboring blocks to calculate the side-match distortions for the codewords in the supercodebook. The example in Fig. 1 reveals that the use of two neighboring blocks does not suffice to encode a block well. In Fig. 1, the central block takes the upper and left blocks as neighboring blocks in SMVQ, and cannot be encoded well because no edge information in the right and lower blocks is available, and the prediction is very poor. In Ref.

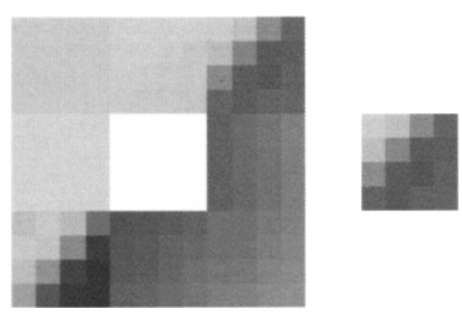

Fig. 1 A block and its neighboring blocks.

0091-3286/2005/$22.00 © 2005 SPIE

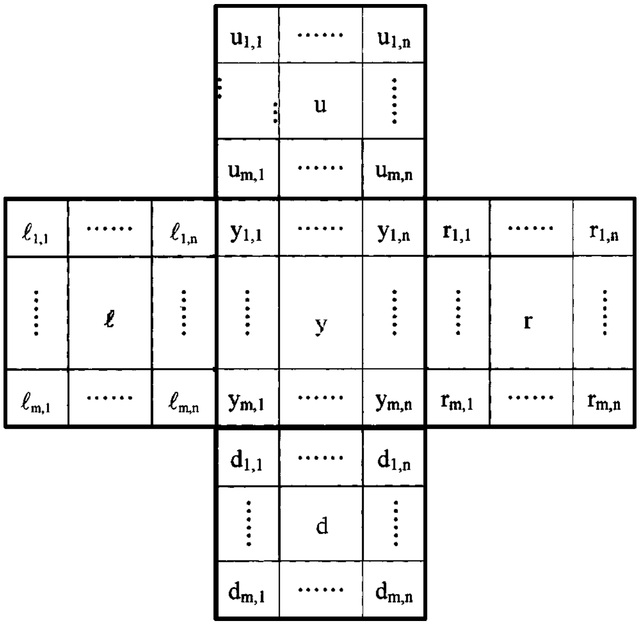

Fig. 2 The codeword *y* and its four neighboring blocks.

15, many blocks are selected as basic blocks when an image is encoded to solve the problem. These basic blocks are first coded using ordinary VQ. However, the bit rate increases when the number of basic blocks is large in SMVQ. In the second problem, the user must specify the state codebook size before encoding the blocks. Although using a large state codebook size to encode blocks can increase the coding quality, the bit rate is also thus increased. The use of a smaller state codebook to encode blocks can reduce the bit rate, but the coding quality of the image decreases.

In this paper, a variable-rate finite-state vector quantizer (VRFSVQ) is proposed to improve the performance of SMVQ and its variants. The VRFSVQ addresses two main design issues. First, it applies the supercodebook directly to the basic blocks, and applies two-stage VQ to the other nonbasic blocks. Unlike in previous studies, each nonbasic block can be encoded by the state codebook, which is obtained by using more than two neighboring blocks to determine the smooth side-match distortions for all the codewords in the supercodebook. Thus, the coding efficiency of the VRFSVQ is improved by using more information about the correlation. Second, users of the VRFSVQ need not specify the state codebook size before encoding the blocks. The state codebook size for each encoding block varies according to the coding quality and the bit rate. Also, high-detail blocks can employ a larger state codebook to enhance coding quality, and low-detail blocks can use a smaller state codebook to reduce the bit rate. Therefore, the coding quality of the VRFSVQ is better than that of traditional SMVQ for a given bit rate.

The rest of this paper is organized as follows. Section 2 shows the smooth side-match method using four neighboring blocks to yield the state codebook. The design of the state codebooks with variable sizes is given in Sec. 3. Section 4 shows the design of the VRFSVQ. Section 5 presents experimental results. Conclusions are given in Sec. 6.

2 The Smooth Side-Match Method Using Four Neighboring Blocks

In Fig. 2, the size of all blocks is $m \times n$. Let x be the central block to be encoded. Let the state space S be defined as $S=\{u \times l \times r \times d : u$ is the codeword for the block above x, l is the codeword for the left block of x, r is the codeword for the right block of x, and d is the codeword for the block below $x\}$. The vertical correlation of states u and d and the horizontal correlation of states l and r define the state s of x. We first define the difference, $\mathrm{dif}(e,f)$, between the gray levels of pixels e and f as follows:

$$\mathrm{dif}(e,f)=(\text{gray level of } e)-(\text{gray level of } f). \quad (1)$$

Then, in encoding the central block x, the smooth side-match distortions using four neighboring blocks for codeword y are defined as

$$D_u(y)=\sum_{j=1}^{n}\left|\frac{\mathrm{dif}(u_{m-1,j},u_{m,j})+\mathrm{dif}(y_{1,j},y_{2,j})}{2}-\mathrm{dif}(u_{m,j},y_{1,j})\right|, \quad (2)$$

$$D_l(y)=\sum_{i=1}^{m}\left|\frac{\mathrm{dif}(l_{i,n-1},l_{i,n})+\mathrm{dif}(y_{i,1},y_{i,2})}{2}-\mathrm{dif}(l_{i,n},y_{i,1})\right|, \quad (3)$$

$$D_r(y)=\sum_{i=1}^{m}\left|\frac{\mathrm{dif}(r_{i,2},r_{i,1})+\mathrm{dif}(y_{i,n},y_{i,n-1})}{2}-\mathrm{dif}(r_{i,1},y_{i,n})\right|, \quad (4)$$

$$D_d(y)=\sum_{j=1}^{n}\left|\frac{\mathrm{dif}(d_{2,j},d_{1,j})+\mathrm{dif}(y_{m,j},y_{m-1,j})}{2}-\mathrm{dif}(d_{1,j},y_{m,j})\right|. \quad (5)$$

Therefore, the smooth side-match distortion $D(y)$ of codeword y is defined as

$$D(y)=D_u(y)+D_l(y)+D_r(y)+D_d(y). \quad (6)$$

The selection algorithm selects N_s codewords that have smaller smooth side-match distortions among the codewords in the supercodebook for state s. These N_s codewords constitute x's state codebook. These selected codewords are sorted in the state codebook according to the smooth side-match distortions of codewords. Then, the codeword nearest to x is selected from the state codebook to replace x, and the index of this codeword is taken as the code of x.

3 The State Codebooks with Various Sizes in the VRFSVQ

In the VRFSVQ, various sizes of state codebooks are used to encode the blocks in an image. Let the size of the super-

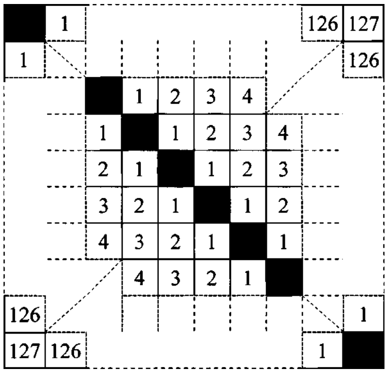

Fig. 3 The basic blocks in an image and the coding order of each nonbasic block in two coding stages.

codebook be $N=2^h-1$. When the block $B_{i,j}$ is encoded in the VRFSVQ, the smooth side-match method is applied to calculate the smooth side-match distortions for all codewords in the supercodebook. The codewords C_1, $C_2,\ldots,C_{2^h-1}$ are then sorted according to the smooth side-match distortions in the supercodebook. The codeword C_1 with the smallest smooth side-match distortion in the supercodebook is called the *best* codeword of the block $B_{i,j}$. Let S_k be the state codebook of size 2^k. Hence, k bits can represent each codeword in S_k. The codewords contained in S_k for $0 \leqslant k \leqslant h-1$ can be defined as

$$S_0=\{C_1\},$$

$$S_1=\{C_2,C_3\},$$

$$S_2=\{C_4,C_5,C_6,C_7\},$$

$$S_3=\{C_8,C_9,C_{10},C_{11},C_{12},C_{13},C_{14},C_{15}\},$$

$$\vdots$$

$$S_{h-1}=\{C_{2^{h-1}},C_{2^{h-1}+1},\ldots,C_{2^h-1}\}.$$

In the VRFSVQ, if the block $B_{i,j}$ is encoded by the state codebook S_k, then the bit rate required for $B_{i,j}$ is both the index of codewords in the state codebook S_k and the code of S_k. For example, if the block $B_{i,j}$ is encoded by the codeword C_{10}, then both the index 010 for the C_{10} in S_3 and the code of the state codebook S_3 are taken as the output for $B_{i,j}$. However, if $B_{i,j}$ is encoded by the best codeword, C_1, then only the code of S_0 is an output for $B_{i,j}$. No index for C_1 is required when S_0 is used to encode a block.

4 Design of the VRFSVQ

This section describes how to encode an image in the VRFSVQ. An encoding image, 512×512 pixels, is divided into 4×4 blocks. Thus, the size of the image can be regarded as 128×128 blocks. These blocks can be classified into two categories, basic and nonbasic blocks. In the VRFSVQ, the diagonal blocks in an image are regarded as the basic blocks. Figure 3 shows an example to illustrate the basic blocks, which are represented by the black blocks. In the VRFSVQ, the basic blocks, $B_{i,i}$ for $1 \leqslant i \leqslant 128$, are directly encoded by the supercodebook, and then the nonbasic blocks, $B_{i,j}$ for $i \neq j$, are encoded in two coding stages. The main goal of these stages is to encode these nonbasic blocks after the basic blocks are encoded in an image. Figure 3 also shows the coding order of each nonbasic block in the two coding stages. Let $C_{i,j}^1$ and $C_{i,j}^2$ indicate the codewords used to encode the nonbasic block $B_{i,j}$ in the first and second coding stages of the VRFSVQ, respectively. Initially, for each basic block $B_{i,i}$, we set $C_{i,i}^1=C_{i,i}^2=C'$, where C' indicates the encoding result of the block $B_{i,i}$ in the supercodebook. In the following, we describe how to encode the nonbasic blocks by two coding stages of the VRFSVQ.

In the first stage, each nonbasic block uses two neighboring blocks to calculate the smooth side-match distortions for all the codewords in the supercodebook to yield the state codebook. Since the diagonal blocks are selected as the basic blocks, they are encoded first. The encoded image is therefore divided into two parts, the upper triangular region and the lower triangular region. In the upper triangular region, the neighboring blocks of the currently encoded block are defined to be its left block and its lower block. In the lower triangular region, the neighboring blocks of each encoded block are defined to be its right block and its upper block. In the first coding stage, the codeword with the smallest smooth side-match distortion in the supercodebook is regarded the encoding result. That is, after the first coding stage of the VRFSVQ, each nonbasic block is encoded by its best codeword. For example, if the block $B_{i,j}$ in the upper triangular region is encoded in the first coding stage, then $B_{i,j}$ uses two codewords, $C_{i,j-1}^1$ and $C_{i+1,j}^1$, as the neighboring blocks to calculate the smooth side-match distortion for each codeword in the supercodebook, and then the codeword with the smallest smooth side-match distortion, namely $C_{i,j}^1$, is the encoding result of $B_{i,j}$.

Before these nonbasic blocks are encoded in the second coding stage of the VRFSVQ, each nonbasic block is examined for whether it is high-detail, as follows. Let the supercodebook contain N codewords, C_1, $C_2,\ldots,C_N$. Then all the codewords in the supercodebook can be sorted by variance. The variance of codeword $C_i \in R^{16}$ is denoted by $\mathrm{var}(C_i)$ and defined as

$$\mathrm{var}(C_i)=\sum_{j=1}^{16}(c_{i,j}-d_i)^2, \quad \text{where } d_i=\frac{\sum_{i=1}^{16}c_{i,j}}{16}. \tag{7}$$

Thus, let

$$\mathrm{var}(C_1) \geqslant \mathrm{var}(C_2) \geqslant \cdots \geqslant \mathrm{var}(C_k) \geqslant \cdots \geqslant \mathrm{var}(C_N). \tag{8}$$

In the second coding stage of the VRFSVQ, the high-detail and low-detail blocks can be identified according to their codes, obtained in the first stage. For example, if the nonbasic block $B_{i,j}$ is encoded by the codeword C_1 in the first stage, then the variance of $B_{i,j}$ is as high as that of C_1, and thus $B_{i,j}$ is considered to be a high-detail block in the second coding stage of the VRFSVQ. Furthermore, if the nonbasic block $B_{i,j}$ is encoded by the codeword C_N, then $B_{i,j}$ has as low a variance as C_N and is thus regarded as a low-detail block. A block is generally regarded as a high-detail block in the second coding stage of the VRFSVQ if it is encoded by one of the codewords C_1, $C_2,\ldots,C_{N/2}$ in the first coding stage; otherwise it is regarded as a low-detail block.

After the nonbasic block $B_{i,j}$ is specified as a high-detail or low-detail block, it is again encoded by the smooth side-match method in the second coding stage of the VRFSVQ. Each nonbasic block $B_{i,j}$ can use more than two neighboring blocks to generate the state codebook in the second coding stage, since all blocks in image are encoded after the first coding stage. However, the size of the state codebook used to encode each block $B_{i,j}$ in the second stage is variable. If the block $B_{i,j}$ is a high-detail block, then the large state codebook is employed to encode the block and the coding quality is enhanced; otherwise, the small state codebook is employed and the bit rate is thus reduced.

The following describes how to encode the nonbasic block $B_{i,j}$ in the second coding stage. In that stage, each nonbasic block will be encoded once again by using the smooth side-match method. For example, when the nonbasic block $B_{i,j}$ in the upper triangular region is encoded again in the second coding stage, $B_{i,j}$ uses the preencoded codewords $C^1_{i,j+1}$, $C^1_{i-1,j}$, $C^2_{i,j-1}$, and $C^2_{i+1,j}$ as the neighboring blocks to calculate the smooth side-match distortions in Eq. (6) for all the codewords in the supercodebook. Let the supercodebook size be N. The codewords C_1, $C_2,\ldots,C_N$ are then sorted according to the smooth side-match distortion in the supercodebook. Before selecting a codeword C_k for $1\leqslant k\leqslant N$, from the supercodebook to encode the block $B_{i,j}$ the distortion $D(B_{i,j},C_k)$ is defined as the Euclidean distance between $B_{i,j}$ and the codeword C_k. Then

$$D(B_{i,j},C_k)=\|B_{i,j}-C_k\|. \tag{9}$$

Let C_k be contained in the state codebook S_r. The bit rate $R(B_{i,j},C_k)$ when $B_{i,j}$ is encoded by C_k is defined as

$$R(B_{i,j},C_k)=R(S_r)+b, \tag{10}$$

where $R(S_r)$ is the number of bits required while using the state codebook S_r, and b bits are required to encode the index of C_k in S_r. The bit rate required for encoding the block $B_{i,j}$ should be as small as possible to reduce the bit rate in the VRFSVQ. Thus, the VRFSVQ selects the best codeword C_1 to encode $B_{i,j}$. Since all the neighboring blocks of $B_{i,j}$ are preencoded in the first coding stage, $B_{i,j}$ can use more than two neighboring blocks to calculate the smooth side-match distortion for each codeword in the supercodebook to find the best codeword C_1, which is better than the codeword $C^1_{i,j}$ encoded in the first coding stage by only using two neighboring blocks in the smooth side-match method. Thus, initially, $C^2_{i,j}$ is set to C_1 for the block $B_{i,j}$. Moreover, the VRFSVQ determines whether a codeword C_v exists that is more appropriate for encoding $B_{i,j}$ than is C_1, to enhance the coding quality of the encoding image. The ratio of the distortion to the bit rate, $\delta(B_{i,j},C_k)$, is defined as

$$\delta(B_{i,j},C_k)=\frac{\Delta D(B_{i,j},C_k)}{\Delta R(B_{i,j},C_k)} =\frac{D(B_{i,j},C_l)-D(B_{i,j},C_k)}{R(B_{i,j},C_k)-R(B_{i,j},C_1)} \quad \text{for } 1<k\leqslant N. \tag{11}$$

Here $\Delta R(B_{i,j},C_k)$ and $\Delta D(B_{i,j},C_k)$ indicate that the bit rate increases and distortion decreases when the best codeword, C_1, is replaced by C_k to encode $B_{i,j}$. Let $\delta(B_{i,j},C_v)$ be the maximum of $\delta(B_{i,j},C_k)$ for $1<k\leqslant N$. If $\delta(B_{i,j},C_v)>\epsilon$, then $C^2_{i,j}$ is set to C_v, which replaces C_1 for encoding $B_{i,j}$, and then the coding quality is enhanced; otherwise, the best codeword, C_1, is used to encode $B_{i,j}$ to reduce the bit rate in the VRFSVQ.

Note that the value of the threshold ϵ is positive. Let S indicate the set of nonbasic blocks in an image. Then the threshold ϵ is in the range

$$\left[0, \max_{\substack{B_{i,j}\in S\\ 1<k\leqslant N}} \delta(B_{i,j},C_k)\right].$$

A small value of ϵ emphasizes the importance of distortion, and tends to produce an encoding image with high PSNR, but requiring an increased bit rate. The PSNR for a block of size $m\times n$ is defined as

$$\text{PSNR}=10\log_{10}\frac{255^2}{\frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}(x_{i,j}-x'_{i,j})^2}, \tag{12}$$

where $x_{i,j}$ and $x'_{i,j}$ denote the original and quantized gray levels, respectively. A large value of ϵ emphasizes the importance of the bit rate, and tends to produce an encoding image with a low bit rate, but with a decreased PSNR. In the second coding stage, two thresholds, ϵ_1 and ϵ_2, are set such that $\epsilon_1<\epsilon_2$. If $B_{i,j}$ is a high-detail block, then the small threshold ϵ_1 is chosen and the coding quality of $B_{i,j}$ is enhanced. If $B_{i,j}$ is a low-detail block, the large threshold ϵ_2 is chosen and the bit rate of $B_{i,j}$ is reduced.

The design of the encoding algorithm of the VRFSVQ is summarized as follows.

Al gorithm VRFSVQ_ENCODER

Input: An image, T, which is divided into 4×4 blocks. Two thresholds, ϵ_1 and ϵ_2, where $\epsilon_1<\epsilon_2$.

Output: The code for each encoding block in T.

1. Step 1. Let the diagonal blocks in T be the basic blocks. Each basic block $B_{i,i}$ in T is directly encoded by the supercodebook. Let C' be the closest code-

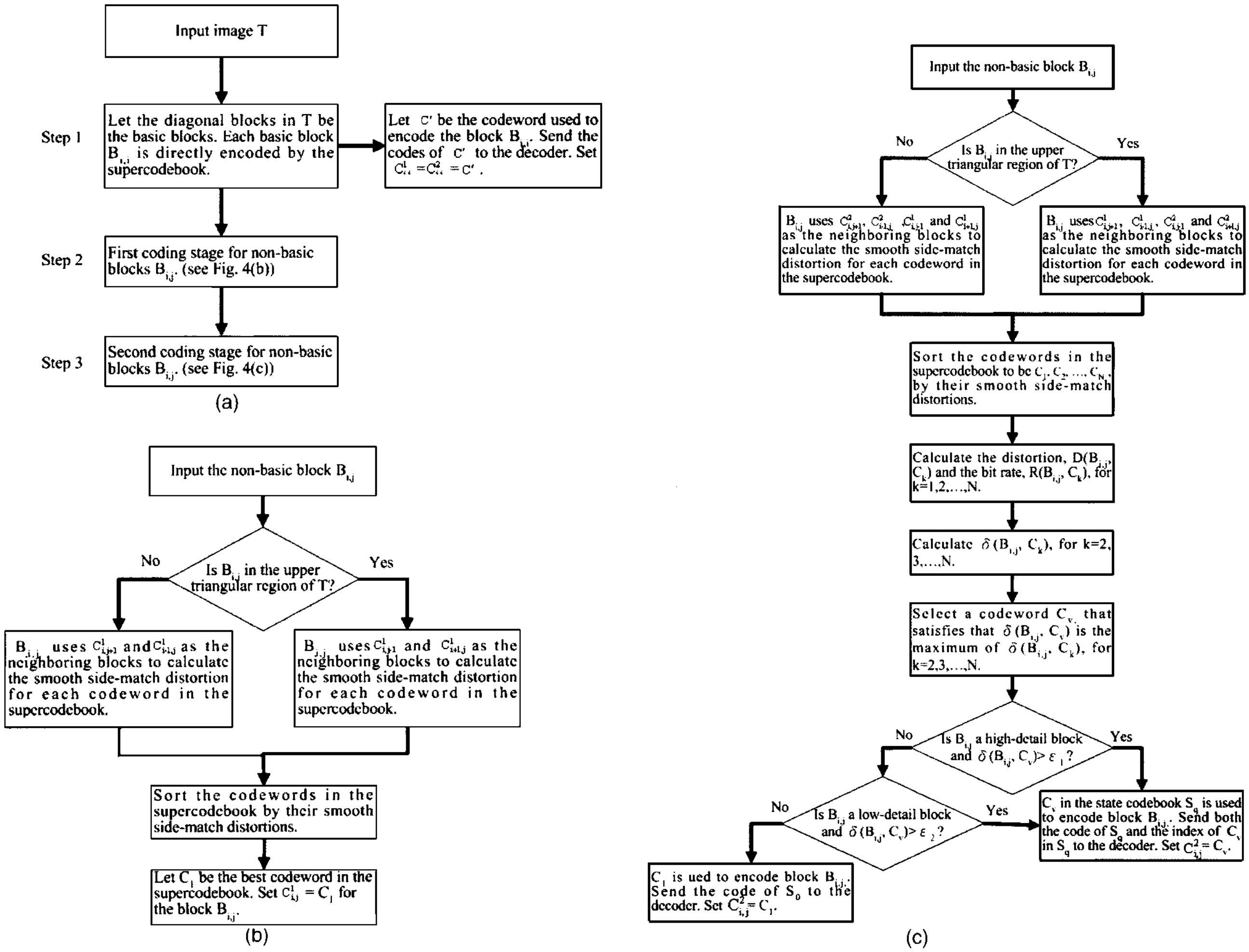

Fig. 4 Flow charts of the VRFSVQ_ENCODER algorithm: (a) the VRFSVQ encoder; (b) the first coding stage for nonbasic blocks; (c) the second coding stage for nonbasic blocks.

word to $B_{i,i}$ in the supercodebook. The code of the codeword C' for $B_{i,i}$ is taken as the output. Set $C^1_{i,i} = C^2_{i,i} = C'$.

2. Step 2. For each nonbasic $B_{i,j}$, $i \neq j$, in T, perform the following:

3. Step 2.1. If $B_{i,j}$ is in the upper triangular region (or lower triangular region) of T, then use the encoding results of two neighboring blocks of $B_{i,j}$, namely $C^1_{i,j-1}$, and $C^1_{i+1,j}$ (or $C^1_{i,j+1}$ and $C^1_{i-1,j}$), to calculate the smooth side-match distortion for each codeword in the supercodebook. Then, sort the codewords in the supercodebook by their smooth side-match distortions. Let C_1 be the codeword with the smallest smooth side-match distortion in the supercodebook. Set $C^1_{i,j} = C_1$ for $B_{i,j}$.

4. Step 3. For each nonbasic $B_{i,j}$, $i \neq j$, in T, perform the following:

5. Step 3.1. If $B_{i,j}$ is in the upper triangular region (or lower triangular region) of T, then use the encoding results of all neighboring blocks of $B_{i,j}$, namely $C^1_{i,j+1}$, $C^1_{i-1,j}$, $C^2_{i,j-1}$, and $C^2_{i+1,j}$ (or $C^2_{i,j+1}$, $C^2_{i-1,j}$, $C^1_{i,j-1}$, and $C^1_{i+1,j}$), to calculate the smooth side-match distortion for each codeword in the supercodebook. Then, sort the codewords in the supercodebook by their smooth side-match distortions. Let the size of the supercodebook be N, and the sorted codewords in the supercodebook be $C_1, C_2, \ldots, C_N$.

6. Step 3.2. For each codeword C_k, for $1 \leqslant k \leqslant N$, calculate the distortion $D(B_{i,j}, C_k)$ as in Eq. (9), and the bit rate $R(B_{i,j}, C_k)$ as in Eq. (10).

7. Step 3.3. Calculate $\delta(B_{i,j}, C_k)$ as in Eq. (11) for $1 < k \leqslant N$.

8. Step 3.4. Select a codeword C_v that satisfies the condition that $\delta(B_{i,j}, C_v)$ is the maximum of $\delta(B_{i,j}, C_k)$ for $1 < k \leqslant N$.

9. Step 3.5. Let S_q be the state codebook that contains the codeword C_v. If $B_{i,j}$ is a high-detail block [or low-detail block] and $\delta(B_{i,j}, C_v) > \epsilon_1$ [or $\delta(B_{i,j}, C_v) > \epsilon_2$], then use the codeword C_v to encode $B_{i,j}$, and let both the code of the state codebook

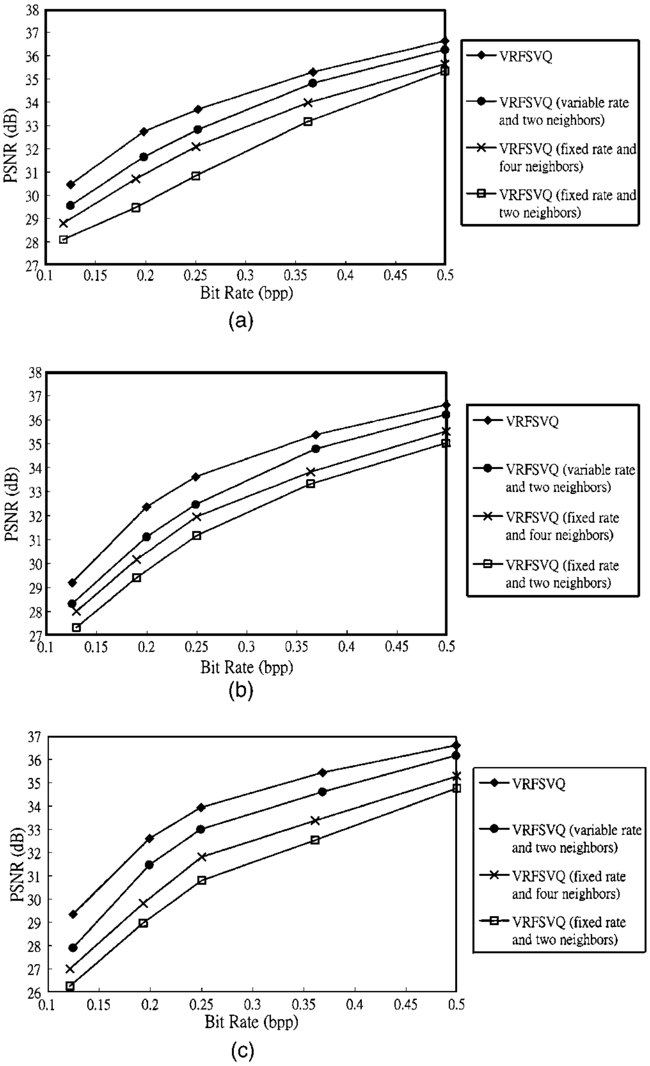

Fig. 5 Comparison of coding quality in the VRFSVQ: (a) "Lena"; (b) "F-16"; (c) "Boats."

S_q and the index of the C_v in that state codebook be the outputs. Set $C_{i,j}^2 = C_v$. Otherwise, use C_1 to encode $B_{i,j}$, regardless of whether the block is a high-detail or a low-detail one; then let the code of S_0 be the output. Set $C_{i,j}^2 = C_1$.

Figure 4 shows the flow chart to express the VRFSVQ_ENCODER algorithm. In the algorithm, steps 2 and 3 can be regarded as the first and second coding stages of the VRFSVQ, respectively. The primary goal of step 2 is to preencode all blocks, and then, in step 3, each nonbasic block can use more than two neighboring blocks to calculate the smooth side-match distortion for each codeword in the supercodebook to generate the state codebook, whereas conventional SMVQ and its variants only use two neighboring blocks. Therefore, the coding quality of each block encoded by the VRFSVQ is better than that of obtained by using conventional side-match methods. In step 3.5, the high-detail blocks can be encoded by a large state codebook, to improve the coding quality; low-detail blocks can be encoded by a small state codebook, to reduce the bit rate. The coding quality and the visual quality of the image can be enhanced in the VRFSVQ.

Finally, we describe how to design the codes of the state codebooks S_k for $0 \leqslant k \leqslant n-1$, as follows. The state codebook S_0 is often used to encode the blocks in the VRFSVQ, since the best codeword is usually the one closest to the encoding block. Therefore, the code of S_0 should be as small as possible to reduce the bit rate further. The design of the codes for the state codebooks with various sizes is similar to that in the VRFSVQ_ENCODER algorithm, and it is described as follows.

Algorithm DESIGN_STATE_CODEBOOK

Input: Q training images, $T_1, T_2, \ldots, T_Q$.

Output: The codes for the state codebooks S_k for $0 \leqslant k \leqslant n-1$.

1. Step 1. Let the size of the supercodebook be $N = 2^n - 1$. Let N_k be the number of encoding blocks whose closest codeword is found in the state codebook S_k, for $0 \leqslant k \leqslant n-1$. Set $N_k = 0$ for $0 \leqslant k \leqslant n-1$.
2. Step 2. Each T_q, $1 \leqslant q \leqslant Q$, performs the following:
3. Step 2.1. Let the diagonal blocks in T_q be the basic blocks. Each basic block $B_{i,i}$ in T_q is directly encoded by the supercodebook. Let C' be the closest codeword to $B_{i,i}$ in the supercodebook. The code of C' for $B_{i,i}$ is taken as the output. Set $C_{i,i}^1 = C_{i,i}^2 = C'$.
4. Step 2.2. For each nonbasic block $B_{i,j}$, $i \neq j$, in T_q, perform the following:
5. Step 2.2.1. If $B_{i,j}$ is in the upper triangular region (or lower triangular region) of T_q, then use the encoding results of two neighboring blocks of $B_{i,j}$, namely $C_{i,j-1}^1$ and $C_{i+1,j}^1$ (or $C_{i,j+1}^1$ and $C_{i-1,j}^1$), to calculate the smooth side-match distortion for each codeword in the supercodebook. Then, sort the codewords in the supercodebook by their smooth side-match distortions. Let C_1 be the codeword with the smallest smooth side-match distortion in the supercodebook. Set $C_{i,j}^1 = C_1$ for $B_{i,j}$.
6. Step 2.3. For each nonbasic $B_{i,j}$, $i \neq j$, in T_q, perform the following.
7. Step 2.3.1. If block $B_{i,j}$ is in the upper triangular region (or lower triangular region) of T_q, then use the encoding results of all neighboring blocks of $B_{i,j}$, namely $C_{i,j+1}^1$, $C_{i-1,j}^1$, $C_{i,j-1}^2$, and $C_{i+1,j}^2$ (or $C_{i,j+1}^2$, $C_{i-1,j}^2$, $C_{i,j-1}^1$, and $C_{i+1,j}^1$), to calculate the smooth side-match distortion for each codeword in the supercodebook. Then, sort the codewords in the supercodebook by their smooth side-match distortions. Then, sort the codewords $C_1, C_2, \ldots, C_{2^n-1}$ according to the smooth side-match distortion in the supercodebook. Let C_q be the closest codeword to

$B_{i,j}$ in the supercodebook and be contained in the state codebook S_r. Then replace $B_{i,j}$ by the codeword C_q. Set $N_r \leftarrow N_r + 1$.

8. Step 3. Let σ_k be the average N_k of a codeword in S_k. Then calculate $\sigma_k = N_k/2^k$ for $0 \leqslant k \leqslant n-1$.
9. Step 4. Construct a Huffman tree, based on the value of σ_k, for $0 \leqslant k \leqslant n-1$. Each leaf node in the Huffman tree is represented by S_k for $0 \leqslant k \leqslant n-1$. Each leaf node in the Huffman tree is traveled, and the code for each state codebook S_k for $0 \leqslant k \leqslant n-1$ is obtained.

5 Experimental Results

In the experiments, two training data sets, T_1={“Cafe”, “Chart”, “Zelda”, “Baboon”, “Crowd”} and T_2={“Drop”, “Tiffany”, “Bridge”, “Cameraman”, “Eff-tower”}, were used to test the performance of the VRFSVQ. Each training image (512×512 pixels with 256 gray levels) is divided into 4×4 blocks. The total number of training blocks is 81,920 for each training data set. The LBG algorithm[20] is applied to design the supercodebook for 4×4 blocks, and the supercodebook size is $1023 = 2^{10} - 1$ for each training data set. These two training data sets were also used to design the codes for the state codebooks S_k for $0 \leqslant k \leqslant 9$. Ten testing images, “Goldhill,” “Peppers,” “Bike,” “Lena,” “Barbara,” “Women,” “F-16,” “Boat,” “Face,” and “Natural,” not included in the two training data sets, were used to test the coding performance of the VRFSVQ. Each test image is also divided into 4×4 blocks before being coded in the VRFSVQ. In the VRFSVQ, the basic blocks in the image are directly encoded by the supercodebook, and the nonbasic blocks are then encoded by the VRFSVQ. In the first coding stage of the VRFSVQ, each nonbasic block is encoded by the best codeword obtained from the supercodebook. In the second coding stage, all nonbasic blocks in an image can be encoded again by the smooth side-match method using four neighboring blocks to yield the state codebook. Each nonbasic block can be encoded in the second coding stage by the variable-size state codebook. However, when that is done, the coding quality depends on the value of a threshold. A large threshold emphasizes the importance of the bit rate and tends to reduce the bit rate of an encoding image. A small threshold emphasizes the importance of distortion and tends to produce an encoding image with a high PSNR. Therefore, in this paper, the high-detail blocks use the small threshold, ϵ_1, to improve the coding quality, and the low-detail blocks use the large threshold, ϵ_2, to reduce the bit rate.

Let the threshold ϵ_2 be $2\epsilon_1$. Figure 5 compares the coding quality between the variable and fixed sizes of codebooks used to encode all nonbasic blocks in the second coding stage of the VRFSVQ, when the training data set T_1 is used to design the supercodebook of the VRFSVQ. In Fig. 5, the VRFSVQ indicates that each nonbasic block uses four neighboring blocks in the smooth side-match method to yield the variable size of the state codebook. The pairs of thresholds $(\epsilon_1, \epsilon_2) = (20,40)$, (15,30), (10,20), (5,10), and (2,4) are used to encode images in the second coding stage. The bit rate of the testing image cannot be further reduced if the values of both thresholds, ϵ_1 and ϵ_2,

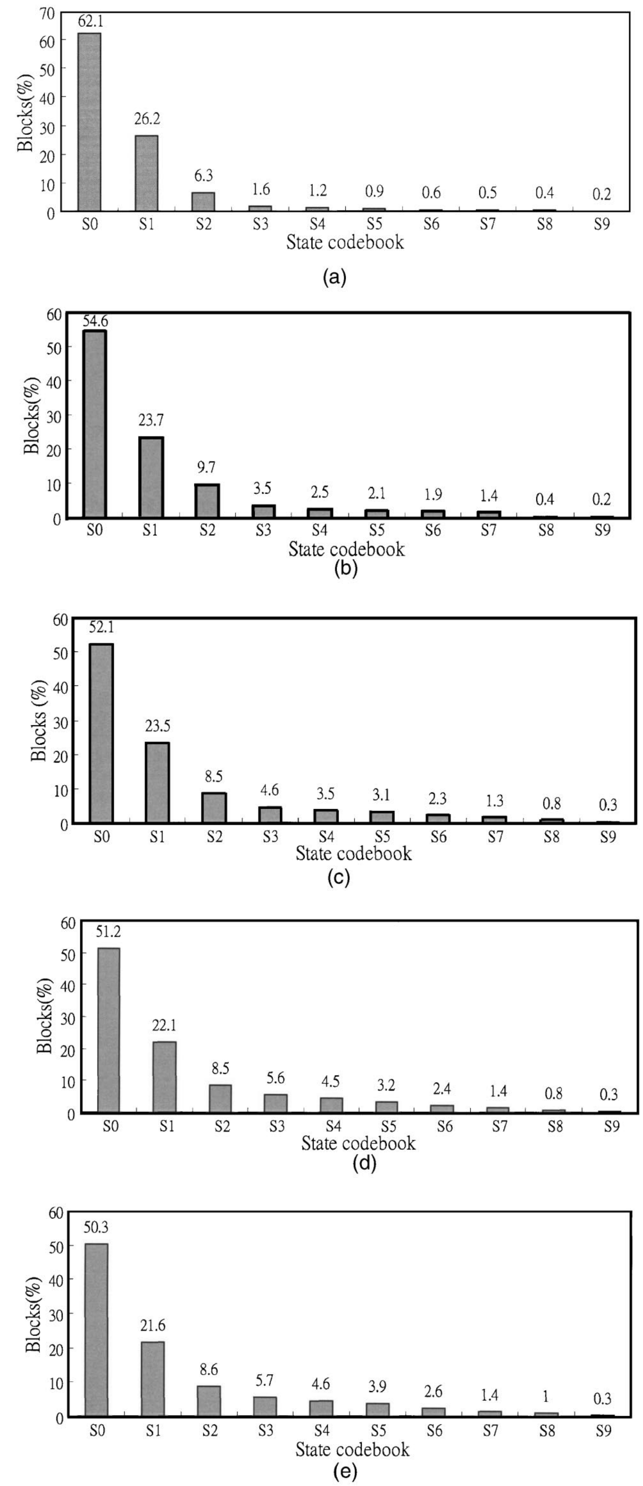

Fig. 6 The percentage of blocks encoded by the variable state codebooks when the “Lena” image is encoded by the VRFSVQ: (a) $\epsilon_1 = 20$ and $\epsilon_2 = 40$; (b) $\epsilon_1 = 15$ and $\epsilon_2 = 30$; (c) $\epsilon_1 = 10$ and $\epsilon_2 = 20$; (d) $\epsilon_1 = 5$ and $\epsilon_2 = 10$; (e) $\epsilon_1 = 2$ and $\epsilon_2 = 4$.

are large enough, because all nonbasic blocks are encoded by the best codeword in the VRFSVQ, to minimize the bit rate of the encoding image. Also, if both ϵ_1 and ϵ_2 are set sufficiently low, then the PSNR of the testing image can be maximized, since all nonbasic blocks are encoded by the

Table 1 PSNR comparisons between VRFSVQ and the other methods.

Test image	Method	PSNR 0.0625	0.125	0.25	0.5	1 bpp
Goldhill	JPEG 2000	24.65	27.43	30.61	33.30	36.65
	SSMVQ[16]	24.12	27.06	30.13	32.92	36.25
	SMVQ[10]	23.88	26.80	29.93	32.65	35.87
	HTSMVQ[15]	24.26	27.08	30.18	32.98	36.26
	VRFSVQ(T_1)	24.34	27.23	30.60	33.39	36.72
	VRFSVQ(T_2)	24.36	27.25	30.62	33.42	36.74
Peppers	JPEG 2000	27.35	30.47	33.33	37.51	40.57
	SSMVQ	26.57	29.94	32.95	37.12	39.78
	SMVQ	26.34	29.03	32.13	36.89	39.13
	HTSMVQ	26.65	30.03	33.02	37.15	40.02
	VRFSVQ(T_1)	26.70	30.18	33.34	37.52	40.58
	VRFSVQ(T_2)	27.64	30.10	33.32	37.49	40.53
Bike	JPEG 2000	28.16	31.11	34.16	37.29	40.36
	SSMVQ	27.57	30.84	33.82	36.84	39.87
	SMVQ	27.02	30.43	33.56	36.47	39.65
	HTSMVQ	27.60	30.88	33.92	36.98	39.94
	VRFSVQ(T_1)	27.83	30.93	34.19	37.30	40.40
	VRFSVQ(T_2)	27.79	30.89	34.17	37.29	40.38
Lena	JPEG 2000	27.85	30.49	33.41	36.65	40.35
	SSMVQ	27.26	29.95	33.04	36.22	39.97
	SMVQ	26.83	29.43	32.67	35.98	39.43
	HTSMVQ	27.28	30.03	33.12	36.25	40.03
	VRFSVQ(T_1)	27.63	30.45	33.46	36.74	40.44
	VRFSVQ(T_2)	27.57	30.32	33.42	36.73	40.43
Barbara	JPEG 2000	22.34	25.43	28.40	32.22	37.16
	SSMVQ	21.57	24.85	27.97	31.86	36.79
	SMVQ	21.02	24.52	27.62	31.57	36.43
	HTSMVQ	21.66	24.94	28.12	31.95	36.81
	VRFSVQ(T_1)	21.98	25.42	28.46	32.33	37.31
	VRFSVQ(T_2)	22.02	25.45	28.48	32.40	37.43
Women	JPEG 2000	25.67	27.43	30.15	33.81	38.67
	SSMVQ	24.82	26.93	29.93	33.77	38.62
	SMVQ	24.33	26.55	29.45	33.12	38.10
	HTSMVQ	24.85	26.95	29.95	33.78	38.63
	VRFSVQ(T_1)	24.92	27.23	30.12	33.98	38.87
	VRFSVQ(T_2)	24.89	27.21	30.11	33.96	38.86
F-16	JPEG 2000	26.21	29.32	33.53	36.43	39.54
	SSMVQ	25.56	28.83	33.21	36.22	39.32
	SMVQ	25.32	28.43	32.98	35.89	38.98
	HTSMVQ	25.57	28.85	33.30	36.24	39.34
	VRFSVQ(T_1)	26.02	29.12	33.54	36.56	39.73
	VRFSVQ(T_2)	26.03	29.14	33.53	36.55	39.75
Boat	JPEG 2000	26.45	29.54	33.89	36.54	39.21
	SSMVQ	26.01	29.28	33.53	36.23	39.01
	SMVQ	25.78	28.89	33.01	35.96	38.89
	HTSMVQ	26.13	29.30	33.58	36.34	39.04
	VRFSVQ(T_1)	26.27	29.32	33.90	36.58	39.34
	VRFSVQ(T_2)	26.28	29.31	33.89	36.59	39.35
Face	JPEG 2000	32.27	34.15	36.54	39.54	43.26
	SSMVQ	31.95	33.78	36.01	39.05	42.67
	SMVQ	29.88	32.12	35.02	38.88	42.44
	HTSMVQ	32.01	33.82	36.08	39.12	42.73
	VRFSVQ(T_1)	32.03	33.84	36.63	39.58	43.33
	VRFSVQ(T_2)	31.98	33.81	36.59	39.55	43.30
Natural	JPEG 2000	22.65	24.83	27.58	31.32	36.19
	SSMVQ	22.02	24.21	27.06	30.98	35.88
	SMVQ	21.89	23.96	26.88	30.65	35.02
	HTSMVQ	22.15	24.32	27.13	31.01	35.91
	VRFSVQ(T_1)	22.21	24.44	27.61	31.43	38.36
	VRFSVQ(T_2)	22.24	24.50	27.62	31.44	38.39

Table 2 MSSIM comparisons between VRFSVQ and JPEG 2000.

Test Image	Method	MSSIM				
		0.0625	0.125	0.25	0.5	1 bpp
Goldhill	JPEG 2000	0.6354	0.7854	0.8924	0.9037	0.9274
	VRFSVQ(T_1)	0.6310	0.7841	0.8933	0.9043	0.9280
	VRFSVQ(T_2)	0.6315	0.7843	0.8934	0.9048	0.9281
Peppers	JPEG 2000	0.6487	0.7957	0.8849	0.9263	0.9353
	VRFSVQ(T_1)	0.6423	0.7946	0.8850	0.9261	0.9355
	VRFSVQ(T_2)	0.6424	0.7943	0.8852	0.9265	0.9356
Bike	JPEG 2000	0.6558	0.8034	0.8906	0.9256	0.9432
	VRFSVQ(T_1)	0.6457	0.7998	0.8957	0.9267	0.9468
	VRFSVQ(T_2)	0.6455	0.7956	0.8962	0.9263	0.9465
Lena	JPEG 2000	0.6573	0.7985	0.8933	0.9182	0.9363
	VRFSVQ(T_1)	0.6534	0.7980	0.8935	0.9192	0.9401
	VRFSVQ(T_2)	0.6537	0.7979	0.8934	0.9189	0.9398
Barbara	JPEG 2000	0.6653	0.8037	0.9032	0.9274	0.9462
	VRFSVQ(T_1)	0.6610	0.7998	0.9037	0.9374	0.9488
	VRFSVQ(T_2)	0.6573	0.7893	0.9011	0.9321	0.9468
Women	JPEG 2000	0.6592	0.7893	0.8940	0.9211	0.9374
	VRFSVQ(T_1)	0.6423	0.7738	0.8943	0.9268	0.9402
	VRFSVQ(T_2)	0.6410	0.7729	0.8901	0.9255	0.9400
F-16	JPEG 2000	0.6539	0.7993	0.8892	0.9289	0.9374
	VRFSVQ(T_1)	0.6502	0.7981	0.8960	0.9320	0.9392
	VRFSVQ(T_2)	0.6498	0.7955	0.8920	0.9298	0.9380
Boat	JPEG 2000	0.6537	0.8028	0.9028	0.9367	0.9402
	VRFSVQ(T_1)	0.6394	0.7993	0.9044	0.9396	0.9455
	VRFSVQ(T_2)	0.6392	0.7935	0.9032	0.9389	0.9422
Face	JPEG 2000	0.6483	0.7939	0.8895	0.9295	0.9428
	VRFSVQ(T_1)	0.6455	0.7901	0.8893	0.9302	0.9455
	VRFSVQ(T_2)	0.6456	0.7902	0.8884	0.9312	0.9456
Natural	JPEG 2000	0.6528	0.8124	0.9024	0.9372	0.9462
	VRFSVQ(T_1)	0.6478	0.8025	0.9033	0.9383	0.9477
	VRFSVQ(T_2)	0.6475	0.8012	0.9027	0.9380	0.9475

codeword that has the maximum ratio of distortion to bit rate, as specified in Eq. (11). However, each nonbasic block can be encoded by the state codebook of fixed size in the second coding stage. When the state codebook of fixed size is employed to encode all nonbasic blocks in the VRFSVQ, only the index of the codewords in the state codebook is required for each nonbasic block, and the code of the state codebook used to encode each nonbasic block can be omitted.

In Fig. 5, the fixed state-codebook sizes 4, 8, 16, 32, and 64 are applied to encode all nonbasic blocks in the second coding stage. Notably, in Fig. 5, the bit rate of the image encoded using the variable state-codebook size is less than that of the image encoded by the state codebook of fixed size, for a given PSNR, because increasing the PSNR when the state codebook of fixed size is used requires that the state codebook be large to encode all nonbasic blocks. However, not all blocks require a large state codebook to enhance coding quality. The use of a large state codebook is inappropriate when the best codeword is the one closest to the encoding block. Therefore, using a state codebook of variable size can reduce the bit rate below that obtained when a state codebook of fixed size is used. Furthermore, Fig. 5 shows that using four neighboring blocks to calculate the smooth side-match distortions in the second coding

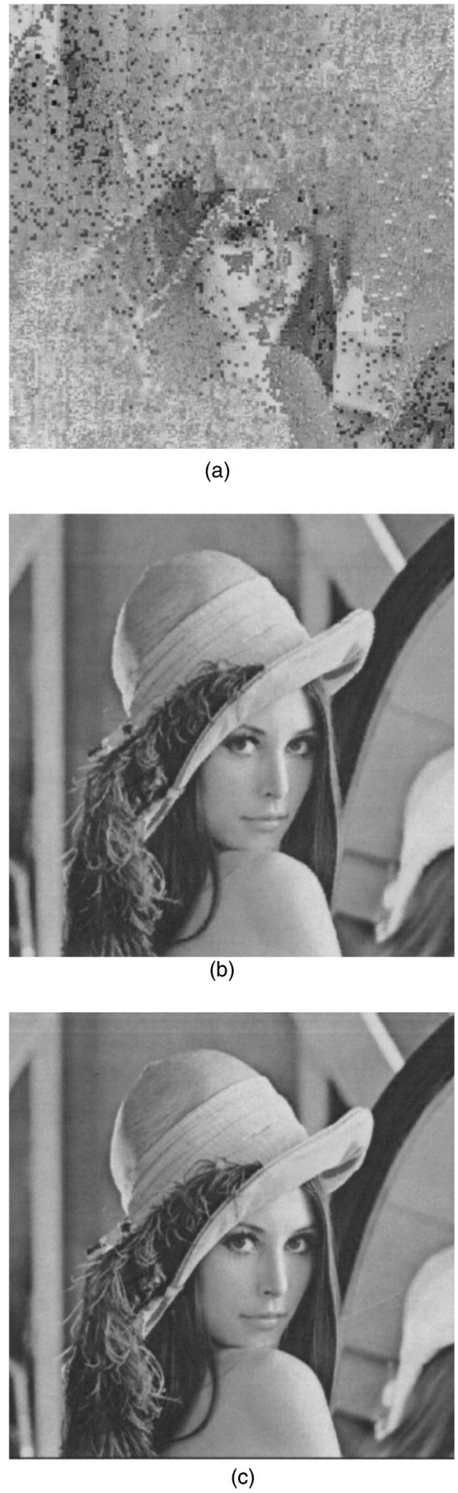

Fig. 7 The coding quality of the "Lena" image using the VRFSVQ: (a) after the first coding stage (9.7 dB, 0.0049 bpp); (b) after the second coding stage (33.46 dB, 0.25 bpp); (c) the original image.

(a)

(b)

(c)

Fig. 8 The coding quality of the "F-16" image using the VRFSVQ: (a) after the first coding stage (8.3 dB, 0.0049 bpp); (b) after the second coding stage (33.53 dB, 0.25 bpp); (c) the original image.

stage of the VRFSVQ yields a higher PSNR than using two neighboring blocks.

Figure 6 reveals that most of the blocks in the "Lena" image can be encoded using the state codebook S_0 in the second coding stage of the VRFSVQ. That is, the best codeword is usually suitable for encoding blocks in the

(a)

(b)

(c)

Fig. 9 The coding quality of the "Boat" image using the VRFSVQ: (a) after the first coding stage of VRFSVQ. (7.8 dB, 0.0049 bpp); (b) after the first coding stage of VRFSVQ. (33.90 dB, 0.25 bpp); (c) the original image.

VRFSVQ. The percentage of blocks encoded by a larger state codebook increases as the value of the pair (ϵ_1, ϵ_2) decreases. Table 1 shows the coding quality of the testing images obtained using the VRFSVQ and other methods. In Table 1, VRFSVQ(T_1) and VRFSVQ(T_2) indicate the

VRFSVQ using the training data sets T_1 and T_2, respectively. They have approximately equal coding qualities at the same bit rate. Hence, the performance of the VRFSVQ is insensitive to the training data set used. The VRFSVQ outperforms other SMVQ methods, and has a larger PSNR than does JPEG 2000 at bit rates of larger than 0.25 bpp in most cases.

Table 2 compares VRFSVQ and JPEG 2000 in mean structural similarity (MSSIM).[21] We observe that VRFSVQ and JPEG 2000 have approximately the same MSSIM, and the users cannot easily distinguish the decoding images obtained by VRFSVQ and JPEG 2000 when the bit rate is the same. Figures 7(a), 8(a), and 9(a) present the coding quality of the images produced using only the first coding stage of the VRFSVQ. Although the PSNR is not very high, the required bit rate is extremely small. The reason is that the bit rate at which the images are encoded is governed only by the codes of the basic blocks. The second coding stage of the VRFSVQ involves continued encoding of the images, as displayed in Figs. 7(b), 8(b), and 9(b).

6 Conclusions

This paper demonstrates the feasibility of using the VRFSVQ for image coding. The VRFSVQ includes two coding stages. In the first stage, the VRFSVQ predicts a codeword to encode each nonbasic block in the image. Then, in the second stage, each nonbasic block can be encoded again using more than two neighboring blocks in the smooth side-match method. Furthermore, in the second stage, each nonbasic block can be encoded by using the variable size of the state codebook. Two thresholds are given to control the coding quality of the image in the second coding stage. A large threshold is chosen for low-detail blocks, which can use the smaller state codebook to produce a low bit rate of the encoding image. A low threshold is chosen for high-detail blocks, which can use the larger state codebook to increase the PSNR. As indicated, the VRFSVQ achieves high coding quality and a low bit rate for image coding.

Acknowledgment

This work was supported by the National Science Council of the Republic of China under Contract NSC92-2213-E-426-002.

References

1. A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression*, Boston: Kluwer (1992).
2. N. M. Nasrabadi and R. A. King, "Image coding using vector quantization: A review," *IEEE Trans. Commun.* **36**, 957–971 (1988).
3. M. O. Dunham and R. M. Gray, "An algorithm for the design of label-transition finite-state vector quantizers," *IEEE Trans. Commun.* **33**(1), 83–89 (1985).
4. J. Foster, R. M. Gray, and M. O. Dunham, "Finite-state vector quantization for waveform coding," *IEEE Trans. Inf. Theory* **31**(3), 348–359 (1985).
5. W. T. Chen, R. F. Chang, and J. S. Wang, "Image sequence coding using adaptive finite-state vector quantization," *IEEE Trans. Circuits Syst. Video Technol.* **2**, 15–24 (1992).
6. S. A. Rizvi and N. M. Nasrabadi, "Finite-state residual vector quantizer using a tree-structured competitive neural network," *IEEE Trans. Circuits Syst. Video Technol.* **7**, 377–390 (1997).
7. N. M. Nasrabadi, C. Y. Choo, and Y. Feng, "Dynamic finite-state vector quantization," *IEEE Trans. Commun.* **COM-42**, 2145–2154 (1994).
8. N. M. Nasrabadi and S. A. Rizvi, "Next-state functions for finite-state vector quantization," *IEEE Trans. Image Process.* **4**, 1592–1601 (1995).
9. J. C. Tsai, C. H. Hsieh, and T. C. Hsu, "A new dynamic finite-state vector quantization algorithm for image compression," *IEEE Trans. Image Process.* **9**(11), 1825–1836 (2000).
10. T. Kim, "Side match and overlap match vector quantizers for images," *IEEE Trans. Image Process.* **1**(2), 170–185 (1992).
11. R. F. Chang and W. T. Chen, "Image coding using variable-rate side-match finite-state vector quantization," *IEEE Trans. Image Process.* **2**(1), 104–108 (1993).
12. R. F. Chang and W. M. Chen, "Adaptive edge-based side-match finite-state classified vector quantization with quadtree map," *IEEE Trans. Image Process.* **5**(2), 378–383 (1996).
13. T. S. Chen and C. C. Chang, "A new image coding algorithm using variable-rate side-match finite-state vector quantization," *IEEE Trans. Image Process.* **6**(8), 1185–1187 (1997).
14. J. K. Chung and C. S. Lin, "Viterbi-based algorithm for side-match vector quantization over noisy channels," *IEEE Trans. Commun.* **44**(11), 1455–1465 (1996).
15. H. C. Wei, P. C. Tsai, and J. S. Wang, "Three-sided side match finite-state vector quantization," *IEEE Trans. Circuits Syst. Video Technol.* **10**(1), 51–58 (2000).
16. S. B. Yang and L. Y. Tseng, "Smooth side-match classified vector quantizer with variable block size," *IEEE Trans. Image Process.* **10**(5), 677–685 (2001).
17. S. B. Yang, "Side-match tree-structured vector quantizer for image progressive coding," *IEE Proc. Vision Image Signal Process.* **150**(1), 6–13 (2003).
18. S. B. Yang, "General-tree-structured vector quantizer for image progressive coding using the smooth side-match method," *IEEE Trans. Circuits Syst. Video Technol.* **13**(2), 193–202 (2003).
19. A. Said and W. A. Pearlman, "New, fast, and efficient image codec based set partition in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.* **6**, 243–249 (1996).
20. Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. Commun.* **28**(1), 84–95 (1980).
21. Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error measurement to structural similarity," *IEEE Trans. Image Process.* **13**(1), 1–14 (2004).

Shiueng-Bien Yang received the BS degree in 1993 and the PhD degree in 1999, both from the Department of Applied Mathematics, National Chung Hsing University, Taichung, Taiwan. He is currently an associate professor of Department of Computer Science and Information Engineering, Leader University, Tainan City, Taiwan. His research interests include pattern recognition, speech coding, image processing, and neural networks.