Gradient Match and Side Match Fractal Vector Quantizers for Images

Hsuan T. Chang

Department of Electrical Engineering
National Yunlin University of Science and Technology
Touliu Yunlin, 640 TAIWAN R.O.C.
E-mail: htchang@pine.yuntech.edu.tw

Abstract

In this paper we propose the *gradient match fractal vector quantizers* (GMFVQs) and the *side match fractal vector quantizers* (SMFVQs), which are two classes of *finite state fractal vector quantizers* (FSFVQs), for the image coding framework. In our previous work, we proposed the non-iterative fractal block coding (FBC) technique to improve the decoding speed and the coding performance for conventional FBC techniques. To reduce the number of bits for denoting the fractal code of the range block, the concepts of the gradient match vector quantizers (GMVQs) and the side match vector quantizers (SMVQs) are employed to the non-iterative FBC technique. Unlike the ordinary vector quantizers, the super codebooks in the proposed GMFVQs and SMFVQs are generated from the affine-transformed domain blocks in the non-iterative FBC technique. The codewords in the state codebook are dynamically extracted from the super codebook with the side-match and gradient-match criteria. The redundancy in the affine-transformed domain blocks is greatly reduced and the compression ratio can be significantly increased. Our simulation results show that 15%–20% of the bit rates in the non-iterative FBC technique are saved by using the proposed GMFVQs.

Keywords: fractal block coding, finite state vector quantization, side match, gradient match, fractal vector quantizer.

December 12, 2001

1 Introduction

Fractal block coding (FBC) and vector quantization (VQ) are two efficient techniques for image coding frameworks. There exists many similarities between both techniques so that the FBC technique is also called the *self-VQ* technique. On the other hand, the major differences between two techniques include: (1) VQ records the indices of the codewords in the codebook and FBC technique records the fractal code of the range blocks, (2) VQ's codebook is pre-designed and FBC's domain blocks are extracted from the image itself, (3) the encoder and the decoder of VQ have the same codebook but we cannot obtain the same domain blocks in the conventional FBC technique. (4) the computational load of the FBC encoding and decoding processes is much higher than that of VQ.

There have been much research work proposed to improve FBC technique via employing the hybrid (both VQ and FBC) techniques [1]–[9]. For example, the low frequency components of an image are coded by VQ, and its residual is coded by the fractal approximation [1]. The Lloyd algorithm for VQ can be used to reduce the number of domain blocks such that the lower encoding complexity can be achieved [5]. The mean shape-gain vector quantization (MSGVQ) is combined with the fractal image compression in [3]. A fractal vector quantizer (FVQ) was proposed to coarsely approximate the source image by fixed basis blocks, and the codebooks are self-trained from the approximated image [4]. However, the concept of the finite state machine for the state codebook design of finite state vector quantizers (FSVQs) [10] is not found in the papers above. FSVQs save the bit rate for the ordinary VQ technique and preserve the image fidelity very well. This is because the state codebook size is always much smaller than the super codebook size and the codewords in the state codebooks are selected according to the next-state function. Therefore, in this paper we will employ the finite state algorithm on the FBC technique to significantly reduce the bit rate.

There were many next-state functions proposed to design the state codebooks in FSVQs [10]. Side match vector quantizers (SMVQs) [11] were proposed to preserve the spatial continuity between block boundaries. In SMVQs, the best selection of the state codebook is the set of codewords whose boundary pixels are the most similar to the reproduction pixels that contribute to the state generation. The bit rate can be drastically reduced when they are used together with variable length noiseless coding. Recently, we proposed the gradient match vector quantizers (GMVQs) [12], the considerably general cases of SMVQs, which can outperform SMVQs in both the bit rate and the peak noise-to-signal ratio (PSNR). The quantization noise in the reconstructed image by the use of GMVQs is less visible than that in the case of SMVQs under the same bit rate.

In conventional FBC techniques, however, we cannot obtain the same domain pool (i.e., the codebook) in both the encoder and the decoder. The concept of FSVQs cannot be directly applied to conventional FBC techniques to design the state codebooks. On the other hand, more than ten bits are required to represent the four parameters (block mean μ, contrast scaling α, isometry ι, and the position of domain block P_D) in the fractal code. Thus using the FBC technique to encode small range blocks is inefficient. The number of the codewords in the state codebook of FSVQs is usually less than 512 to achieve a low bit rate. Therefore, to achieve a lower bit rate, the number of the bits to denote four parameters should be reduced to encode the range block.

Recently, we proposed a non-iterative FBC technique [9], [13] to solve the problem of long decoding time due to the iteration process. Since the block mean is one of the parameters in the fractal code, the domain pool is generated from the mean image whose pixel values are the block means of all the range blocks in the encoder. The decoder receives the on-line transmitted mean information and thus can reconstruct an identical domain pool. The

finite state algorithm now can be applied to the non-iterative FBC technique since both the encoder and the decoder thus have the same domain pool and domain blocks. The gradient match and side match criteria used in the next-state function design have shown their superiority to reduce the redundancy among the codewords in the super codebook. We here propose the *gradient match fractal vector quantizers* (GMFVQs) and the *side match fractal vector quantizers* (SMFVQs) for image coding framework. Both techniques are two classes of the *finite state fractal vector quantizers* (FSFVQs) that employ the finite state algorithms to design the state codebooks for the non-iterative FBC technique. Instead of recording the fractal code in the FBC technique, the block mean and the indices of the codewords in the state codebooks are used to denote the encoded range blocks. The bit rate is significantly reduced since the number of codewords in the state codebook is much smaller than that of the codewords in the super codebook, which are all possible combinations of the affine-transformed domain blocks in the non-iterative FBC technique.

2 Finite-State Fractal Vector Quantizers (FSFVQs)

The proposed FSFVQs are based upon the non-iterative FBC technique and the FSVQs. Since the GMVQs and SMVQs are two classes of FSVQs, the proposed GMFVQs and SMFVQs can also be considered as two classes of FSFVQs. FSVQs and FSFVQs can share the same next-state functions to design the state codebooks although their construction methods of the super codebooks are different,

A. Encoder and Decoder

The block diagrams of the encoder and the decoder in the proposed FSFVQs are shown in Figure 1. As shown in Figure 1(a), an image is first partitioned into non-overlapping range blocks. After measuring the mean value and the variance of each range block, a mean

image can be constructed for the encoding purpose. The domain blocks Ds are obtained by subsampling the mean image with a T-pixel period. If the variance is less than a threshold value V_{th}, the range block is coded by its mean value. Otherwise, we perform the affine transformation to determine the fractal code of the range block if the range block is located at the first row or first column. An attached header is used to denote the coding status of a range block. The detailed procedures of the non-iterative FBC can be found in our previous work [9], [13]. For the range blocks that are not located at the first row or the first column, they are coded by the proposed FSFVQs technique that will be described below. In FSVQs, the range blocks at the first row or the first column are not necessary to be coded at first. Other arrangements such as the blocks at diagonal axis [14] and at sampled positions [15] have been proposed to improve the coding performance of FSVQs.

In conventional FBC techniques, the diversity of both the contrast scaling and isometry operations complicates the encoding process very much. To denote these two parameters, usually more than six bits are required. If the block size is smaller than 4×4, the compression ratio is low since the number of bits for the fractal code of a range block may be more than 20. For example, it requires 21 bits to denote the fractal code if we choose one bit for the header, three bits for contrast scaling factor, three bits for isometry types, six bits for mean value, and eight bits for domain block's position. From the viewpoint of FSVQs, all of the possible affine-transformed domain blocks are the codewords in a super codebook. That is, there would be a huge number of codewords, which construct a huge and inefficient super codebook. The finite state algorithms can be used to design the state codebook of the much smaller size than the super codebook. The range blocks are coded by the indices by using the nearest neighboring search in the state codebook. In addition to the mean value, the output of the encoder is the index of the codeword in the state codebook except for the range

blocks located at the first row or the first column (they are coded by the affine transform).

The decoder is basically a symmetric version of the encoder, especially for the FSVQ part. First, the mean image is reconstructed from the mean values of all range blocks. Then the same super codebook as that in the encoder is reconstructed. The state codebook for each range block is reconstructed with the next-state function. According to the headers, the range blocks are reconstructed sequentially. For the range blocks located at the first row and the first column, they are decoded either by the mean value or by the affine transform. As to the range blocks which are located elsewhere and not coded by mean, they are represented by the indices of the codewords in the state codebook. These range blocks can be directly reconstructed by the table-look-up operation. Note that the initial state in both the encoder and the decoder should be the same.

B. State Codebook Design

Instead of using training images to generate the super codebook, we use all of possible affine-transformed domain blocks to construct the super codebook. The codeword $\mathbf{c}$ is defined as

$$\mathbf{c} = \iota\{\alpha(D - \mu_D) + \mu_{\mathbf{x}}\}, \tag{1}$$

where μ_D and $\mu_{\mathbf{x}}$ denote the mean values of the domain block and the range block respectively. Since the domain blocks are selected from the mean image of the test image, the constructed super codebook in the proposed FSFVQs is image-dependent. The test image is like one of the training images in codebook design. Thus a better coding performance can be expected. Unfortunately, the number of the codewords in this super codebook is much greater than that in common vector quantizers. For example, if we use three bits for contrast scaling factor, three bits for isometry, and select 1024 domain blocks from the mean image, there are $2^3 \times 2^3 \times 1024 = 65536$ affine-transformed domain blocks (i.e., codewords) in the

super codebook. The required number to denote the index of the codeword is 16, which is inefficient for encoding 4×4 blocks (the bit rate will be more than 1 bit/pixel). To reduce the bit rate, it should design the state codebooks of the size much smaller than the super codebook.

The state codebook is dynamically generated by the next-state function in the proposed GMFVQs and SMFVQs. There were many methods proposed to design the next-state function in FSVQs [10]. Recently, the SMVQs [11] the GMVQs [12] are proposed to exploit the spatial correlation and preserve the spatial continuity of boundaries between adjunct blocks. Both techniques greatly improve the compression ratio of the ordinary VQ technique while the PSNR performance is well preserved. The block diagram of SMVQ and GMVQ is shown in Figure 2. In the next-state function design, only two past reproduction blocks, $\mathbf{x'}_{\text{north}}$ and $\mathbf{x'}_{\text{west}}$, at the north and the west of the current block $\mathbf{x}$ contribute to the generation of the state. Then the state s and the next-state function f is represented as

$$s = f(\mathbf{x'}_{\text{north}}, \mathbf{x'}_{\text{west}}). \tag{2}$$

As shown in Figure 3, both blocks are continuous neighbors of the current block $\mathbf{x}$ in the north and the west directions. The labeled area in Figure 3 corresponds to the pixels that contribute to the state generation for encoding the current block. The next-state functions for SMVQs and GMVQs use the side match and gradient match criteria to select the codewords from the super codebook respectively. For each affine-transformed domain block in the super codebook, we measure its gradient match error and side match error. Suppose that an input block $\mathbf{x}$ is of size $m \times n$. The corresponding state codebook (SC) is a subset of the super codebook that contains N codewords with the smallest gradient match error E_{gm} or with the smallest side match error E_{sm}. The gradient match error E_{gm} for a codeword placed in

the position of the current block $\mathbf{x}$ is defined as [12]

$$\begin{aligned} E_{\mathrm{gm}} &= E_{\mathrm{gm_v}} + E_{\mathrm{gm_h}} \\ &= \sum_{j=1}^{n}(|u_{m-1,j} - 2u_{m,j} + x_{1,j}| + |u_{m,j} - 2x_{1,j} + x_{2,j}|)^2 \\ &\quad + \sum_{i=1}^{m}(|l_{i,n-1} - 2l_{i,n} + x_{i,1}| + |l_{i,n} - 2x_{i,1} + x_{i,2}|)^2, \end{aligned} \tag{3}$$

where $E_{\mathrm{gm_v}}$ and $E_{\mathrm{gm_h}}$ represent the gradient match error for the block boundaries in vertical and horizontal directions, respectively. On the other hand, the side match error of a codeword placed in the position of the current block $\mathbf{x}$ is defined as [11]

$$\begin{aligned} E_{\mathrm{sm}} &= E_{\mathrm{sm_h}} + E_{\mathrm{sm_v}} \\ &= \sum_{j=1}^{n}(x_{1,j} - l_{m,j})^2 + \sum_{i=1}^{m}(x_{i,1} - u_{i,n})^2, \end{aligned} \tag{4}$$

where $E_{\mathrm{sm_h}}$ and $E_{\mathrm{sm_v}}$ denote the horizontal and vertical side-match error respectively. Finally, we encode a block using the quantizer with the corresponding state codebook. Note that the generated state codebooks in both the encoder and the decoder are the same due to the same mean images and super codebooks.

C. Image Partition

The partition of an image affects the bit rate of the compressed image. For examples, the block size is 8×8 in JPEG format and usually 4×4 in ordinary VQs. In FBC techniques, we can partition an image into the range blocks with two-level sizes (8×8 parent range blocks $\mathbf{x}_8$ and 4×4 child range blocks $\mathbf{x}_4$) to compromise the bit rate and PSNR for the coded image. The quadtree partition is also efficient and flexible for different coding techniques. To simplify the comparison with the non-iterative FBC technique, we perform two types of partition for the test images: (1) single block size, 8×8 or 4×4, and (2) two-level block sizes, 8×8 (parent level) and 4×4 (child level). Nevertheless, the proposed GMFVQs and

SMFVQs can be modified to fit other partition schemes such as the quadtree partition for the test images.

The block diagram of the proposed FSFVQs in Figure 1 is only for the single block size only. For the two-level block sizes, the corresponding block diagram will require some modifications. An image is first partitioned into parent range blocks to which the same coding procedures as those in the non-iterative FBC technique are applied. The proposed GMFVQs or SMFVQs are not applied here since we try to maximize the number of parent range blocks coded to reduce the bit rate. In the parent level, however, if the distortion between the range block and the affine-transform-coded blocks is greater than the threshold value E_{th}, the parent range block is split into four child range blocks for further processing. We measure the mean and variance for each child range block and then construct the mean image at the child level. The super codebook is constructed by affine-transforming the domain blocks subsampled from this mean image. Then the child range blocks are coded by the proposed GMFVQs and SMFVQs. Note that in the proposed FSFVQs, the child range blocks at the first row and the first column are still coded by the affine transformation in the non-iterative FBC technique.

3 Experimental Results

In the computer simulation, two 512×512 images (Lena and Jetplane) shown in Figure 4(a) and 4(b) with the eight-bit grayscale resolution are used to test the proposed FSFVQs. The performance of the decoded image quality is evaluated by PSNR and the bit rate. In our simulation, an image is partitioned into range blocks of a single size, either 8×8 or 4×4; or an image is partitioned into two-level block sizes: 8×8 and 4×4. A general form for the

PSNR of the decoded image is defined as

$$\mathrm{PSNR} = 10\log_{10}\frac{255^2}{\sum_{i=1}^{N_8}\mathrm{MSE}(\mathbf{x}_{8_i},\mathbf{x}'_{8_i})+\sum_{i=1}^{N_4}\mathrm{MSE}(\mathbf{x}_{4_i},\mathbf{x}'_{4_i})} \quad \mathrm{dB}, \tag{5}$$

where N_8 and N_4 are the total numbers of the 8×8 range block $\mathbf{x}_8$ and the 4×4 range block $\mathbf{x}_4$ respectively. The distortion between the original and coded range blocks is represented by the mean-squared-error (MSE) measurement defined as

$$\mathrm{MSE}(\mathbf{x},\mathbf{x}') = \frac{1}{B^2}\sum_{0<i,j\le B}(x_{i,j}-x'_{i,j})^2, \tag{6}$$

where $B\times B$ is the block size and $x'_{i,j}$ denotes the gray level of the (i,j)th pixel in the coded range block $\mathbf{x}'$. For all the cases in our simulation, we set 25 to the threshold values E_{th} for the variance of 8×8 and 4×4 range blocks. The bit rate calculation for different partitions will be given in the following subsections. The number of domain blocks in a domain pool represents its size. There are four sizes (16, 64, 256, and 1024) of the domain pool used in non-iterative FBC in our simulation. The corresponding numbers of the codewords in the super codebooks are 1024, 4096, 16384, and 61536. On the other hand, ten sizes of state codebook are used in the proposed FSFVQs. They are 16, 32, 64, 128, 256, 512, 1024, 2048, 4096, and 8192.

A. Single Block Size

The length of the attached header I_{h} to the fractal code for each range block is only one bit (i.e., I_{h}=1) because it only denotes whether or not the range block is coded by the block mean. Therefore, the bit rate can be calculated by

$$\begin{aligned}\mathcal{B} &= \{N_\mu(I_{\mathrm{h}}+I_\mu)+N_{\mathrm{AT}}(I_{\mathrm{h}}+I_\mu+I_\alpha+I_\iota+I_{\mathrm{P_D}})\\ &\quad +N_{\mathrm{SC}}(I_{\mathrm{h}}+I_\mu+I_{\mathrm{SC}})\}/512^2 \qquad \mathrm{bit/pixel},\end{aligned} \tag{7}$$

where I_μ, I_α, I_ι, $I_{\mathrm{P_D}}$ and I_{SC} denote the required bits for the block mean, contrast scaling, isometry, the position of the domain pool, and the index of the codeword in the state

codebook respectively. In addition, N_{μ}, N_{AT}, and N_{SC} denote the number of the range blocks coded by the block mean, the number of the range blocks coded by the affine transformation, and the number of the range blocks coded by the codeword in the state codebook respectively. Note that only the range blocks located at the first row or the first column can be coded by the affine transform.

For an image partitioned into 8×8 or 4×4 range blocks, we measure the mean value and the variance of every range block and then construct a 64×64 or 128×128 mean image. If the block variance is less than the threshold value $V_{\mathrm{th}} = 25$, the range block is coded by its mean value. Otherwise, the range block is coded by the proposed FSFVQs. The domain pool is constructed with the domain blocks extracted from the mean image. The rate-PSNR comparisons of the proposed GMFVQs, SMFVQs, and non-iterative FBC techniques for the Lena image in the cases of single block size are shown in Figures 5 and 6. In non-iterative FBC techniques, we determine the coding performance with the affine transformation under the different sizes of the domain pool. The numbers shown in the figure represent the different sizes of the domain pool. On the other hand, the numbers shown in the curves for GMFVQs and SMFVQs denote the sizes of the state codebooks. As shown in both figures, the bit rates in the GMFVQs and SMFVQs are significantly reduced, and the GMFVQs save more bit rate than the SMFVQs.

B. Two-Level Block Sizes

To identify the partition state for the parent range block, we attach a variable-length header to the fractal code. Table 1 shows the headers, subheaders, and the bit allocation for parent range blocks $\mathbf{x}_8$ and child range blocks $\mathbf{x}_4$. We assign '0' as the header of the mean-coded parent range block. For the parent range block coded by the affine transformation, '10' is the header. The header '11' represents that a parent range block is split into four child

blocks. Then, the subheader '0' represents the child range block coded by the mean. Another subheader '1' represents two possible coding statuses for the child range blocks: (1) coded by the affine transformation when the block is located in the first row or the first column; (2) coded by the proposed FSFVQs when the block is located elsewhere. Therefore, the header has various lengths for different parent range blocks. In the case of two-level block sizes, the finite state algorithm is not used for parent range blocks. For a 64×64 mean image in the parent level, we can select 57×57 parent domain blocks at most. Thus 12 bits are required to denote the domain blocks. We perform the full search algorithm in the parent level so that more parent range blocks can be coded in this level to reduce the bit rate. After the coding process in the parent level is completed, the parent range blocks that do not satisfy the MSE criterion are split into four child range blocks. An 128×128 mean image is first derived from the mean values of the range blocks coded in the parent level and the child level. Then the super codebook is constructed from the child domain blocks and their affine transformations. Here the child domain blocks are subsampled from the mean image with an adjustable sampling period T.

Now, only the child range blocks are coded with the similar method shown in Subsection 3-A. The bit rate can be calculated by

$$\begin{aligned} \mathcal{B}_2 &= \{N_{8_\mu}(I_{\mathrm{h}8_\mu} + I_\mu) + N_{8_{\mathrm{AT}}}(I_{\mathrm{h}8_{\mathrm{AT}}} + I_\mu + I_\alpha + I_\iota + I_{P_D}) \\ &\quad + I_{h84}N_{84} + N_{4_\mu}(I_{\mathrm{h}4_\mu} + I_\mu) + N_{4_{\mathrm{AT}}}(I_{\mathrm{h}4_{\mathrm{AT}}} + I_\mu + I_\alpha + I_\iota + I_{P_D}) \\ &\quad + N_{4_{\mathrm{FSFVQ}}}(I_{\mathrm{h}4_{\mathrm{FSFVQ}}} + I_\mu + I_{\mathrm{SC}})\}/512^2 \quad \text{bit/pixel}, \end{aligned} \tag{8}$$

where $N_{8_\mu}, N_{8_{\mathrm{AT}}}, N_{84}$, $N_{4_{\mathrm{AT}}}$, and $N_{4_{\mathrm{FSFVQ}}}$ denote the number of the parent range blocks coded by the mean, the number of the parent rang blocks coded by affine transformation, the number of the parent range blocks partitioned into four child range blocks, the number of the child range blocks coded by the affine transformation, and the number of the child range

blocks coded by the proposed FSFVQs, respectively. The rate-PSNR comparisons of the proposed GMFVQs, SMFVQs, and non-iterative FBC techniques for the Lena and Jetplane images in the case of two-level block sizes are shown in Figure 7. We found that, under the same PSNR, the bit rates for both coded images are significantly reduced in the proposed methods. In average, GMFVQs achieve 15%–20% reduction and SMFVQs achieve 10%–12% reduction in the bit rates. Note that here the entropy coding techniques for the indices of the codewords are not employed. In GMVQs and SMVQs, the distribution of the indices is highly nonuniform [10], [11] when the block size is small, e.g. 4×4. We can expect more reduction in the bit rate with the entropy coding technique such as the Huffman coding. The reconstructed images and their corresponding error images in the case of two-level block sizes are shown in Figure 8. The error at the edges are not obvious. That is, the edge continuity in coded images is preserved very well. Note that the error images are biased at the gray level 128 and enhanced by three times. The blocking effects that appear at the uniform area can be post processed by the method proposed in Refs. [16] and [17] to enhance the visual quality. We also found that the number N_{DB} of the domain blocks selected from the child mean image (128×128) affects the PSNR performance. The number N_{DB} is determined by the sampling period T. For example, if T=4, the number N_{DB} is 128/4×128/4 = 32×32. As shown in Tables 2 and 3, if the state codebook size is less than 1024, the PSNRs of decoded Lena image for N_{DB}=32×32 are better than those for N_{DB}=63×63 in both the GMFVQs and SMFVQs. Therefore, it is not necessary to select a large number of the domain blocks from the mean image in the parent level.

4 Conclusion

In this paper, we propose two classes of FSFVQs, the GMFVQs and SMFVQs, for the image coding framework. The concepts in GMVQs and SMVQs are applied to the non-iterative FBC technique so that the number of the bits required for coding a range block is greatly reduced. In the proposed GMFVQs and SMFVQs, the codewords in the state codebooks are extracted from the super codebook so that the contiguity at block boundaries can be preserved. In our experiments, three different block partitions for the image are tested: (1) 8×8 range blocks, (2) 4×4 range blocks, and (3) 8×8 (parent) and 4×4 (child) range blocks. The simulation results show that the proposed GMFVQs and SMFVQs save about 10%–20% in the bit rate for the non-iterative FBC technique. On the other hand, the reconstructed images have the excellent quality with negligible blocking effects at edges. The only limitation of the proposed techniques is the extra computations in constructing large super codebooks and sorting the codewords to obtain state codebooks from the super codebook. In our future work, we will try to reduce the computational load and speed up the encoding process for the proposed FSFVQs. We can select the significant domain blocks with useful rules rather than regularly sample domain blocks from the mean image. Then some redundant transformed blocks can be filtered out before the sorting process in order to reduce the comparisons for sorting.

Acknowledgment

This research was partially supported by the National Science Council, Taiwan, under contract NSC 89-2213-E-324-050.

References

[1] I. Kim and R. Kim, "Still image coding based on vector quantization and fractal approximation," *IEEE Trans. On Image Processing*, vol. 5 no. 4, pp. 587–597, April 1996

[2] F. Davoine, M. Antonini, J.-M. Chassery, and M. Barlaud, "Fractal image compression based on Delaunay triangulation and vector quantization," *IEEE Transactions on Image Processing*, vol. 5 no. 2, pp.338–346, Feb. 1996

[3] R. Hamzaoui and D. Saupe, "Combining fractal image compression and vector quantization," *IEEE Trans. on Image Processing*, vol. 9, no. 2, pp. 197–208, February 2000

[4] C. Kim, R. Kim, and S. Lee, "A fractal vector quantizer for image coding," *IEEE Trans. on Image Processing*, vol. 7, no. 11, pp. 1598–1602, November 1998

[5] K. Masselos, Y. A. Karayiannis, and T. Stouraitis, "Image coding using a fractal/vector quantization model," *Proceedings of the 1997 13th International Conference on Digital Signal Processing (DSP 97)*, vol. 2, pp. 797–800, 1997

[6] K. A. Saddi, Z. Brahimi, and N. Baraka, "Hybrid approach for still image compression based on fractal approximation and vector quantization," *Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society, 1998. IECON '98*, vol. 3, pp.1487–1492, 1998

[7] Hsuan T. Chang and C. J. Kuo, "Fractal block coding using simplified finite-state algorithm," *SPIE's Symposium on Visual Communication and Image Processing'95*, vol. 2501, no. 3, pp. 536–544, Taiwan, May 1995

[8] Hsuan T. Chang and C. J. Kuo, "Finite-state fractal block coding of images," *Proceedings of 1996 IEEE International Conference on Image Processing*, vol.1, pp.133–136, 1996

[9] Hsuan T. Chang and C. J. Kuo, "Iteration-free fractal image coding based on efficient domain pool design," *IEEE Trans. on Image Processing*, vol. 9, no. 3, pp. 329–339, March 2000

[10] A. Gersho and R. Gray, *Vector quantization and signal compression*, Chapter 14, Kluwer Academic, Taipei, Taiwan 1992

[11] T. Kim, "Side match and overlap match vector quantization for images," *IEEE Trans. on Image Processing*, vol. 1, no. 2, pp. 170–185, 1992

[12] Hsuan T. Chang, "Gradient match vector quantizers for images," *Optical Engineering*, vol. 39, no. 8, pp. 2046–2057, August 2000

[13] Hsuan T. Chang and C. J. Kuo, "A novel non-iterative scheme for fractal image coding," *Journal of Information Science and Engineering*, vol. 17, pp. 429 443, April 2001

[14] T.-S. Chen and C.-C. Chang , "A new image coding algorithm using variable-rate side-match finite-state vector quantization," *IEEE Transactions on Image Processing*, Vol. 6, no. 8, pp. 1185–1187, Aug. 1997

[15] H.-C. Wei, P.-C. Tsai, and J.-S. Wang, "Three-sided side match finite-state vector quantization," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 10, no. 1, pp. 51–58, February 2000

[16] Y. Fisher, *Fractal Image Compression: Theory and Applications*, Y. Fisher, Ed. New York: Springer Verlag, January 1995

[17] Hsuan T. Chang and C. J. Kuo, "Adaptive schemes for improving fractal block coding of images," *Journal of Information Science and Engineering*, vol. 15, no. 1, pp. 11–25, 1999

Table 1: HEADER/SUBHEADER AND BIT ALLOCATION FOR THE PARENT AND THE CHILD RANGE BLOCKS.

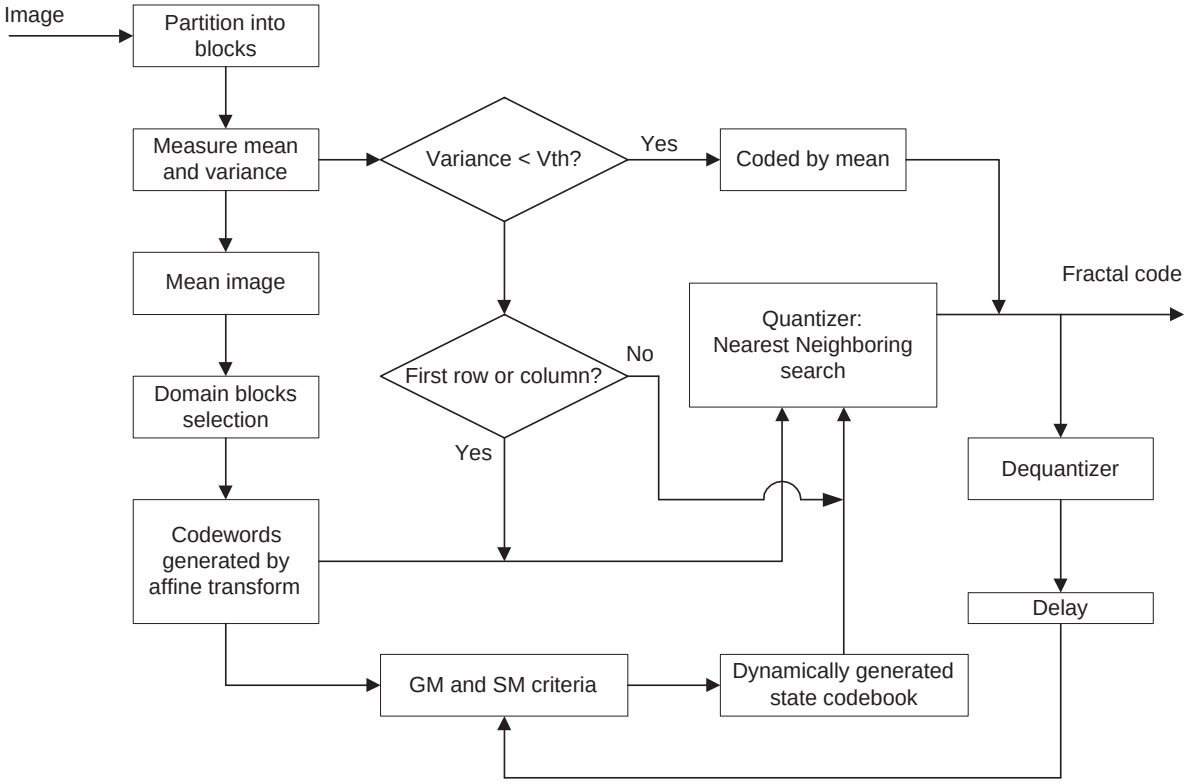
BBLOCK STATUS	HEADER/	BIT ALLOCATION				
	SUBHEADER	I_{h}	I_{ι}	$I_{\mu_{\mathbf{x}}}$	I_{α}	I_{P_D}
$\mathbf{x}_8$ CODED BY $\mu_{\mathbf{x}}$	'0'	$I_{\mathrm{h}8_{\mu}}=1$		6		
$\mathbf{x}_8$ CODED BY AT	'10'	$I_{\mathrm{h}8_{\mathrm{AT}}}=2$	3	6	3	12
$\mathbf{x}_8$ SPLIT INTO FOUR $\mathbf{x}_4$	'11'	$I_{\mathrm{h}84}=2$				
$\mathbf{x}_4$ CODED BY $\mu_{\mathbf{x}}$	'0'	$I_{\mathrm{h}4_{\mu}}=1$		6		
$\mathbf{x}_4$ CODED BY AT	'1'	$I_{\mathrm{h}4_{\mathrm{AT}}}=1$	3	6	3	12
$\mathbf{x}_4$ CODED BY FSFVQ	'1'	$I_{\mathrm{h}4_{\mathrm{FSFVQ}}}=1$		$6+I_{SC}$		

Table 2: THE PSNR (IN DECIBELS) COMPARISON FOR THE CHILD-LEVEL STATE CODEBOOKS THAT USE DIFFERENT NUMBERS OF THE SAMPLED DOMAIN BLOCKS, N_{DB}, IN GMFVQs.

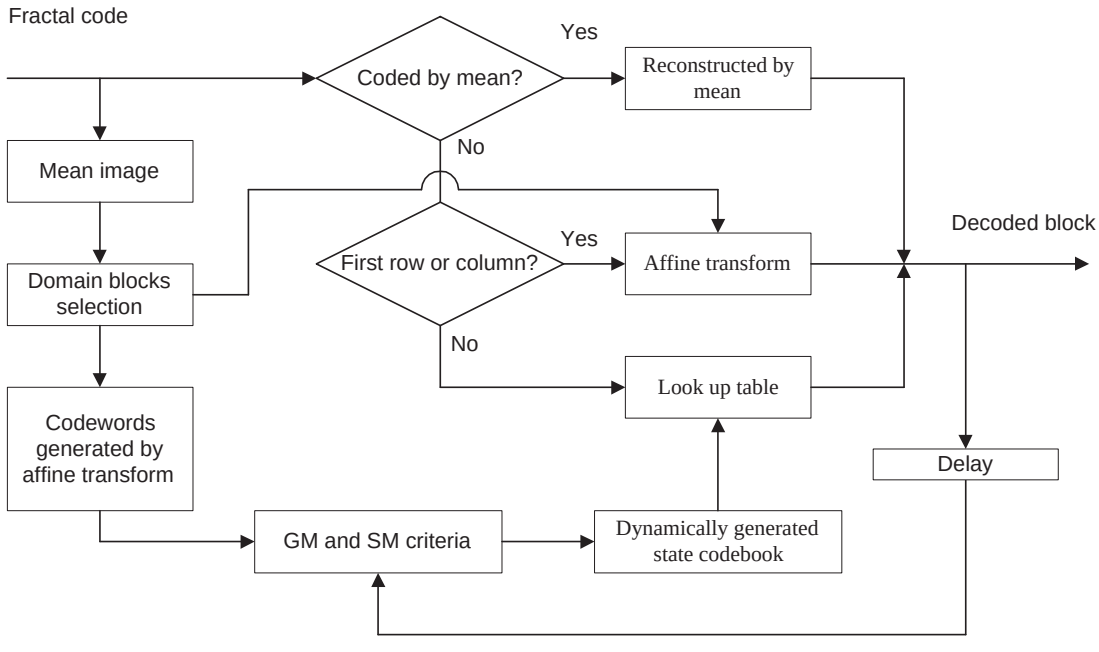
SC SIZE	16	32	64	128	256	512	1024	2048	4096	8192
BIT RATE (bpp)	0.330	0.343	0.356	0.369	0.382	0.395	0.408	0.421	0.434	0.457
N_{DB}=16×16	30.60	31.47	32.22	32.70	33.15	33.51	33.68	33.79	33.88	34.03
N_{DB}=32×32	30.30	31.29	31.97	32.51	33.08	33.54	33.96	34.18	34.36	34.47
N_{DB}=63×63	29.27	30.47	31.49	32.21	32.83	33.39	33.81	34.23	34.49	34.71

Table 3: THE PSNR (IN DECIBELS) COMPARISON FOR THE CHILD-LEVEL STATE CODEBOOKS THAT USE DIFFERENT NUMBERS OF THE SAMPLED DOMAIN BLOCKS, N_{DB}, IN SMFVQs.

SC SIZE	16	32	64	128	256	512	1024	2048	4096	8192
BIT RATE (bpp)	0.346	0.363	0.379	0.395	0.412	0.429	0.445	0.462	0.478	0.494
N_{DB}=16×16	29.66	30.54	31.17	31.66	32.13	32.45	32.70	32.89	32.97	33.08
N_{DB}=32×32	28.90	29.87	30.71	31.55	32.30	32.98	33.50	33.91	34.19	34.39
N_{DB}=63×63	28.16	29.21	30.05	30.82	31.62	32.41	33.08	33.70	34.14	34.50



(a) Encoder



(b) Decoder

Figure 1: THE BLOCK DIAGRAM OF THE PROPOSED FSFVQs.

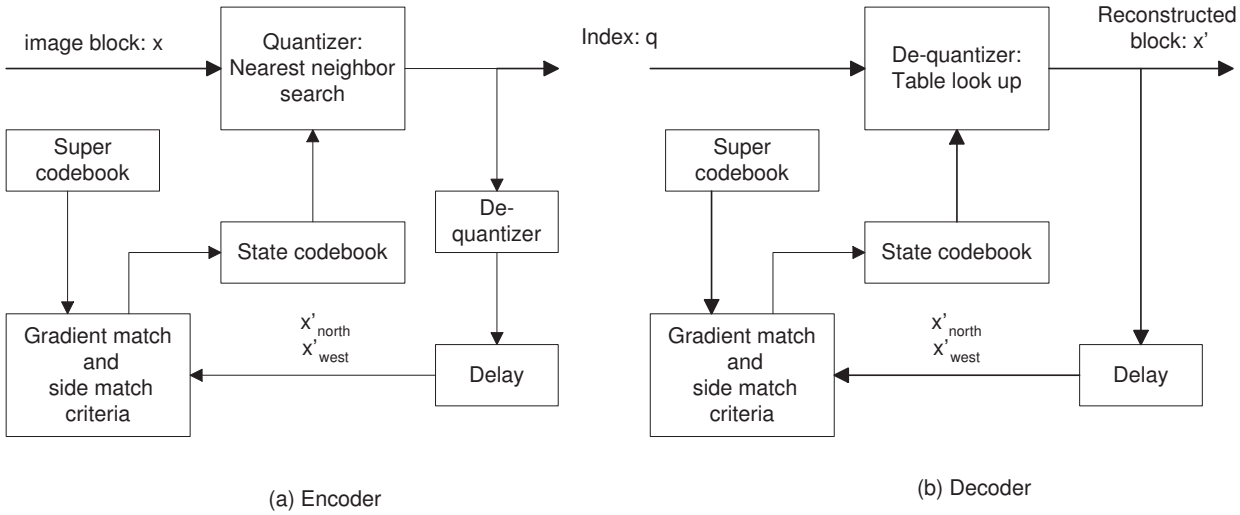


Figure 2: THE BLOCK DIAGRAM OF SMVQs AND GMVQs: (a) THE ENCODER AND (b) THE DECODER.

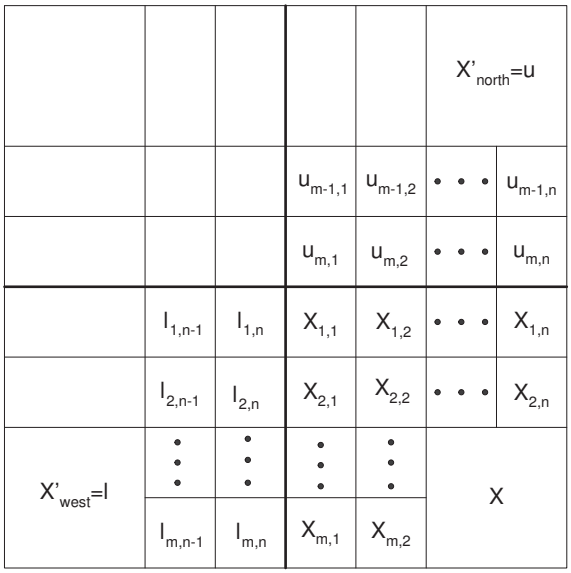


Figure 3: THE PIXELS THAT CONTRIBUTE TO THE GENERATION OF THE STATE CODEBOOK IN SMVQs AND GMVQs.

(a) (b)

Figure 4: THE TEST IMAGES: (a) LENA AND (b) JETPLANE.

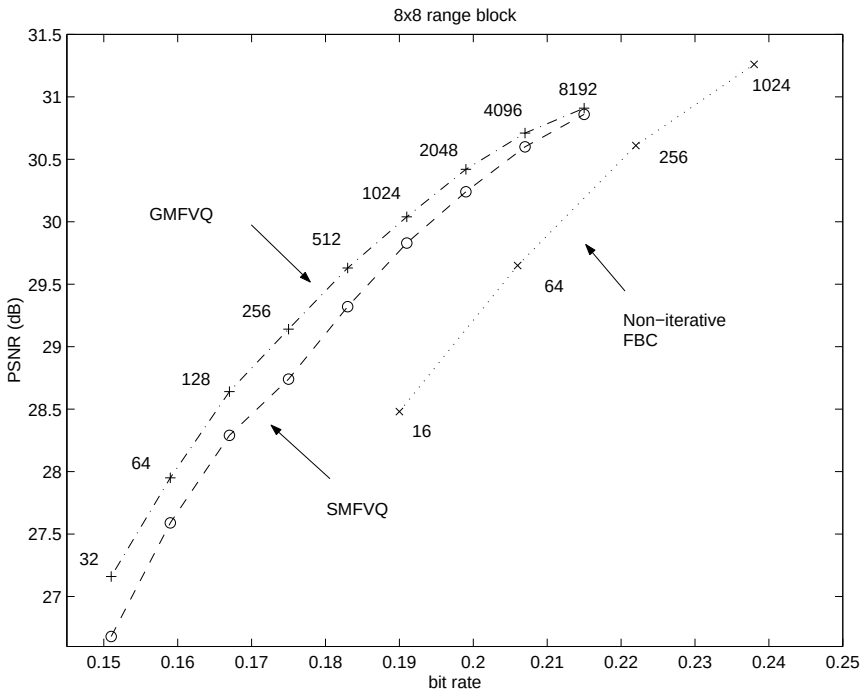


Figure 5: THE RATE-PSNR COMPARISON OF THE PROPOSED SMFVQs, GMFVQs AND NON-ITERATIVE FBC TECHNIQUES USING BLOCK SIZE 8×8.

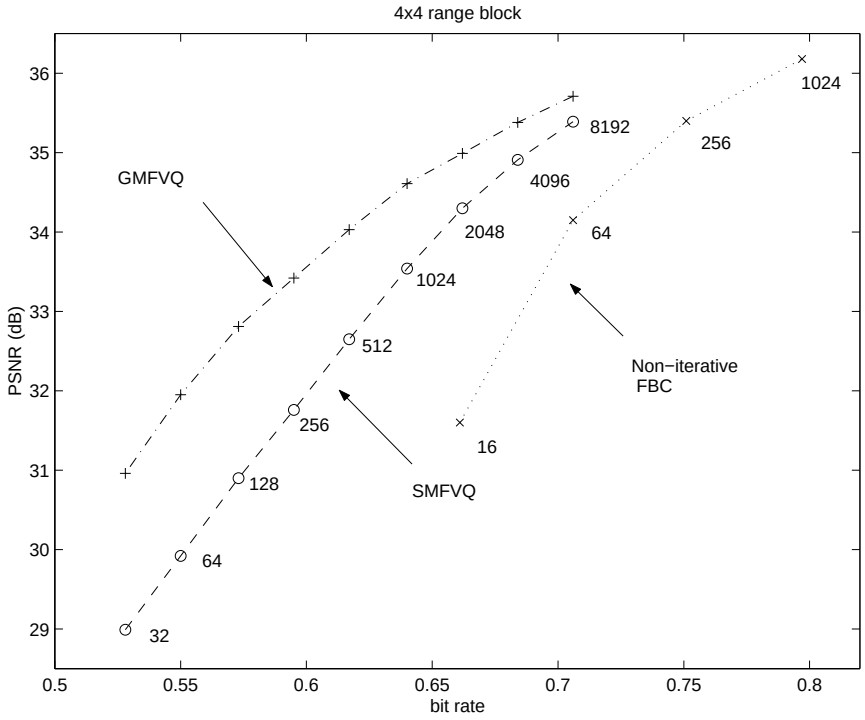


Figure 6: THE RATE-PSNR COMPARISON OF THE PROPOSED SMFVQs, GMFVQs AND NON-ITERATIVE FBC TECHNIQUES USING BLOCK SIZE 4×4.

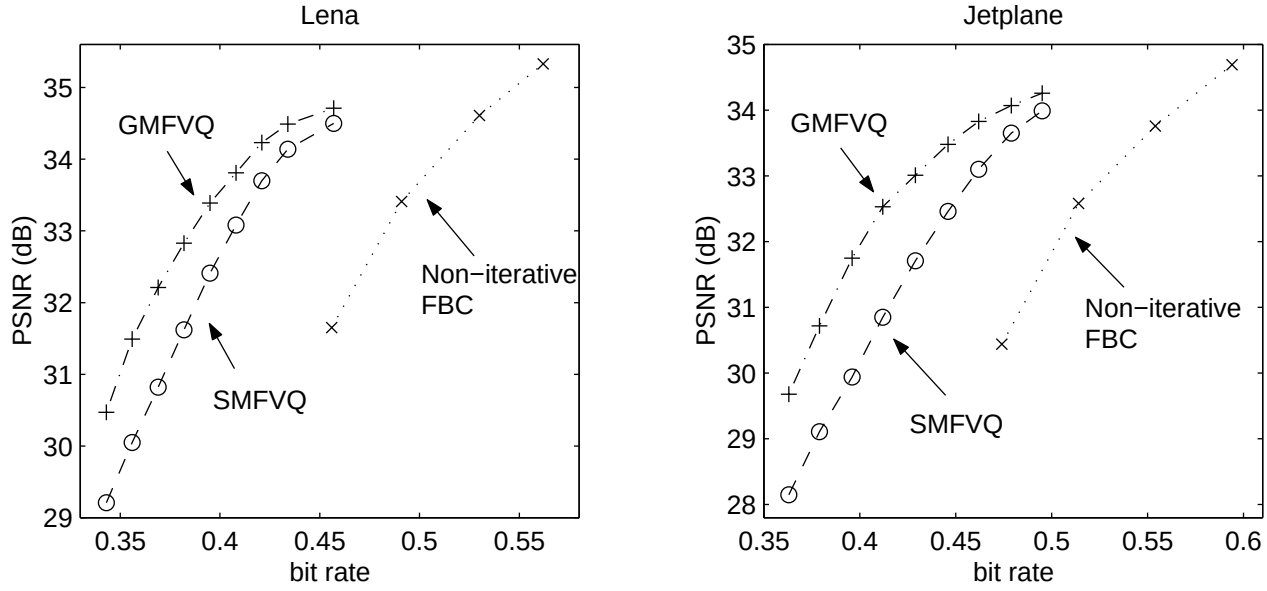


Figure 7: THE RATE-PSNR COMPARISON OF THE PROPOSED SMFVQs, GMFVQs AND NON-ITERATIVE FBC TECHNIQUES USING TWO-LEVEL BLOCK SIZES 4×4 AND 8×8: (a) LENA, (b) JETPLANE.

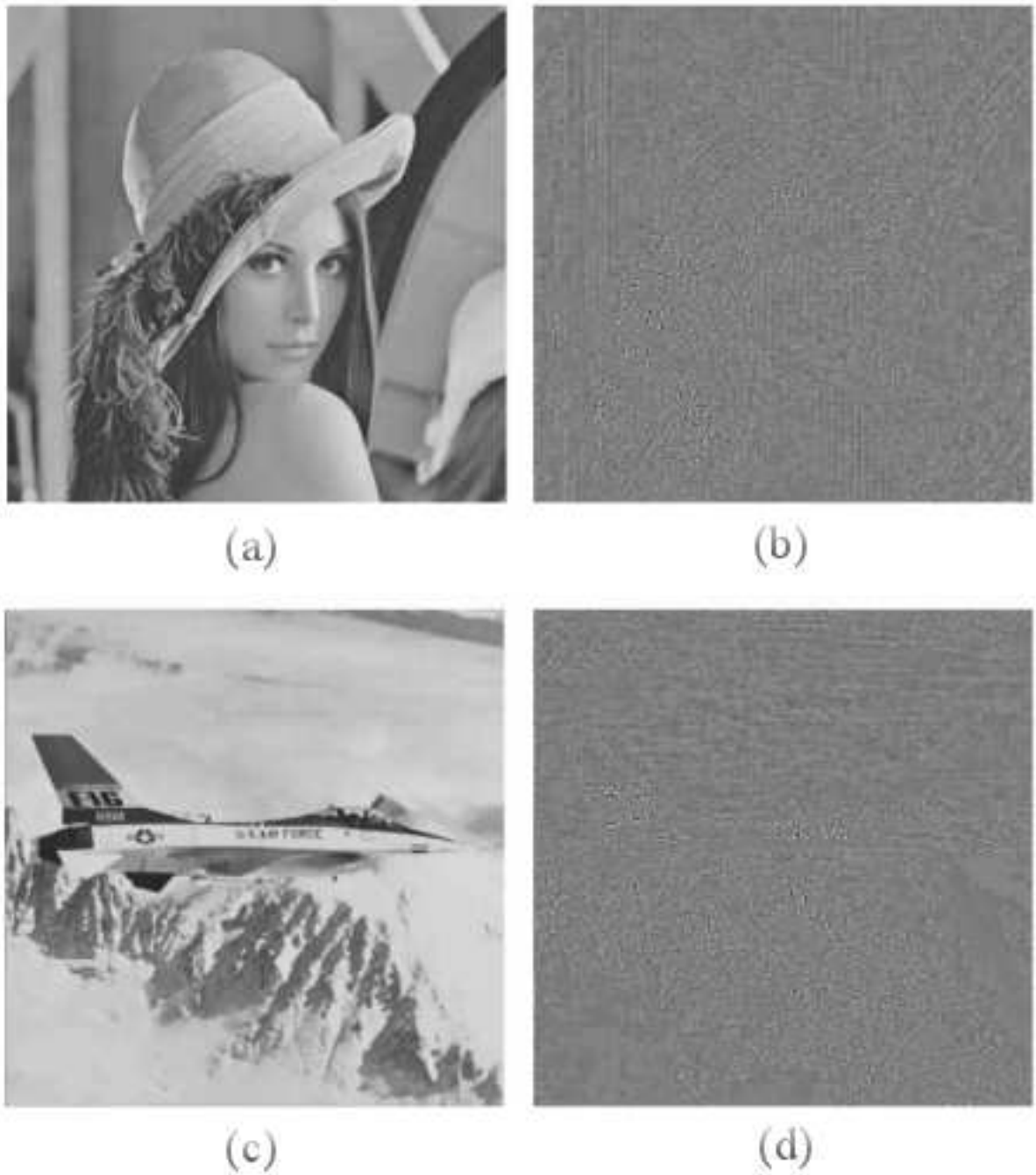


Figure 8: THE RECONSTRUCTED IMAGES AND THE CORRESPONDING ENHANCED ERROR IMAGES IN THE CASE OF TWO-LEVEL BLOCK SIZES: (a) LENA, 34.5 dB AT 0.407 BIT/PIXEL, (b) THE ENHANCED ERROR IMAGE OF (a); (c) JETPLANE, 34.07 dB AT 0.465 BIT/PIXEL, (d) THE ENHANCED ERROR IMAGE OF (c).