International Journal of Systems Science
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/tsys20

Adaptive self-quantization in wavelet-based fractal image compression

Bing-Fei Wu , Yi-Qiang Hu & Hung-Hseng Hsu
Published online: 26 Nov 2010.

To cite this article: Bing-Fei Wu , Yi-Qiang Hu & Hung-Hseng Hsu (1999) Adaptive self-quantization in wavelet-based fractal image compression, International Journal of Systems Science, 30:5, 541-549, DOI: 10.1080/002077299292281

To link to this article: http://dx.doi.org/10.1080/002077299292281

International Journal of Systems Science, 1999, volume 30, number 5, pages 541–549

Adaptive self-quantization in wavelet-based fractal image compression

Bing-Fei Wu†, Yi-Qiang Hu† and Hung-Hseng Hsu†

Finding a model to quantize the scale factors in wavelet-based fractal image compression is a complicated issue. To avoid error, it is helpful to model the distribution of the scale factors and quantize them before the computation process by iterated function systems. Traditionally, a fixed model with uniform distribution was frequently adopted. This is not sophisticated enough, however, to quantize these scale factors from errors since, in general, the factors are not uniformly distributed. We propose an adaptive algorithm with self-quantization to overcome this drawback. Except for the functions of adaptation and self-quantification, the approach has the optimal property that the fundamental objective is to reduce the quantization errors.

1. Introduction

In the application of image processing, discrete wavelet transforms (DWT) are employed to extract the coded image into several sub-images with different resolutions (Mallat 1989, Antonini *et al.* 1992, Strang and Nguyen 1996). The methods of image compression using DWT could provide high compression ratios (CR) and high image fidelity as well (Antonini *et al.* 1992, Vetterli and Kovačević 1995, Hsu *et al.* 1997). These subimages, except for the ones with lowest frequencies, which are similar to the original image in general, are called detailed images. Rinaldo and Calvagno (1995) further pointed out the redundancy of the wavelet-based images across scales from the perspective of fractals. Hence, it is said that an image performed after DWT has the intrinsic property of fractals and can be manipulated by fractal compression methods for further compressing this image effectively. Moreover, the application of fractal coders to wavelet-based images over original images is highly recommended for the application's ability to enhance the similarity among the detailed sub-images, but also to reduce the blocking effect in most other block-based coding techniques since we extract the similarities in the frequency domain instead of the spatial domain.

Received 20 October 1997. Revised 28 May 1998. Accepted 29 May 1998.

† Department of Electrical and Control Engineering, National Chiao Tung University, Hsinchu, Taiwan; e-mail: bwu@cc.nctu.edu.tw

A block coding method by means of the technique of Interactive Function Systems (IFS), introduced by Barnsley and Jacquin (1988) and Jacquin (1993), was widely used in fractal image compression. It has proven successful for compressing images at low bit rates. The main procedure of IFS is to find and manipulate the domain block such that it can be the most matched one for a fixed range block in some measure senses (Jacobs *et al.* 1992, Fisher 1994). Following that, we have to determine all the IFS parameters corresponding to the range blocks. The IFS maps can be iteratively applied to find the fixed point, which is an approximation to the image to be coded. To reduce the decoding time and error, this study presents a new prediction approach that differs slightly from IFS. In the coding part of the predictor, the domain and range pools belong to different subbands with the same orientation and block size. The domain pool is constituted by the subimage with a lower frequency, thereby ensuring smaller values of scale factors since the image has the property of power 'decayness' (Antonini *et al.* 1992).

The IFS parameters include the positions of the domain blocks, eight isometries of the square achieved from the compositions of reflections and 90° rotations, scale factors and offset values. There is no doubt that the positions and isometries belong to integers and need not be quantized before coding them. However, generally, the scale factors and offset values should be floating numbers. We need to quantize them for coding. The main purpose of this paper will focus on the relationship of the quantization and IFS coding error. We try to find

a better way to avoid errors accumulating, under the situation that IFS coding is followed by quantization. A complete code including DWT, IFS and the entropy coding is addressed in our other work (Wu *et al.* 1997). The offset values here allow us to adjust the DC components of the range blocks (Antonini *et al.* 1992). More precisely, the offset values in IFS coding can be taken as the DC values corresponding to range blocks (Davis 1995). Naturally, we assume that the distribution of the offset values for all range blocks is similar to that of the encoded image. For the images performed after DWT, the distributions of detailed images can be modelled on generalized Gaussian distributions (Daubechies 1988). Hence, the distribution of the offset values was obtained and their quantization was found. Next, we will face a serious problem when quantizing these scale factors.

If we quantize the floating scale factors after IFS, the 'second error' will be introduced. That is the quantization error and the error induced by IFS would be compounded. A better way is to quantize these floating numbers before processing in IFS, provided that we know and model the histogram of these scale factors precisely. In fact, we have no idea how to model the histogram, other than the fact that the distribution of these scale factors was concentrated around zero and was decayed on both sides, and was observed from the experimental results. Therefore the simplest way is to approximate it with a uniform distribution (Jacquin 1993). This would result in a less optimal case for searching for IFS parameters. In this case we propose an Adaptive Self-Quantizer (ASQ) to overcome the above disadvantage in quantizing these scale factors. As we know, adaptivity will be a trend for lossless data compression in the future. It will do well in the case that the statistics of the input source are either unknown *a priori* or varying over time. Some recent work of adaptive quantization (Steinberg and Gutman 1993, Chan and Vetterli 1995) was designed with existing codebooks, which require background information. The objective of altering the support region of a uniform scalar quantization is also demonstrated in Jayant (1973), Crisafulli and Bitmead (1993); however, the modelling source is not adjustable. ASQ is initiated as a uniform model intuitively and is permitted to update the model adaptively according to the input values. Based on the adaptive model and inputs, the quantized outputs can also be self-adjusted. This means that no more bits are necessary to specify the codebook. After that, we acquire an effective model and reduce the error in quantization of the scale factors. The concept of optimality is included in this algorithm to reduce quantization error. There are two approaches to reconstruct the coding data, the forward and backward methods, which depend on the necessary encoded bits and the quality of the reconstructed image. Moreover, we apply the concept of the source modelling from Ortega and Vetterli (1997), which estimates the probability density function (p.d.f.) at the mid-point of the intervals by taking a matrix inversion through Gaussian substitution methods. However, we have estimated the p.d.f. at the values of decision levels by a kind of curve fitting technology, Least Squared Error (LSE) method. Next, we also depict the distribution by linear interpolation. In addition, the approach can be considered as one kind of adaptive filters (Widrow and Stearns 1985). We can find the geometric ratio of the algorithm from the viewpoint of adaptive signal processing and consider the influence of the ratio to the adaptation capability and reconstruction fidelity.

The organization of this paper is as follows: the problem of quantizing scale factors in fractal image compression is formulated in the next section. In section 3, we will introduce the algorithm of the adaptive self-quantizer, which is designed by three key points. The reconstruction procedures are also illustrated. After that, we will address some intrinsic properties about the algorithm. The experimental results, as listed in section 4, reveal that the adaptive self-quantizer is superior to the uniform model that was usually adopted before. As a result, a brief conclusion is made in the final section.

2. Problem formulations and notations

We will put more emphasis on the quantization of the scale factors, for it plays an important role in fractal block coding. In order to quantize the scale factors effectively, more precisely, and to reduce the composite errors—which are produced by fractal coding errors and quantization errors—we shall introduce ASQ before applying the IFS coding.

2.1. *Problem formulations*

Let us formulate the quantization problem first. Consider a d-bit scalar quantizer, then there are 2^d output levels corresponding to 2^d input intervals. The I/O relationship of a 2-bit scalar quantizer is shown in figure 1.

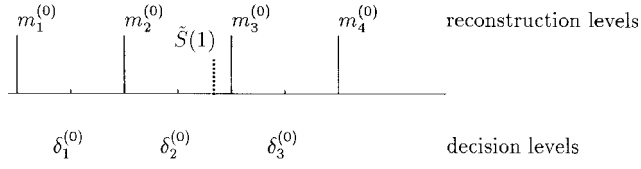

Figure 1. A 2-bit scalar quantizer.

2.2. *Notation*

For convenience in depicting the ASQ algorithm, we should define the notations of the d-bit scalar quantizer.

- d: bits of the quantizer;
- n_f: the number of scale factors;
- $S(n)$: the nth scaling factor, for $n \in N \triangleq \{1, 2, \ldots, n_f\}$;
- $\tilde{S}(n)$: the estimated input;
- $\delta_j^{(n)}$: decision levels at stage n of the quantizer, $j \in Q - \{2^d\}$, where $Q \triangleq \{1, \ldots, 2^d\}$;

$$y_i^{(n)} \equiv \begin{cases} [\delta_{i-1}^{(n)}, \delta_i^{(n)}], & \text{for } i \in Q - \{1, 2^d\}, \\ (-\infty, \delta_1^{(n)}], & \text{for } i = 1, \\ [\delta_{2^d-1}^{(n)}, \infty), & \text{for } i = 2^d, \end{cases}$$

which are called the ith interval of the quantizer;

- $X_i^{(n)}$: the number of elements in $y_i^{(n)}, i \in Q$. It is also called the ith counter. Moreover, $X_i^{(n)}/n$ can be considered as the probability in $y_i^{(n)}$;
- $m_i^{(n)}$: reconstructed levels at stage n of the quantizer, $i \in Q$.

3. Adaptive Self-Quantizers

In fractal block coding, the uniform models were usually adopted to quantize the scale factors since we cannot obtain their distribution in advance. Unfortunately, in general, the scale factors would not be uniformly distributed. Hence, we have the idea of designing an adaptive quantizer that can self-adjust the model to an appropriate one. The key points of ASQ are in the following statements.

(1) The distribution of these scale factors without quantization concentrates around zero and decays on both sides in general. That is, the distribution is a model with smoothly decaying tails (Ortega and Vetterli 1997). When applying fractal coding to a detailed subimage, the concentrating histogram phenomenon can be seen in figure 2.

(2) The ASQ satisfies the optimal solution in a scalar quantizer (Max 1960, Hang and Woods 1995, Wu and Hsu 1996). That is,

   1. the reconstruction level is the centroid of the interval, and
   2. the decision level is the average of neighbouring reconstruction levels.

(3) The more concentrated the histogram, the smaller the interval.

Before illustrating the ASQ algorithm, we need to make an initialization.

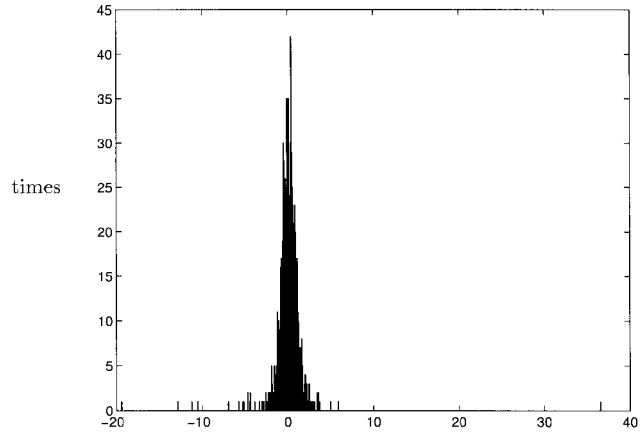

Figure 2. The histogram of scale factors without quantization in D_{h2}: the horizontal oriented subimage with resolution $\frac{1}{4}$ performed after DWT.

- $\delta_j^{(0)}$: the values to decide, such that we can uniformly split into 2^d intervals for some reasonable range R, $j \in Q - \{2^d\}$;
- $\delta_0^{(0)}, \delta_{2^d}^{(0)}$: the values used to define the dynamic range R of ASQ;
- $m_i^{(0)}$: the centroids are initially chosen to be the mid-point of the ith interval;
- $X_i^{(0)}$: the counters are all set to unity;

where $j \in Q - 2^d$ and $i \in Q$..

After initializing the algorithm, a uniform model follows. In other words, we suppose that ASQ starts running from a uniform model.

3.1. *The ASQ Algorithm*

We depict the ASQ algorithm as follows (see equations (1)–(9) at top of next page).

The ASQ algorithm will stop when the input sequence $S(n)$ is terminated ($n = n_f$). The coding data include the integer terms: $\hat{S}(n) = i$, for $n = 1, 2, \ldots, n_f$, and, if necessary, the floating terms: $m_j^{(n_f)}$, for $j \in Q$.

Remarks: The choice of a and b ($a < b$) in the ASQ algorithm depends on the first key point shown before such that the estimated input $\tilde{S}(n)$ can be located at the position closer to the origin. For example, we set $a = 1$ and $b = 2$. It is observed that the choice of a and b will affect the quantization error. Detailed discussion for different values of a/b will be illustrated in the next section. We will estimate the shape of the source distribution linearly by the parameters a and b.

3.2. *Reconstruction of ASQ algorithms*

There are two approaches to reconstruct the coding data, which depend on the necessary encoded bits and

for $S(n) \in y_i^{(n-1)}$,

$$\tilde{S}(n) = \begin{cases} \dfrac{a\delta_{i-1}^{(n-1)} + b\delta_i^{(n-1)}}{a+b} = \dfrac{am_{i-1}^{(n-1)} + (a+b)m_i^{(n-1)} + bm_{i+1}^{(n-1)}}{2(a+b)}, i \in Q_1 \triangleq \{1, \ldots, 2^{d-1}\} \\ \dfrac{b\delta_{i-1}^{(n-1)} + a\delta_i^{(n-1)}}{a+b} = \dfrac{bm_{i-1}^{(n-1)} + (a+b)m_i^{(n-1)} + a_{i+1}^{(n-1)}}{2(a+b)}, i \in Q_2 \triangleq \{2^{d-1}+1, \ldots, 2^d\}, \end{cases} \quad (1)$$

$$X_j^{(n)} = \begin{cases} X_i^{(n-1)} + 1, & \text{for } j = i, \\ X_j^{(n-1)}, & \text{for } j \neq i, \end{cases} \quad (2)$$

$$m_i^{(n)} = \frac{m_i^{(n-1)} X_i^{(n-1)} + \tilde{S}(n)}{X_i^{(n)}}, \quad (3)$$

Case I. $i \in Q_1$

$$m_j^{(n)} = m_j^{(n-1)}, \text{ for } j > i, \quad (4)$$

$$m_j^{(n)} = m_j^{(n-1)} + m_i^{(n)} - m_i^{(n-1)}, \text{ for } j < i, \quad (5)$$

$$\left.\begin{array}{l} \delta_i^{(n)} = \dfrac{m_i^{(n)} + m_{i+1}^{(n)}}{2} \\ \delta_j^{(n)} = \delta_j^{(n-1)}, \text{ for } j > i, \\ \delta_j^{(n)} = \delta_j^{(n-1)} + m_i^{(n)} - m_i^{(n-1)} = \frac{m_j^{(n)} + m_{i+1}^{(n)}}{2}, \text{ for } j < i, \end{array}\right\} \delta_k^{(n)} = \frac{m_k^{(n)} + m_{k+1}^{(n)}}{2}, \text{for } k \in Q_1. \quad (6)$$

Case II. $i \in Q_2$

$$m_j^{(n)} = m_j^{(n-1)}, \text{ for } j < i, \quad (7)$$

$$m_j^{(n)} = m_j^{(n-1)} + m_i^{(n)} - m_i^{(n-1)}, \text{ for } j > i, \quad (8)$$

$$\left.\begin{array}{l} \delta_i^{(n)} = \dfrac{m_i^{(n)} + m_{i+1}^{(n)}}{2} \\ \delta_j^{(n)} = \delta_j^{(n-1)}, \text{ for } j < i-1, \\ \delta_j^{(n)} = \delta_j^{(n-1)} + m_i^{(n)} - m_i^{(n-1)} = \dfrac{m_j^{(n)} + m_{j+1}^{(n)}}{2}, \ j, \ldots > i-1, \end{array}\right\} \delta_k^{(n)} = \frac{m_k^{(n)} + m_{k+1}^{(n)}}{2}, \text{for } k \in Q_2. \quad (9)$$

the quality of the reconstructed image. One approach is called the *forward method*. The meaning of 'forward' is that the reconstructed procedure follows an ASQ algorithm directly. That is, we repeat the ASQ algorithm from $n = 1$ to n_f by means of $\hat{S}(n)$, for $n = 1, 2, \ldots, n_f$ and the default initialized model (R is known). Another approach is called the *backward method*. 'Backward' represents that the fact we can rebuild the quantized values from $n = n_f$ to 1 by the use of the coding data $\hat{S}(n)$ and $m_j^{(n_f)}$, for $n = 1, 2, \ldots, n_f$ and $j \in Q$.

The differences between these two methods are as follows.

- The forward method only needs to store the integer term $\hat{S}(n)$ of the coding data. No more bits are required to quantize the scale factors. However, the initial model must be fixed and considered as a default model in advance.
- The backward method must save all coding data including integer and floating terms. We can reduce the quantization error at the cost of having to assign more bits to code these floating terms. However, it is unnecessary to be concerned with the initial model. But we can try to look for better solutions for different initial models, or different values of R.

3.2.1. *The forward method.* According to the default initial model and $\hat{S}(n)$, the ASQ algorithm repeats itself. For each stage n, we pick out the reconstruction level $m_i^{(n)}$, where $i = \hat{S}(n)$. During this time we can obtain the reconstructed values of scale factors for $n = 1, 2, \ldots, n_f$.

3.2.2. *The backward method.* In order to run the backward method, we need to have the coding data which include the integer terms: $\hat{S}(n) = i$, for $n = 1, 2, \ldots, n_f$, and the floating terms: $m_j^{(n_f)}$, for $j \in Q$. The course of the backward method is listed as follows:

(1) constitute $X_i^{(n_f)}$, $i \in Q$, by means of $\hat{S}(n)$, for $n = 1, 2, \ldots, n_f$;

(2) rebuild $X_i^{(n-1)}$:

$$X_i^{(n-1)} = \begin{cases} X_i^{(n)} - 1, & i = \hat{S}(n), \\ X_i^{(n)}, & \text{otherwise;} \end{cases}$$

(3) reconstruct $m_i^{(n-1)}$:

(i) $i \in Q_1$:

$$m_j^{(n-1)} = m_j^{(n)}, \quad \text{for } j > i,$$

$$m_i^{(n-1)} = \frac{[2(a+b)X_i^{(n)} + a]m_i^{(n)} - am_{i-1}^{(n)} - bm_{i+1}^{(n)}}{2(a+b)X_i^{(n-1)} + 2a + b},$$

$$m_j^{(n-1)} = m_j^{(n)} - [m_i^{(n)} - m_i^{(n-1)}], \quad \text{for } j < i;$$

(ii) $i \in Q_2$:

$$m_j^{(n-1)} = m_j^{(n)}, \quad \text{for } j < i,$$

$$m_i^{(n-1)} = \frac{[2(a+b)X_i^{(n)} + a]m_i^{(n)} - bm_{i-1}^{(n)} - am_{i+1}^{(n)}}{2(a+b)X_i^{(n-1)} + 2a + b},$$

$$m_j^{(n-1)} = m_j^{(n)} - [m_i^{(n)} - m_i^{(n-1)}], \quad \text{for } j > i.$$

For each stage n, we choose the reconstruction level $m_i^{(n)}$, where $i = \hat{S}(n)$. Therefore we can obtain the reconstructed values of scale factors for $n = 1, 2, \ldots, n_f$.

3.3. *Another version of ASQ algorithms*

In the previous subsection, we estimated the decaying property of the distribution from the choices a and b. That is, we must decide the values of a and b before running the ASQ algorithm. Therefore, the performances of ASQ algorithms depend heavily on the selection of a and b. But there seems to be a lack of robustness. In order to increase the robustness, the idea of estimating the decaying distribution by fixed values of a and b has to be replaced by another different concept: that is, to estimate the p.d.f., which is taken as the values at decision levels by the LSE method. The concept of Ortega and Vetterli (1997) estimates the p.d.f. at the mid-points of intervals and therefore obtains the solution by taking a matrix inversion through the Gaussian substitution methods. After determining the estimated p.d.f., $\hat{f}(\delta_j^{(n)})$, $j \in Q - \{2^d\}$, we can rebuild the source distribution more successfully by interpolating the estimated p.d.f. linearly. So, the value of $\tilde{S}(n)$ in (1) is substituted by the centroid of the ith interval in stage n for more robustness.

Let us describe this course in detail. In defining the accumulated probability at the ith interval in stage n as $P_i^{(n)}$ and

$$\begin{aligned} P_i^{(n)} &= \frac{X_i^{(n)}}{n + \sum_{k \in Q} X_k^{(0)}}, \\ &= \int_{\delta_{i-1}}^{\delta_i} \hat{f}(x)\,dx, \\ &= \frac{1}{2}(\delta_i - \delta_{i-1})\{\hat{f}(\delta_i) + \hat{f}(\delta_{i-1})\}, \end{aligned} \tag{10}$$

we assume that the p.d.f. at boundary points are set to zeros, i.e., $\hat{f}(\delta_0^{(n)}) = \hat{f}(\delta_{2^d}^{(n)}) = 0$. Hence, (10), for $i \in Q$, can be represented as a matrix form (see at bottom of page).

We abbreviate the matrix equation as $\Delta F = 2\bar{P}$, where $\Delta^{\mathrm{T}} \triangleq [\phi(1)\ \phi(2) \cdots \phi(2^d)]$, $\phi(\cdot)$ is a $(2^d - 1) \times 1$ column vector, $F \triangleq [\hat{f}(\delta_1^{(n)})\, \hat{f}(\delta_2^{(n)}) \cdots \hat{f}(\delta_{2^d-1}^{(n)})]^{\mathrm{T}}$ and $Y \triangleq [P_1^{(n)}\ P_2^{(n)} \cdots P_{2^d}^{(n)}]^{\mathrm{T}}$. This is inconsistent, generally, since there are $2^d - 1$ unknown parameters and 2^d equa-

$$\frac{1}{2}\begin{bmatrix} \delta_1^{(n)} - \delta_0^{(n)} & 0 & 0 & 0 & \cdots & 0 & 0 & 0 \\ \delta_2^{(n)} - \delta_1^{(n)} & \delta_2^{(n)} - \delta_1^{(n)} & 0 & 0 & \cdots & 0 & 0 & 0 \\ 0 & \delta_3^{(n)} - \delta_2^{(n)} & \delta_3^{(n)} - \delta_2^{(n)} & 0 & \cdots & 0 & 0 & 0 \\ \vdots & & & & \ddots & & \vdots & \\ 0 & 0 & 0 & 0 & \cdots & 0 & \delta_{2^d-1}^{(n)} - \delta_{2^d-2}^{(n)} & \delta_{2^d-1}^{(n)} - \delta_{2^d-2}^{(n)} \\ 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \delta_{2^d}^{(n)} - \delta_{2^d-1}^{(n)} \end{bmatrix} \begin{bmatrix} \hat{f}(\delta_1^{(n)}) \\ \hat{f}(\delta_2^{(n)}) \\ \hat{f}(\delta_3^{(n)}) \\ \vdots \\ \hat{f}(\delta_{2^d-2}^{(n)}) \\ \hat{f}(\delta_{2^d-1}^{(n)}) \end{bmatrix} = \begin{bmatrix} P_1^{(n)} \\ P_2^{(n)} \\ P_3^{(n)} \\ \vdots \\ P_{2^d-1}^{(n)} \\ P_{2^d}^{(n)} \end{bmatrix}.$$

tions in this matrix equality. Therefore, we will take the LSE solution

$$F_{\mathrm{LSE}} = (\Delta^{\mathrm{T}}\Delta)^{-1}\Delta^{\mathrm{T}}(2\bar{P}), \tag{11}$$

where $F_{\mathrm{LSE}} \triangleq \left[f_{\mathrm{LSE}}(\delta_1)\cdots f_{\mathrm{LSE}}(\delta_{2^d-1})\right]^{\mathrm{T}}$. It is a time-consuming task to obtain the offline solution in (11). The least squared solution can be acquired by a recursive form as follows (Goodwin and Sin 1984):

$$P_F(k) = P_F(k-1) - \frac{P_F(k-1)\phi(k)\phi^{\mathrm{T}}(k)P_F(k-1)}{1+\phi^{\mathrm{T}}(k)P_F(k-1)\phi(k)}, \tag{12}$$

$$F(k) = F(k-1) + P_F(k)\phi(k)\{P_k - \phi^{\mathrm{T}}(k)F(k-1)\}, \tag{13}$$

where $P_F(k)$ is the uncertainty matrix of the parameter vector $F(k)$. We initiate the values of $F(0)$ and $P_F(0)$ as a zero vector and an identity matrix with very large values, say $10^5 I$, respectively. The reason for setting very large values to $P_F(0)$ is to reduce the influence of $F(0)$, i.e. it reveals that the uncertainty of $F(0)$ is very high. After 2^d recursive steps, the value of F_{LSE} is obtained, i.e. $F_{\mathrm{LSE}} = F(k)|_{k=2^d}$.

According to the estimated p.d.f., F_{LSE}, we can reconstruct the distribution by linear interpolation between two decision levels. Moreover, we consider the estimated input, $\tilde{S}$, as the ith centroid, which is calculated by the estimated distribution,

$$\tilde{S} = \frac{\int_{\delta_{i-1}}^{\delta_i} xf(x)dx}{\int_{\delta_{i-1}}^{\delta_i} f(x)dx},$$

where

$$f(x) = \frac{(\delta_i - x)f_{\mathrm{LSE}}(\delta_{i-1}) + (x-\delta_{i-1})f_{\mathrm{LSE}}(\delta_i)}{\delta_i - \delta_{i-1}},$$

which is obtained from the linear interpolation. Therefore,

$$\tilde{S} = \frac{(\delta_i + 2\delta_{i-1})f_{\mathrm{LSE}}(\delta_{i-1})(2\delta_i + \delta_{i-1})f_{\mathrm{LSE}}(\delta_i)}{3\left[f_{\mathrm{LSE}}(\delta_i) + f_{\mathrm{LSE}}(\delta_{i-1})\right]}. \tag{14}$$

The value of $\tilde{S}$ in (1) is replaced by the closed form in Eq.(14) and the remainder of the algorithm is the same as that of the previous version.

Remarks: The reconstruction of this version ASQ follows that of the forward method mentioned above. □

3.4. *Discussions*

In this subsection, we will make some comments on the ASQ algorithm. From an insight into these intrinsic properties, we observe that the ASQ algorithm is well-designed for applications in fractal codings.

1. *Adaptive properties*
   The ASQ algorithm is initiated with a uniform model. When an input $S(n)$ lies in an interval $y_i^{(n-1)}$, for $i \in Q_1$, it belongs to Case I. The ASQ will attempt to alter (concentrate or dilate) this interval, to shift all left-hand side intervals of $y_i^{(n-1)}$ to the right (if concentrated) or left (if dilated); and to fix the right-hand side intervals. If $S(n)$ is regarded as Case II, we will change this interval upon which $S(n)$ locates, to shift all right-hand side intervals and to fix the left-hand side intervals. This is the fundamental essense of the first and third key points described earlier. Hence, a quantization model with less error will be obtained.
2. *Self-adjustabilities:*
   By (3), the new value of the reconstruction level is combined linearly by the old one and the estimated input $\tilde{S}(n)$. It reveals that the reconstruction levels will be self-adjusted by the inputs of the quantizer. Therefore, the quantized values would be more proper according to the probabilitiy of scale factors.
3. *Optimal properties:*
   The optimal sense is the most important issue in ASQ. At first, we recall (3):

$$m_i^{(n)} = \frac{m_i^{(n-1)}X_i^{(n-1)} + \tilde{S}(n)}{X_i^{(n)}},$$

$$= \frac{m_i^{(n-1)}\dfrac{X_i^{(n-1)}}{n} + \tilde{S}(n)\dfrac{1}{n}}{\dfrac{X_i^{(n)}}{n}}. \tag{15}$$

   In (15), the denominator represents the probability of the ith interval. $m_i^{(n-1)}$ is referred to the $(n-1)$th stage output with a probability of $X_i^{(n-1)}/n$. $\tilde{S}(n)$ is the new coming input with a probability of $1/n$. That is, (15) is similar to the closed form of the ith centroid as shown below,

$$i\text{th centroid} \triangleq \frac{\sum_{j=1}^{X_i^{(n)}} z_j P(z_j)}{\sum_{j=1}^{X_i^{(n)}} P(z_j)},$$

   where $P(z_j)$ is the probability corresponding to every element in the ith interval and can be considered to be $1/n$. As a result, from (15) the centroid of the ith interval, which is one optimal condition, is calculated. Moreover, the solutions in (6) and (9), which show that the decision levels

are considered as the mid-points of the neighbouring reconstruction levels, satisfy another optimal condition (Hang and Woods 1995, Wu and Hsu 1996). As a result, we conclude that the ASQ algorithm has the potential of optimality.

4. *Alterations:*
There are two alterations in the ASQ algorithm. One is to vary the ratio of a/b to change the decaying property in the modelling source. We set the ratio to be smaller if the distribution of scale factors without quantization is sharper. Otherwise, we should enlarge the ratio of a/b. Another alteration is to alter the geometric ratio γ (Widrow and Stearns 1985), which decides the adaptation speed of this algorithm.

$$m_i^{(n)} = \frac{m_i^{(n-1)} X_i^{(n-1)} + \tilde{S}(n)}{X_i^{(n)}}, \\ = \gamma m_i^{(n-1)} + (1-\gamma)\tilde{S}(n), \qquad (16)$$

where $\gamma \triangleq X_i^{(n-1)} / X_i^{(n)}$, $0 < \gamma < 1$.

The difference between the denominator and numerator is equal to the increment of $X_i^{(n)}$ in (2) and is set to 1. For example, the values of γ will be $\frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \ldots$ etc. Naturally, the value of γ will increase. As a result, the weighting factor in estimated input $\tilde{S}(n)$ is reduced. It reveals that the adaptation speed will slow down. In order to maintain constant, or increase, the adaptation performance, we have to make sure that the values of γ are either constant or decreasing. Due to the preservation of adaptation capability, this method will be less satisfying than the previous one for a source with static distributions.

4. Experimental Results

An example of a 2-D image, Lena, is presented to illustrate the function of the ASQ algorithm mentioned before. The testbed images are of 512×512 pixels with 8-bit grey levels. Daubechies' (1988, 1992) filter with length 20 is adopted in the DWT decomposition since it is orthogonal and compactly supported. A_1 represents the lowest frequency subimage of the first layer (resolution $\frac{1}{2}$) DWT decomposition. D_{h1}, D_{v1} and D_{d1} are the horizontal, vertical and diagonal oriented subimages with resolution $\frac{1}{2}$, respectively. As a result, the 3-layer DWT decomposition is derived and shown in figure 3, which reveals that the characteristic of self-similarity in the fractal would appear between scales with different resolutions. Therefore, D_{h1} is regarded as the 'domain pool' and D_{h2} is considered to be the 'range pool'.

The improvement of mean squared error (MSE) on using the fixed uniform model is listed in table 1. It reveals that the scalar quantizer: (i) must be placed in front of the IFS algorithm to reduce the composite error, and more time is spent since every possible scale factor, while searching in IFS coding, should be quantized in advance; (ii) the ASQ with forward reconstruction can perform well with no more bits being necessary to record, and the additional time used to execute the ASQ algorithm is acceptable; (iii) the backward reconstruction method can acheive the best MSE at the cost of increasing encoded bits and computing power. In practice, this reconstruction method is hard to use since some ASQ algorithms combining the IFS coding need to be checked to obtain the MSE better. This is a time-consuming task, especially in the IFS coding procedures.

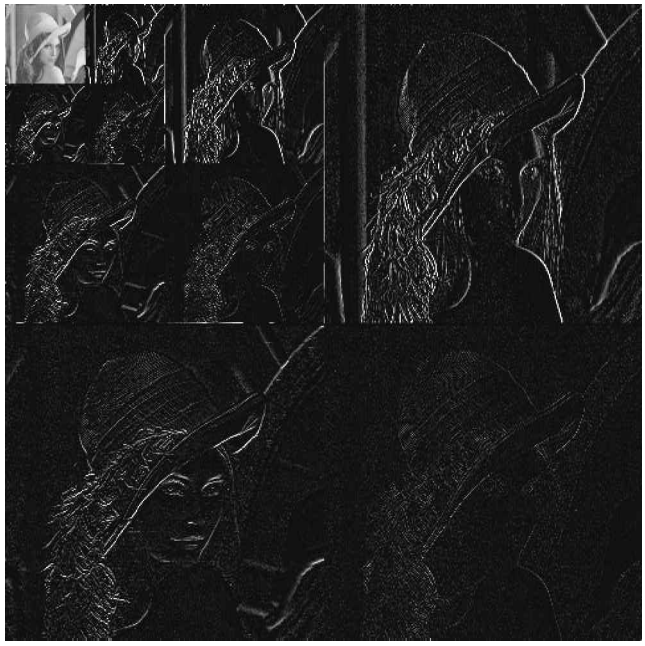

Figure 3. The image of Lena performed after the 3-layer DWT.

Next, we will show that the ASQ is superior to the uniform quantizer from the histogram point of view. The histogram of quantized data with a uniform model, as shown in figure 4(*a*), is uniformly spread on the dynamic range R. On the other hand, the histogram of quantization output with the ASQ algorithm in figure 4(*b*) is similar to that of scale factors without quantization in figure 2.

We also consider another version of the ASQ algorithm, which is more robust than the previous one since this method estimates the distribution by means of the Least Squared Error method instead of the predefined values of a and b. For the cases of $R = \pm 4$ in table 2, the modified ASQ algorithms perform well in robustness. Moreover, it also appears that the reconstruction MSE is highly related to the ratio of a/b. That is, the results would perform less successfully if the source is modified inappropriately.

Table 1. The comparison of mean squared error and execution time on uniform model 'before' and 'after' IFS coding, and the ASQ algorithm with different reconstruction modes: forward and backward. $a = 1$, $b = 5$, default dynamic range $R = \pm 8$.

			ASQ	
	Horizontal oriented	Uniform model SQ	forward	backward
MSE	SQ before IFS	12.1749/0.092	11.1665/0.108	11.1665/0.105 ($R = \pm 8$)
/Time	SQ after IFS	40.4110/0.101	12.9345/0.096	12.4675/0.107 ($R = \pm 7$)
	encoded bits	3072	3072	3328
			ASQ	
	Vertical oriented	Uniform model SQ	forward	backward
MSE	SQ before IFS	5.6952/0.097	5.3465/0.110	5.2891/0.111 ($R = \pm 7$)
/Time	SQ after IFS	20.0851/0.092	6.1019/0.113	5.9658/0.106 ($R = \pm 7$)
	encoded bits	3072	3072	3328
			ASQ	
	Diagonal oriented	Uniform model SQ	forward	backward
MSE	SQ before IFS	2.7074/0.084	2.4900/0.106	2.4780/0.105 ($R = \pm 6$)
/Time	SQ after IFS	9.0462/0.098	2.8718/0.110	2.7939/0.097 ($R = \pm 6$)
	encoded bits	3072	3072	3328

Table 2. The comparison of mean squared error in $R = \pm 4$ on ASQ and another version of ASQ with different values of a/b.In each subimage, the minimum MSE is marked ★.

	ASQ							
D_{h2} $\frac{a}{b}$	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{1}{6}$	$\frac{1}{7}$	$\frac{1}{8}$	ASQ (LSE)
MSE	11.1815	11.1993	11.3702	11.7887	12.2652	12.6266	13.0398	10.9594 ★
	ASQ							
D_{h2} $\frac{a}{b}$	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{1}{6}$	$\frac{1}{7}$	$\frac{1}{8}$	ASQ (LSE)
MSE	5.2800 ★	5.3526	5.4554	5.6068	5.7122	5.7918	5.8808	5.3677
	ASQ							
D_{h2} $\frac{a}{b}$	$\frac{1}{2}$	$\frac{1}{3}$	$\frac{1}{4}$	$\frac{1}{5}$	$\frac{1}{6}$	$\frac{1}{7}$	$\frac{1}{8}$	ASQ (LSE)
MSE	2.4712 ★	2.4780	2.4838	2.5120	2.5468	2.5884	2.6150	2.5201

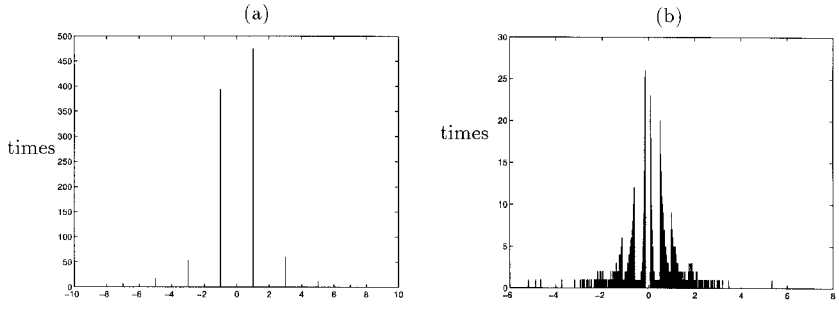

Figure 4. The histograms of the quantized output: (*a*) uniform mnodel and (*b*) ASQ algorithm with $a = 1$, $b = 5$ and $R = \pm 8$.

5. Conclusion

We have proposed the adaptive self-quantizer to solve the quantization problem of scale factors in wavelet-based fractal image compression. In source modelling, we introduced the Least Squared Error method to estimate recursively the probability density function at the values of decision levels. Therefore, the distribution can be obtained by interpolating the estimated density functions linearly. The quantizer has the functions of adaptivity and self-adjustment. In addition, it introduces the optimal sense in adapting the reconstruction levels of the quantizer such that we can reduce the quantization error generated by a fixed uniform model.

Acknowledgments

This work was supported by National Science Council under Grant NSC87-2213-E-009-043.

References

Antonini, M., Barlaud, M., Mathieu, P., and Daubechies, I., 1992, Image coding using wavelet transform. *IEEE Transactions on Image Processing*, **1**, 205–220.

Barnsley, M. F., and Jacquin, A., 1988, Application of recurrent iterated function systems to images. *Proceedings SPIE*, **1001**, 122–131.

Chan, C., and Vetterli, M., 1995, Lossy compression of individual signals based on string matching and one pass codebook design. In *Proceedings ICASSP'95,* Detroit, MI, pp. 2491–2494.

Crisafulli, S, and Bitmead, R. B., 1993, Adaptive quantization: Solution via nonadaptive linear control. *IEEE Transactions on Communication*, **41**, 741–748.

Davis, G. M., 1995, Self-quantization of wavelet subtree in *Proceedings SPIE, Wavelet Applications II,* Orlando. Edited by H. H. Szu, **2491**, 141–152.

Daubechies, I., 1988, Orthonormal bases of compactly supported wavelets. *Communications on Pure and Applied Mathematics,* **41**, 906–966; 1992, *Ten Lectures on Wavelets* (Philadelphia, PA: SIAM).

Fisher, Y., 1994, *Fractal Compression: Theory and Application to Digital Images* (New York: Springer Verlag).

Goodwin, G. C., and Sin, K. S., 1984, *Adaptive Filtering Prediction and Control* (Englewood Cliffs, New Jersey: Prentice-Hall).

Hang, H.-M., and Woods, J. W., 1995, *Handbook of Visual Communications.* (San Diego: Academic Press).

Hsu, H.-H., Hu, Y.-Q., and Wu, B.-F., 1998, An integrated method in wavelet-based image compression, *The Journal of the Franklin Institute*, **335B**, 1053–1068.

Jacobs, E. W., Fisher, Y., and Boss, R. D., 1992, Image compression: a study of the iteratied transform method. *Signal Proceeding,* **29**, 251–263.

Jacquin, A. E., 1993, Fractal image coding: a review. *Proceedings of the IEEE*, **81**, 1451–1465.

Jayant, N. S., 1973, Adaptive quantization with a one-word memory. *Bell Systems Technology Journal*, **52**, 1119–1144.

Mallat, S. G., 1989, A theory for multiresolution signal decomposition: the wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **11**, 674–693.

Max, J., 1960, Quantizing for minimum distortion. *IRE Transactions Information Theory*, **6**, 7–12.

Ortega, A., and Vetterli, M., 1997, Adaptive scalar quantization without side information. *IEEE Transactions Image Processing*, **6**, 665–676.

Rinaldo, R., and Calvagno, G., 1995, Image coding by block prediction of multiresolution subimages. *IEEE Transactions on Image Processing*, **4**, 909–920.

Steinberg, Y., and Gutman, M., 1993, An algorithm for source coding subject to a fidelity criterion, based on string matching. *IEEE Transactions on Information Theory*, **39**, 877–886.

Strang, G., and Nguyen, T., 1996, *Wavelets and Filter Banks* (Cambridge, MA: Wellesley-Cambridge).

Vetterli, M, and Kovačević, J., 1995, *Wavelets and Subband Coding* (Englewood Cliffs, New Jersey: Prentice-Hall).

Widrow, B., and Stearns, S. D., 1985, *Adaptive Signal Processing* (Englewood Cliffs, New Jersey: Prentice-Hall).

Wu, B.-F., Hsu, H.-H., and Hu, Y.-O., 1998, The global minimum of scalar quantization errors by discrete wavelet transforms in image compression, accepted by Proceedings of the National Science Council, Part A: Physical Science and Engineering.

Wu, B.-F., Hsu, H.-H., and Hu, Y.-Q., 1997, Linear block prediction in wavelet-based fractal image compression. Submitted to *Signal Processing*.