Hybrid Fractal Zerotree Wavelet Image Coding

Taekon Kim, Robert E. Van Dyck, and David J. Miller

Taekon Kim
Intel Corporation
CH7-212, 5000 W. Chandler Blvd.
Chandler, AZ 85226
Telephone (480) 552-1097
taekon.kim@intel.com

Robert E. Van Dyck
National Institute of Standards and Technology
100 Bureau Dr. Stop 8920
Gaithersburg, MD 20899
Telephone (301) 975-2923
vandyck@antd.nist.gov

David J. Miller
227C Electrical Engineering West
The Pennsylvania State University
University Park, PA 16802
Telephone (814) 865-6510
miller@perseus.ee.psu.edu

List of Symbols

$S_l^{Orientation}$	a subband at the l_{th} decomposition level and at one of three orientations
D-R	distortion-rate
T_0	initial threshold
$c(i,j)$	the wavelet coefficient at location (i,j)
$\lfloor x \rfloor$	a function which chooses the largest integer less than x
D_l	the domain tree whose coarsest coefficients are in the decomposition level l
R_{l-1}	the range tree whose coarsest coefficients are in the decomposition level l-1
x	the ordered set of coefficients of a range tree
y	the ordered set of coefficients of a subsampled domain tree isometry
T	the contractive transformation
S	subsampling
O	the orientation operation
$\boldsymbol{a}$	the scaling factor
R_a	the encoding bit rate
R_b	a rate slightly less than the encoding rate
R_f	the fractal coding rate
R_n	the new rate
$w_k(i,j)$	the coefficient of the coordinate (i,j) in the k_{th} subband
$\hat{w}_k(i,j)$	the corresponding decoded coefficient
t_k	the weighting value for the k_{th} subband

No. of Pages

Twenty-seven pages

No. of Tables

Four tables

No. of Figures

Eight figures

Keywords

hybrid fractal coding, zerotree, image compression, wavelets, perceptual image quality

Abstract

In this paper, a hybrid fractal zerotree wavelet (FZW) image coding algorithm is proposed. The algorithm couples a zerotree-based encoder, such as the embedded zerotree wavelet (EZW) coder or set partitioning in hierarchical trees, and a fractal image coder; this coupling is done in the wavelet domain. Based on perceptually-weighted distortion-rate calculations, a fractal method is adaptively applied to the parts of an image that can be encoded more efficiently relative to an EZW coder at a given rate. In addition to improving compression performance, the proposed algorithm also allows one to impose desirable properties from each type of image coder, such as progressive transmission, the zerotree structure, and range-domain block decoding.

I. INTRODUCTION

A. *Image compression in the wavelet domain*

Subband coding [27], based on the wavelet theory, provides a multi-resolution decomposition of images. The discrete wavelet transform exhibits strong decorrelating properties such as space-frequency localization, spatial clustering of significant structures within each subband, and clustering of similar structures across subbands [1]. These statistical properties of subbands [2] have been studied extensively and have been exploited by wavelet image coders. In particular, the pyramidal, or dyadic, wavelet decomposition has shown excellent energy compaction. To efficiently encode the subbands, Shapiro [24] introduced the EZW encoder, which uses both bit-plane coding and the zerotree structure. An alternative algorithm, set partitioning in hierarchical trees (SPIHT), was proposed by Said and Pearlman [22]. These EZW coders have shown excellent distortion-rate (D-R) performance with low computational complexity, while generating an embedded bit stream. This latter property enables one to send images in a progressive manner and to encode images at any target bit rate. Xiong *et al.* [29] utilized the zerotree in an adaptive manner and showed, perhaps, the best results among the zerotree-based image coders; however, the computational complexity of their method is quite high and their method is not progressive.

While EZW coders exploit the inter-subband correlation through a tree, some other coders exploit either intra-subband or inter-subband correlation through structures closely related to trees. Taubman and Zakhor [25] proposed layered zero coding (LZC) for still images and video. This coder uses adaptive arithmetic coding [26] more efficiently than other methods, but requires some amount of side information. Servetto *et al.* [23] suggested a morphological representation of the wavelet data. The clustering property of significant coefficients within and between subbands was also exploited by Chai *et al.* [3]. The last two encoders emphasized the morphologically significant structures among subbands, and they showed comparable results to zerotree-based encoders. Recently, other adaptive image coding algorithms, based on the use of side information, have shown promising results. Joshi *et al.* [11] investigated classification of image subbands. Their algorithm is a forward adaptive technique. There have been other types of coders that treat overhead information differently, called backward adaptive techniques. These methods utilize the causal (received) data to estimate the statistical properties of coefficients in the subbands [17][4][30]. By eliminating overhead, backward adaptive techniques have better distortion-rate performance than forward adaptive ones, at the cost of computational complexity at the decoder.

Fractal image compression has been shown to be very effective in exploiting the self-similarity in the spatial image domain [9][6][18]. Recently, the self-similarity in a subband decomposition has also been studied by using fractal coding methods. Pentland and Horowitz [20] first suggested the possible link. Rinaldo and Calvagno [21] proposed this type of algorithm for image coding. Davis [5] and Krupnik *et al.* [14] independently introduced wavelet-based fractal image coders that are generalizations of fractal block coding in the spatial domain. Davis showed that a total tree-based fractal compression method is very efficient in representing zerotree-like structures and straight edges that have self-similarity. Li and Kuo [15] proposed a hybrid wavelet-fractal image coder and showed good distortion-rate performance. In fact, fractal coding in the wavelet domain can be viewed as a technique for block prediction from the lower resolution subbands to the higher resolution ones.

In this paper, we propose hybrid fractal zerotree wavelet coding, yielding a coder we call a *fractal zerotree wavelet* (FZW) coder. The basic idea is to choose between the similar wavelet-domain structures that occur in EZW and tree-based fractal encoders by utilizing locally optimal, distortion-rate calculations. A fractal method is adaptively applied to the parts of an image that can be encoded perceptually losslessly and more efficiently than with an EZW coder; this usually corresponds to edges and texture areas of an image. In our previous work based on the MPEG-4 still image coding algorithm [12], the fractal coding method in the wavelet domain saved significant bits without perceptual image degradation. By basing FZW on the SPIHT coder, it can outperform SPIHT. Moreover, FZW shows one of the best distortion-rate performances in the class of fractal image coders.

B. *Paper organization*

General EZW coders and fractal image coding methods are briefly reviewed in Section II. The FZW algorithm is proposed in Section III, and simulation results with FZW are provided in Section IV. Conclusions are presented in Section V.

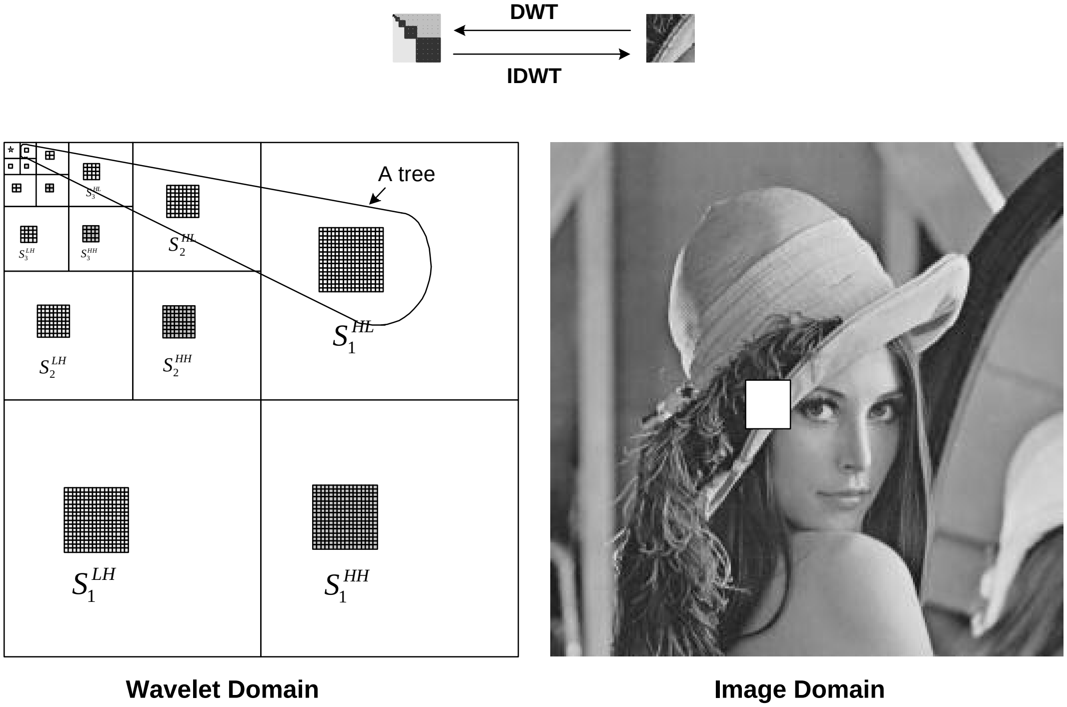


Fig. 1 Hierarchical structures in the dyadic (pyramidal) decomposed wavelet domain. A *tree* is composed of a wavelet coefficient (node) and all its descendents. Trees are formed from coefficients of the same orientation (indicated by small boxes of the same style). The union of three trees of different orientations is called a *total tree.* A total tree plus one coefficient in the LFS (star mark), comprises a *square tree.* A square tree corresponds to a square block in the image domain (white block in the right figure). The corresponding blocks in the wavelet and image domains are shown on top of the figure. $S_l^{Orientation}$ represents a subband at the l_{th} decomposition level and at one of three orientations (LH, HL, HH).

II. BACKGROUND

A. EZW Image Coding

There have been many versions of EZW coders since Shapiro introduced his algorithm in 1993 [24]. The SPIHT algorithm [22] shows excellent results in this class of coders. Here, we briefly explain the general procedures of EZW image coders. For the following review, it is important to define and understand the hierarchical structures in the wavelet domain. The tree structure, called simply a *tree*, is a set of wavelet coefficients corresponding to the same spatial location and orientation (see Fig. 1.). The assembly of three trees, which specify the same spatial location, is called a *total tree*. The union of three trees (a total tree) and one coefficient in the LFS, called a *square tree*, corresponds to a square block in the image domain. In other words, a square tree has complete information about the corresponding square block. It is noted that most EZW coders could be modified to encode each square tree, total tree, or tree, independently. The basic assumption in zerotree-based image coders is that if there are insignificant coefficients in low frequency subbands in a tree, then the corresponding coefficients in the higher frequency subbands are likely also insignificant. A tree with all coefficients insignificant with respect to a given threshold is called a *zerotree*. While zerotrees can be very efficiently coded, a substantial number of bits is required to specify non-zerotree structures.

Although there are some minor differences among the EZW image coders, their encoding procedures can be summarized by three operations: (1) the significance map pass, (2) the zerotree map pass, and (3) the refinement pass. In the significance map pass, the significance function, with respect to a given threshold, is applied to each wavelet coefficient using a predefined scanning order. The two possible results for each coefficient are significant (1 symbol) or insignificant (0 symbol). This is a form of simple binary quantization. Usually, the initial threshold T_0 is given by the following formula.

$$T_0 = 2^{\left\lfloor \log_2 (\max_{i,j} |c(i,j)|) \right\rfloor},$$

where *c(i,j)* is the wavelet coefficient at location *(i,j)*, and $\lfloor x \rfloor$ chooses the largest integer less than *x*. In the next pass, the threshold is generally decreased to $T_0/2$. In the zerotree map pass, the zerotree function, which also has two possible outputs with respect to a given threshold, is applied to the trees. If there are no significant coefficients in a tree, the zerotree function outputs the ‘insignificant’ symbol. Otherwise, this function outputs the ‘significant’ symbol, and the positions of the significant coefficients in the tree should be specified by an appropriate method for a given threshold. The choice of this specifying method determines the computational efficiency and distortion-rate performance of an EZW coder. In the refinement pass, each coefficient that turned out to be significant is given a refined quantized representation. One bit is given for each coefficient. The algorithm can stop at any time, coding in an embedded and progressive manner.

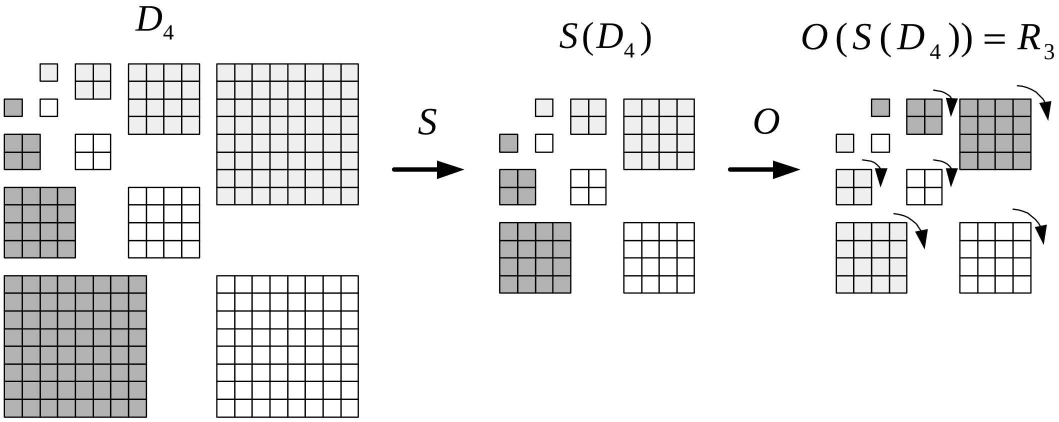


Fig. 2 Procedure of approximating a range tree using a domain tree. A domain tree (D_4) that has its node in the third decomposition level is subsampled by truncating all the coefficients in the highest frequency subbands. The size of a subsampled domain tree $S(D_4)$ is the same as that of a range tree (R_3). The orientation consists of isometry operations within subbands, and a switch of HL and LH subbands in a tree.

B. *Fractal Image Coding*

Generalizations of fractal block coding from the image domain to the wavelet domain have been proposed in recent years [20][21][5][14]. The motivation for these methods stems from the existence of self-similarities in the multiresolutional wavelet representation. In fact, fractal image coding in the wavelet domain has quite different characteristics from the spatial domain coders and can be interpreted as the prediction of a set of wavelet coefficients in the higher frequency subbands from those in the lower ones. A contractive mapping associates a domain tree of wavelet coefficients with a range tree that it approximates. Various structures have been used for the domain to range mappings [21][5][15]. A fractal coding method that uses total trees for the range and domain trees is briefly reviewed for understanding the proposed algorithm.

A range tree is fractally encoded by a bigger domain tree. The approximating procedure is very similar to that in the spatial domain: subsampling and determining the orientation and scaling factor. Note that one does not need an additive constant because the wavelet tree does not have a constant offset. Subsampling matches the size of a domain tree with that of a range tree by truncating all coefficients in the highest subbands of the domain tree. The orientation operation consists of a combination of a 90 degree rotation and a flip, and it is done within each subband. A switch of HL and LH subbands is the next step. The scale factor is then multiplied with each wavelet coefficient in the tree. Figure 2 shows some of these steps.

Let D_l denote the domain tree, which has its coarsest coefficients in decomposition level *l*, and let R_{l-1} denote the range tree, which has its coarsest coefficients in decomposition level *l-1*. The contractive transformation (T) from domain tree D_l to range tree R_{l-1}, is given by

$$T(D_l) = a * O(S(D_l)),$$

where S denotes subsampling, O is the orientation operation, and a is the scaling factor. Let $x = (x_1, x_2, ..., x_n)$ be the ordered set of coefficients of a range tree and $y = (y_1, y_2, ..., y_n)$ the ordered set of coefficients of a subsampled domain tree isometry. Then, the mean squared error is

$$MSE = \left\| R_{l-1} - T(D_l) \right\|_2^2 = \sum_{i=1}^{n} (x_i - a * y_i)^2 .$$

The optimal a is then

$$a = \frac{\sum_{t=1}^{n} x_t * y_t}{\sum_{t=1}^{n} y_t^2} .$$

We search over all eight orientations and all trees in the domain pool to find the best-matching domain tree for a given range tree. The encoded parameters are the position of the domain tree, index of orientation, and the scaling factor. The bits for the position of a domain tree vary according to the dimension of the domain pool. In practice, we store the position of either the range or the domain tree, 5 bits for the quantized scale factor, and 3 bits for specifying the orientation.

III. FRACTAL ZEROTREE WAVELET IMAGE CODING

The FZW algorithm is based on the observation that in some cases, the use of an explicit zerotree structure for a given bit plane can cost a substantial number of bits with very little contribution to image quality. That is, the basic assumptions underlying EZW coders can be broken in practice. This occurs when there are a number of trees in an image that have isolated zeros in low frequency subbands. These trees typically correspond to highly textured areas and edges. For these parts of the image, a substantial number of bits can be saved if the coding method is altered. While in general, fractal compression methods have not been as effective as the state-of-the-art image coders, they do encode certain parts of images quite efficiently. Specifically, wavelet domain fractal encoders are good at representing constant gradients, textured areas, and straight edges that have self-similarity. In other words, by selectively replacing the EZW coding trees by a fractal encoding method, it is possible to obtain distortion-rate points for the hybrid coder that are below the EZW coder's operational distortion-rate curve.

In contrast to EZW encoders, fractal coding methods in the wavelet domain usually do not generate an embedded bit stream. Moreover, fractal coding methods cannot be easily applied to the coefficients in the lowest four subbands. Despite these differences, fractal and EZW encoding can be combined. The proposed fractal coding algorithm, FZW, is based on the EZW algorithm. As in EZW, the encoding and decoding are done in the wavelet domain. However, in FZW, the regions of the image are either coded by a fractal or by an EZW method, selected according to a distortion-rate criterion.

A. Encoding Procedure

The fundamental concept of fractal encoding is to find the best matching domain tree for each range tree, in the sense of minimizing a chosen distance metric. Often, both the range and domain trees consist of uncompressed wavelet coefficients. However, if one considers both rate *and* distortion when choosing the matching domain tree for a given range tree, then the chosen domain tree may depend on the encoding rate of the system. This fact forces us to perform EZW encoding and decoding (called trial-encoding and trial-decoding) for a given bit rate before fractal encoding in FZW starts. The FZW encoding method works as follows: (1) EZW trial encoding and decoding are performed to a rate near (and slightly above) a target rate. (2) Then, the instantaneous slope of the D-R curve of EZW near the given bit rate is calculated. (3) In this EZW decoded wavelet domain, a range-domain search is performed, with the D-R slope value for adopting the fractal coding method for each range tree calculated and compared to the D-R slope of EZW. If this alternative fractal coding yields a superior D-R characteristic, the range tree is fractally encoded. (4) The fractal encoding information is side information, which is placed before the EZW bit stream. Then, the bits that specify the EZW coding for these fractally encoded parts are eliminated from the EZW bit stream.

To both retain and enhance the progressive property of EZW coders, the FZW decoder applies the fractal decoding process at each bit-plane threshold, ***i.e.,*** improved fractal range tree estimates are formed based on the refined domain trees obtained at each bit plane threshold. Note that this decoding is not an exact inverse of the FZW encoding. An alternative decoding approach (where the decoder is an exact inverse of the encoder) would first entirely EZW decode the image (to the target rate) and only then apply fractal decoding to the appropriate range trees. This would reduce computational complexity, since the fractal decoding is then done only once, instead of at each bit plane. There would still be some progressiveness, obtained by use of the EZW algorithm. However, our proposed approach of doing fractal decoding at each bit plane is supported from three standpoints: (1) The increase in computational complexity is not that large, since only one fractal iteration is done for each bit plane. This fact should be contrasted with typical block-based fractal decoders that perform a number of iterations. (2) There is no degradation in performance at the target rate compared to the alternative decoding approach, because the final fractal decoding uses the entire EZW decoded image in both cases (The fractal decoded range trees are overwritten from one bit plane to the next.). 3) Most importantly, we have found that, although our method is optimized for a target rate, it achieves gains in progressive performance (attributable to fractal decoding at each bit plane) not only in comparison with the alternative decoding approach, but also in comparison with standard SPIHT. Thus, the proposed progressive fractal decoding makes FZW an even more progressive scheme. Further discussion on progressive decoding, including performance results, is given in the Experimental Results section.

Initially, as some trees are selected for fractal coding, the distortion-rate gain over the EZW coder grows. However, as more and more trees are fractally encoded, further reducing the rate, the gains over EZW shrink. This occurs because the FZW method seeks improvements via fractal coding starting from an EZW coder at a specified rate. Thus, improvement can only be achieved over a region of rates less than (but in the vicinity of) the target rate. To summarize, the proposed method seeks improvement over EZW coders in a local region near (and below) the initial (pure EZW coder) rate. This characteristic of FZW is shown in Fig. 3. This figure also indicates the rate at which the FZW method achieves its maximum gain over EZW.

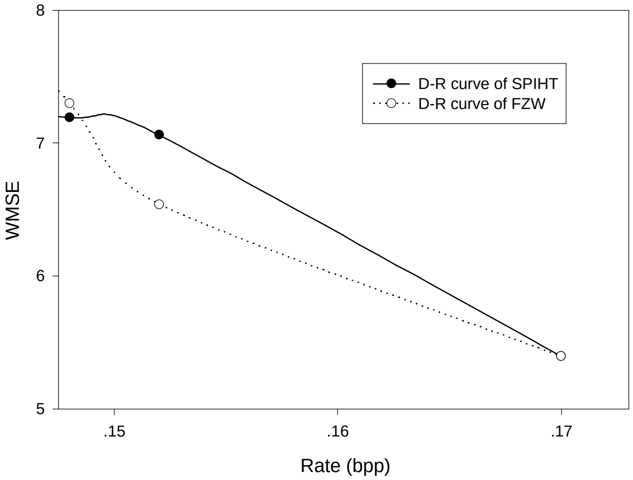


Fig. 3 Distortion-rate curves of SPIHT and FZW on the Lena image for encoding rates near 0.170 bpp. Using a distortion-rate selection of trees to be fractally encoded, FZW approaches its optimal gain at a rate slightly above 0.15 bpp. As more range trees are adopted beyond this point, the gain of FZW is reduced. The distortion measure is a weighted MSE measure (described in Section IV).

B. FZW algorithm

Encoder

Step 1 (*discrete wavelet transformation*): transform an image into a pyramid form for a given number of decomposition levels.

Step 2 (*trial-encoding*): encode an image using an EZW coder at a given bit rate, and generate Count_map and the EZW bit stream. Count_map is a square matrix whose elements hold the number of bits and encoding information of range trees for the square trees. The matrix dimension is the same as the size of the lowest frequency subband.

Step 3 (*trial-decoding*): decode an image using the bit stream, and calculate the mean squared error (MSE) distortion[1] between the original and reconstructed images at the encoding bit rate, R_a, and at a bit rate slightly less than the decoding rate, R_b. Denote these MSE values as MSE_a and MSE_b, respectively. Next, determine the instantaneous slope of the distortion-rate curve for the image at this bit rate using

$$D-R\,slope=(MSE_a-MSE_b)/(R_a-R_b)\ .$$

Without loss of generality, we assume that this value is always negative.

Step 4 (*fractal range-domain search*): find the best matching domain tree for each range tree; save the fractal coding parameters and the distortion value MSE_f between the original range and fractal encoded range trees. The fractal coding parameters include the position of the domain tree, orientation, and the scaling factor. The fractal coding rate, R_f, is fixed, i.e. the fractal coding cost is a constant number of bits.

Step 5 (*distortion-rate comparison*): compute the number of encoded bits when a range tree, among the four range trees in a total tree, is eliminated from the EZW coding process. Denote this value as R_{TT-RT}. The new rate R_n is

$$R_n=R_{TT-RT}+R_f\ .$$

The D-R slope value for adopting the fractal coding method (*D-R fractal*) is then

[1] We describe the FZW algorithm with the mean square error as the distortion measure. However, other distortion measures can also be used – we will introduce a perceptual distortion measure in the results section.

$$D - R\,fractal = \frac{MSE_a - MSE_f}{R_a - R_n},$$

where MSE_f is the mean squared error when using this combined fractal method. Without loss of generality, $MSE_a - MSE_f$ is assumed to be negative. By comparing *D-R slope* and *D-R fractal*, we determine if fractal coding is applied to a range tree or not. When $R_a - R_n$ is positive and the following condition is satisfied, then fractal coding is adopted:

$$\left| D - R\ slope \right| \geq \left| D - R\ fractal \right|.$$

This step is repeated for the other three range trees in the total tree. Apply Step 5 for every total tree.

Step 6 (*encoding*): send side information for each square tree first. This includes one bit to specify if fractal coding is applied to at least one range tree and four bits to specify the positions of the fractal encoded range trees. The fractal coding parameters follow. Using information in Count_map, the encoded bits for these fractally encoded parts are eliminated from the EZW bit stream. A further discussion of the bits used for fractal encoding is provided in the Experimental Results section.

Decoder

Step 1 (*side information*): read side information of image size, decomposition level, initial threshold, and fractal coding for each total tree.

Step 2 (*EZW decoding*): decode using the EZW coder starting from the initial threshold. Apply EZW decoding to the parts of the image EZW-encoded at this threshold. After this step, make the threshold half.

Step 3 (*fractal decoding*): apply fractal decoding for the fractal encoded parts of the image using both side information and the decoded image in Step 2. These decoded parts are overwritten (at every threshold, fractal decoding is performed and overwritten). Due to this step, FZW is more progressive in transmission.

Step 4 (*iteration*): repeat Steps 2 & 3 until the EZW bit stream ends.

Step 5 (*inverse discrete wavelet transformation*): perform inverse transformation.

IV. EXPERIMENTAL RESULTS

A. *Results with the mean square error distortion*

For our experiments, we chose the set partitioning in hierarchical tree (SPIHT) algorithm [22] as our EZW image coder. This algorithm shows excellent performance without iterative computations. The biorthogonal 9/7 filter bank [1] and 512 × 512 gray scale images with 8 bits per pixel are used for the experiments. Six-level decompositions are constructed by a symmetric extension at the image edges. The domain tree is a total tree that has the lowest frequency coefficients in the fifth decomposition level. That is, the domain trees, D_5, are used, and they form disjoint domain pools. The corresponding range trees, R_4 , are total trees in the fourth decomposition level. Eight bits are needed to select from the domain pool of size 16 × 16. Five bits are used for the scale factor and three bits for the orientation. So, the fractal coding costs are fixed at 16 bits per total tree (range tree). Additionally, an overhead of 256 bits per image is required to indicate whether fractal coding is applied to any range trees in each domain tree in D_5. If fractal coding is used in a domain (square) tree, then 4 additional bits are needed to specify the 15 possible combinations of 4 range trees.

Table I compares the D-R performance of FZW versus SPIHT and other image coders including: (1) fractal block coding in the spatial domain (FRAC) [8], (2) a predictive pyramid coder (PPC) [21], (3) self-quantization of subtrees (SQS) [5], and (4) a hybrid wavelet-fractal coder (WFC) [15]. The PPC and SQS algorithms encode the image using only a fractal method. WFC combines the layered zero coder (LZC) [25] and a fractal coding method. The two hybrid coders, WFC and FZW, demonstrate excellent performance compared to other fractal coding methods, and they achieve performance comparable to state-of-the-art wavelet image coders. A D-R comparison of just the fractal coders is plotted in Fig. 4.

The fractal encoded parts of the Lena image are indicated by square boxes in Fig. 5. As we expect, most of the boxes are related to edges and texture areas, and FZW encodes these parts efficiently without image degradation. Note that only a handful of the boxes are fractal encoded, and this is why the increase in PSNR is only 0.02 dB. Moreover, there are other edges in the image that are not fractal encoded, suggesting that further performance improvements are possible. This topic is addressed in the next subsection, where a perceptually-based distortion measure is used.

One of the most noticeable results of this experiment is that the fractal range-domain relation pattern in FZW is quite different from that in fractal coders that are based in the spatial domain. In spatial domain fractal coders, a range block tends to be spatially close to its matching domain block – the distribution of spatial distances between range and domain blocks is highly peaked at zero distance [10][28]. In other words, a range block is likely to be encoded by a domain block that includes the range one. This tendency enables one to design a domain pool that is spatially restricted to a range block, which can speed up the encoding time. The range-domain relation patterns on the Lena image for FZW are shown in Fig. 6 (at given rates). The histogram counts in Fig. 6. are nearly uniform for different values of the range-domain distance. Furthermore, these figures show that there are few range trees that have a matching tree that is in the same square tree. These characteristics of fractal coding in FZW have been exploited by an error resilient variant of the FZW algorithm [13], although this robust algorithm is not included in this paper.

Table I
Rate-PSNR performance of various image coders on the Lena image.

	Rate	PSNR	Rate	PSNR	Rate	PSNR	Rate	PSNR	Rate	PSNR
	(bpp)	(dB)	(bpp)	(dB)	(bpp)	(dB)	(bpp)	(dB)	(bpp)	(dB)
FRAC [8]					0.218	30.71	0.448	33.40	0.763	35.92
PPC [21]			0.18	31.20	0.26	32.78	0.37	34.00		
SQS [5]	0.036	25.86	0.080	28.55	0.180	31.80	0.366	34.92	0.767	38.14
WFC [15]	0.036	26.42	0.083	29.41	0.182	32.68	0.369	35.84	0.758	39.02
SPIHT[22]	0.036	26.49	0.083	29.47	0.189	32.90	0.386	36.08	0.787	39.25
FZW	0.036	26.49	0.083	29.48	0.189	32.92	0.386	36.11	0.787	39.28

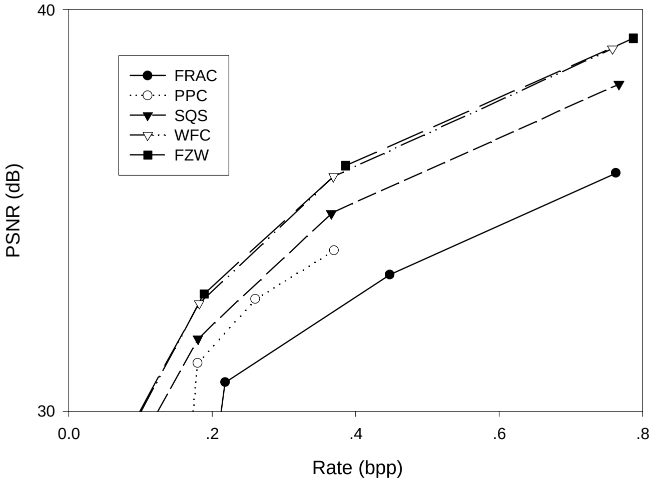


Fig. 4 Distortion-rate comparison of fractal image coders for the Lena image.

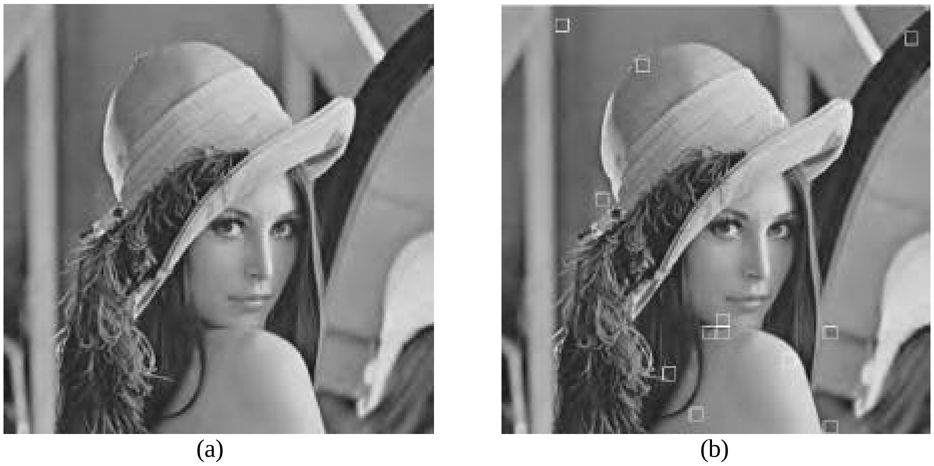
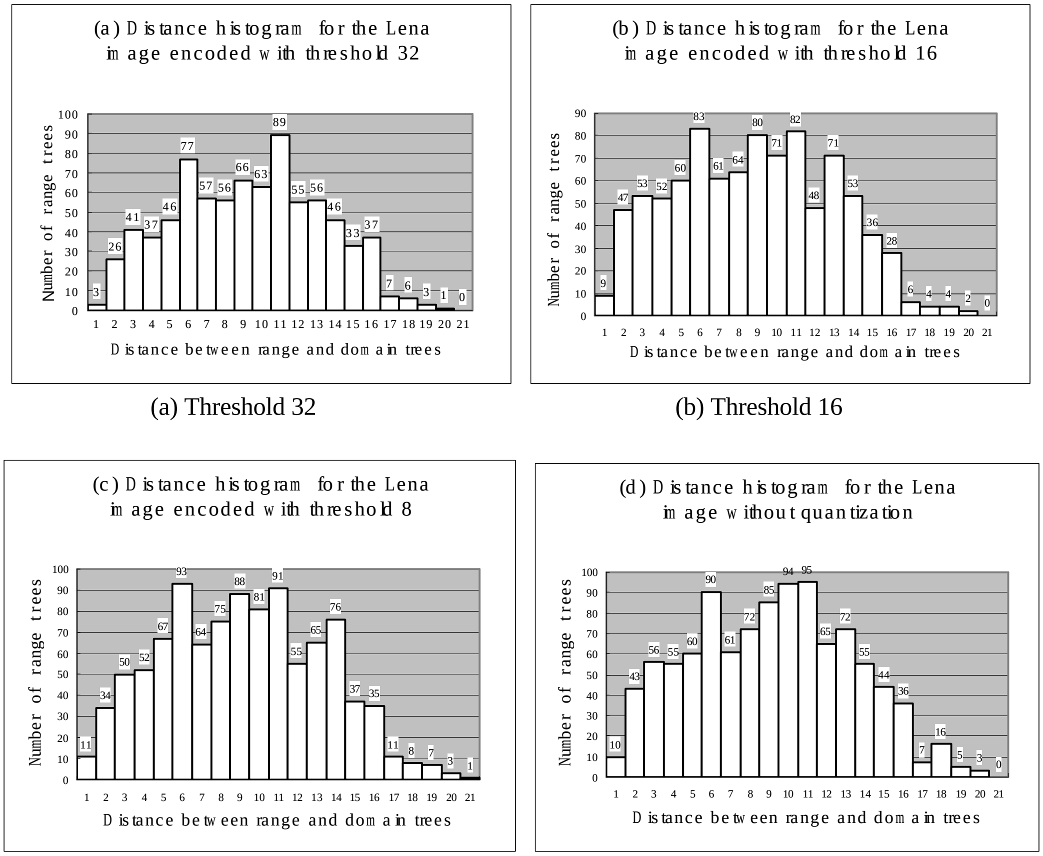


Fig. 5 The SPIHT (a) and FZW (b) images encoded at 0.189 (bpp). (a) PSNR is 32.90 (dB). (b) PSNR is 32.92 (dB). The parts of the image encoded with the fractal method are marked by white boxes. These areas are highly textured or constant regions.



(a) Threshold 32

(b) Threshold 16

(c) Threshold 8

(d) Without quantization

Fig. 6 Distance distribution of the range-domain relation at various encoding rates for the Lena image. The distance is measured between the matching domain tree and the domain tree that includes a range tree. The 16 x 16 disjoint domain pool is used in the experiment. The distance is based on the coordinates of two domain trees and ranges from 0 to 21. For example, distance 0 means that the two domain trees are identical and distance 1 means that the matching domain is one of the four nearest domain trees. The range tree that consists of all zero coefficients is called zero-range, and this is excluded in this measurement.

B. *Results with a weighted mean square error distortion*

While FZW shows superior performance to other fractal coders, it does not significantly improve on the performance of SPIHT. The reason is that the parts of the image that can be encoded more efficiently with fractal coding than with SPIHT are relatively few. There are two factors contributing to this phenomenon. Firstly, the normalization unit of SPIHT is multiplied successively to coefficients in the same decomposition level. Secondly, fractal coding is better suited for a perceptually-tuned distortion measure, rather than MSE. The first factor, the normalization unit operating in the wavelet domain, emphasizes the coefficients in low frequency subbands more than ones in high frequency subbands. In other words, this unit usually forces hierarchical structures to be zerotree-like. In general, EZW coding is more efficient for these structures than fractal coding. More seriously, fractal coding in the wavelet domain does not match well with the MSE measure, and this problem limits the performance of FZW. Instead of coding each coefficient in a total tree as in SPIHT, fractal coding predicts a block of coefficients as a whole. This coding often causes significant MSE distortion for some coefficients in the higher frequency subbands, yet without much perceived image degradation.

Recently, research on perceptual image coding has been done, including coders related to the EZW algorithm [19][7][16]. These coders are based on minimizing perceptual image distortion measures that are matched to the human visual system (HVS). The perceptual weightings for the subbands, which are based on the "just noticeable distortion" (JND), are used to increase perceptual image quality. A perceptual distortion measure that is commonly used is based on the Minkowski metric, and is called weighted mean square error (*WMSE*). This is given by

$$WMSE = \frac{1}{N^2} \sum_{k} \sum_{i,j} \left[\frac{w_k(i,j) - \hat{w}_k(i,j)}{t_k} \right]^2 ,$$

where $w_k(i,j)$ is the coefficient of the coordinate *(i,j)* in the k_{th} subband, $\hat{w}_k(i,j)$ is the corresponding decoded coefficient, and t_k is the weighting value for the k_{th} subband.

The order of subbands in an *l*-level decomposition is given by

$$\{1, 2, \cdots, M\} = \{S_i^{LL}, S_l^{HL}, S_i^{LH}, S_l^{HH}, S_{l-1}^{HL}, S_{l-1}^{LH}, \cdots, S_1^{HH}\}$$

where $M = 3l + 1$. In this experiment, the following set of weights is adopted for the six-level decomposition [16]:

$$W = \{t_k\} = \{1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.0, 1.37, 1.37, 1.5, 1.88, 1.88, 2.2, 2.6, 2.6, 3.3, 3.6, 3.6, 5.0\}.$$

FZW is modified to use *WMSE* for the distortion measure instead of *MSE,* and the results are given in terms of perceptual PSNR (PPSNR), which is defined by

$$PPSNR = 10 \times \log \frac{255^2}{WMSE}.$$

Tables II, III, and IV show the distortion-rate comparisons of FZW and SPIHT for the Lena, Barbara and Goldhill images, respectively. Note that for the Lena and Barbara images, the gain in PPSNR is higher at lower (target) bit rates. This suggests the fractal interpolation is able to give a reasonable approximation to the image, even at bit rates as low as 0.04 to 0.06 bpp. At the relatively high rates of 0.652 and 0.699 bpp, there are still noticeable gains of 0.21 and 0.16 dB. The Goldhill image is more complex, so it is difficult to obtain a large performance improvement at 0.028 bpp. Still, there is a gain of 0.09 dB over SPIHT. This gain increases to 0.21 dB at 0.064 bpp, with gains also observed at the other rates.

Fig. 7 shows SPIHT and FZW decoded images of Lena at 0.158 bpp. The PPSNR is the same, and the perceptual image quality is almost the same[2]. A 3.8% savings in the encoding rate is achieved.

Table II
Distortion (*PPSNR*)-rate performances of SPIHT and FZW for the Lena image.

Rate (bpp)	0.067	0.152	0.301	0.652
SPIHT (*PPSNR*)	35.37	39.64	43.49	48.05
FZW (*PPSNR*)	35.73	39.98	43.72	48.26
Gain (dB)	0.36	0.34	0.23	0.21

[2] There are some minor perceptual differences, such as slightly more distortion in the edges of the hat. However, the perceptual quality of the two images is very close.

Table III
Distortion (*PPSNR*)-rate performances of SPIHT and FZW for the Barbara image.

Rate (bpp)	0.041	0.137	0.334	0.699
SPIHT (*PPSNR*)	29.84	33.60	38.01	43.45
FZW (*PPSNR*)	30.37	34.01	38.20	43.61
Gain (dB)	0.51	0.41	0.19	0.16

Table IV
Distortion (*PPSNR*)-rate performances of SPIHT and FZW for the Goldhill image.

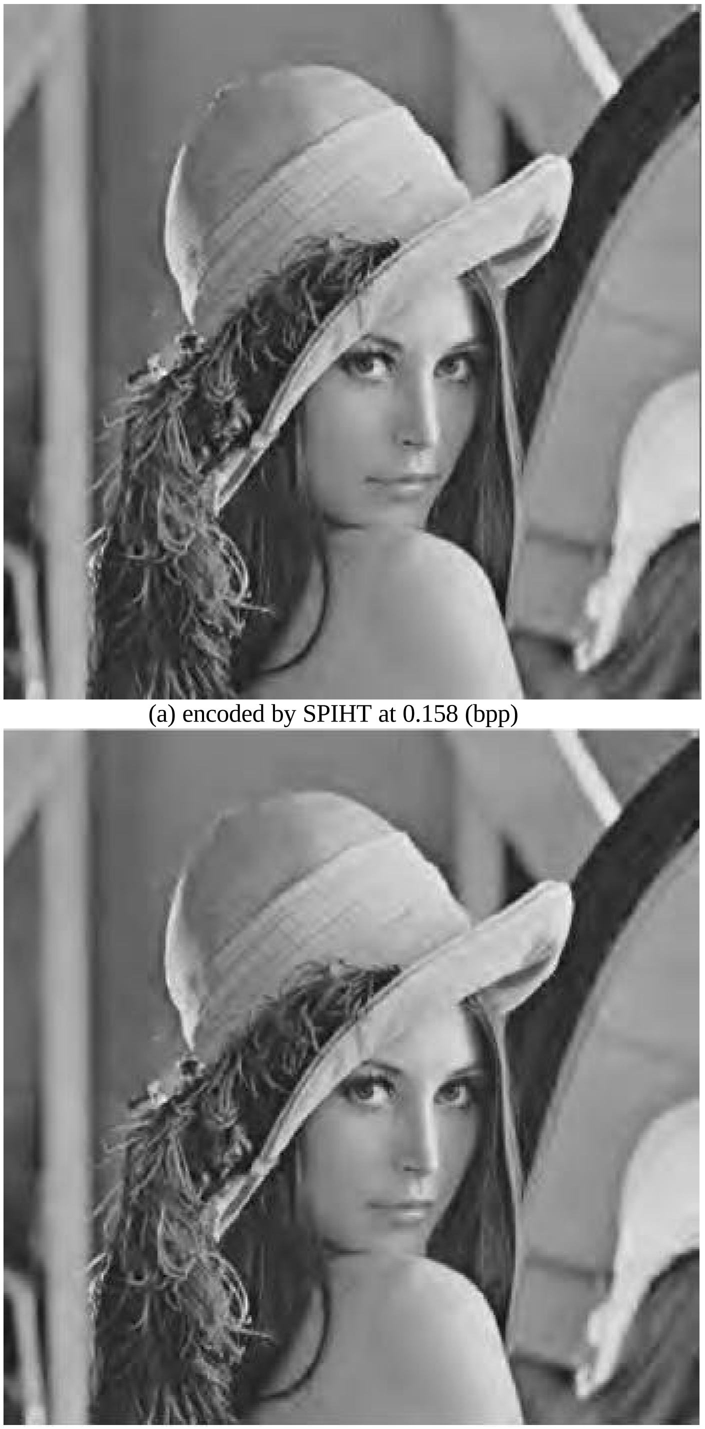
Rate (bpp)	0.028	0.064	0.203	0.499
SPIHT (*PPSNR*)	31.30	34.00	38.84	42.30
FZW (*PPSNR*)	31.39	34.21	38.99	42.41
Gain (dB)	0.09	0.21	0.15	0.11

C. *Progressiveness of the FZW Algorithm*

The fact that the FZW algorithm maintains significant progressiveness is one of its key benefits. Table V shows the progressive performance of standard SPIHT, FZW, and a lower complexity variant of FZW, denoted FZW_alt, which does the fractal decoding only once, after completing the SPIHT decoding up to the target rate. In this experiment, the target rate is set at 0.652 bpp, and the image is encoded at this bit rate by all three encoders. During the progressive decoding process, the PPSNR is computed at three intermediate bit rates, 0.071, 0.153, and 0.302 bpp. Over this entire range, FZW is able to outperform standard SPIHT. A gain of 0.45 dB is achieved at the lowest bit rate. Fig. 8 shows the images corresponding to Table V.

A few comments are in order. Firstly, the use of fractal decoding at each bit plane leads to a substantial increase in the progressive decoding performance. This is seen by comparing the results of FZW and FZW_alt. At 0.302 bpp, there is a gain of over 3 dB. At the target rate of 0.652 bpp, this gain increases to almost 7 dB before FZW_alt does its single fractal decoding pass. However, after FZW_alt fractal decodes, at the target rate, its performance is the same as that of FZW.

Note that the substantial progressive gains of FZW over FZW_alt come essentially “for free”, i.e. without any increase in rate. Secondly, comparing Tables II and V, one sees that the progressive FZW performance achieved at 0.153 bpp (based on a target rate of 0.652 bpp) is higher than the performance of FZW when the target rate is directly set at 0.152 bpp. This result is somewhat surprising. However, there are a couple of possible explanations. Firstly, the range trees that are chosen for fractal encoding are different for different target rates. In this case, there are more trees fractally encoded at the higher target rate of 0.652 bpp than at 0.152 bpp. Second, since the fractally-encoded range trees are different, the SPIHT-decoded domain trees (available for fractal interpolation) are not identical. Both of these factors may be contributing. We conjecture that this phenomenon will not hold in general, i.e. for arbitrary higher and lower target rates.



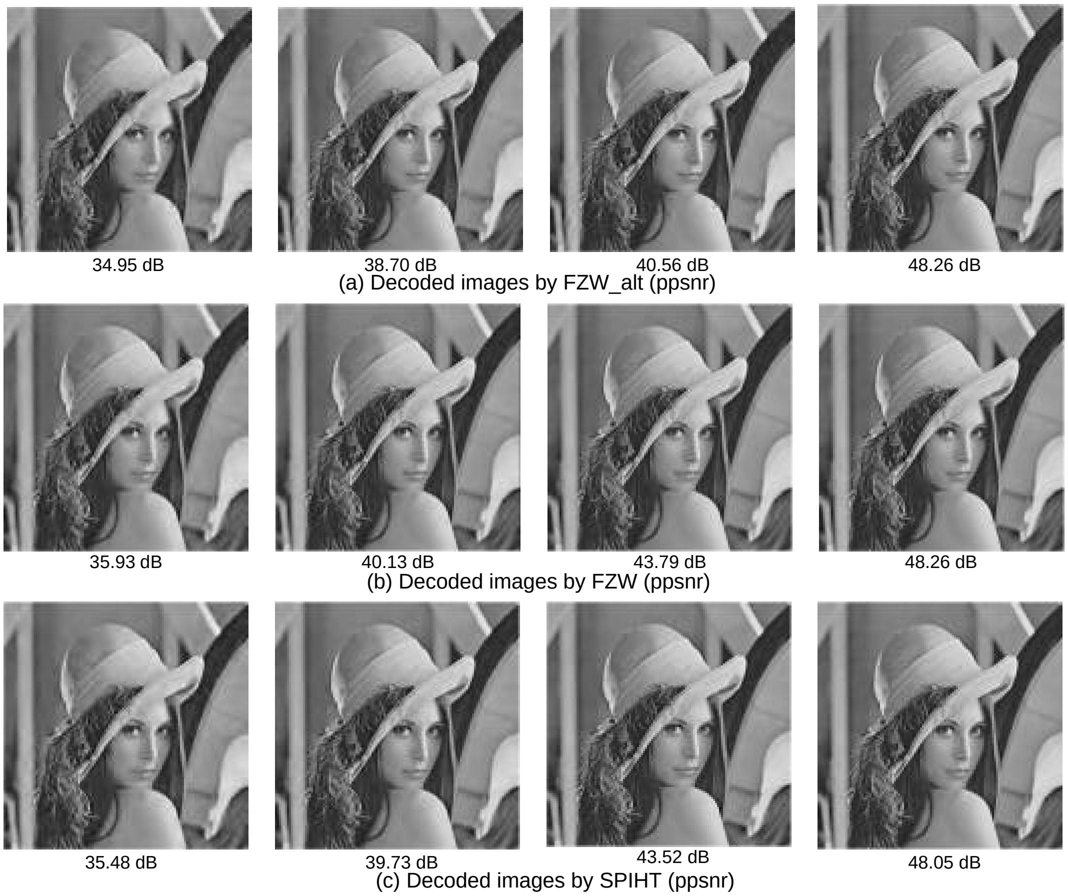
(a) encoded by SPIHT at 0.158 (bpp)

(b) encoded with FZW at 0.152 (bpp)

Fig. 7 Lena images coded by SPIHT and FZW. Both images have the same WMSE (6.54).

Table V
Comparison of the progressiveness of SPIHT and the two FZW decoding algorithms for the Lena image. FZW applies fractal coding at each coding pass, while FZW_alt applies fractal encoding only once at the very end. The Lena image is encoded at 0.652 bpp.

Rate (bpp)	0.071	0.153	0.302	0.652	0.652
SPIHT (*PPSNR*)	35.48	39.73	43.52	48.05	48.05
FZW (*PPSNR*)	35.93	40.13	43.79	48.26	48.26
FZW_alt (*PPSNR*)	34.95	38.70	40.56	41.44	48.26



(a) Decoded images by FZW_alt (ppsnr)

(b) Decoded images by FZW (ppsnr)

(c) Decoded images by SPIHT (ppsnr)

Fig.8 Comparison of the degree of progressiveness of FZW_alt, FZW, and SPIHT.

V. CONCLUSIONS

While extensive research has been done on fractal image compression, fractal encoders are generally not competitive with state-of-the-art image coders. Likewise, significant bits can be saved if we modify some coding schemes in EZW coders. In this paper, we proposed a hybrid fractal zerotree wavelet image coding algorithm that couples EZW and fractal image coding. This new algorithm reduces the bit rate required to achieve a given level of perceptual image quality (as measured by WMSE). It also keeps desirable properties from both types of coders, including progressive transmission, the zerotree structure, and range-domain block decoding.

VI. REFERENCES

[1] M. Antonini, M. Barlaud, P. Mathieu, I. Daubechies, Image coding using wavelet transform, IEEE Trans. on Image Processing, 1 (2) (April 1992) 205-220.

[2] R. W. Buccigrossi and E. P. Simoncelli, Image compression via joint statistical characterization in the wavelet domain, IEEE Trans. on Image Processing, 8 (12) (December 1999) 1688-1701.

[3] B. -B. Chai, J. Vass, and X. Zhuang, Significance-linked connected component analysis for wavelet image coding, IEEE Trans. on Image Processing, 8 (6) (June. 1999) 774-784.

[4] C. Chrysafis and A. Ortega, Efficient context based entropy coding for lossy wavelet image compression, in Proc. Data Compression Conf., Snowbird, UT, 1997.

[5] G. Davis, A wavelet-based analysis of fractal image compression, IEEE Trans. Image Processing, 7 (2) (Febrary 1998) 141-154.

[6] Y. Fisher, Fractal Compression: Theory and application to Digital images, New York: Springer-Verlag, 1994.

[7] L. Hontsch, L. J. Karam, and R. J. Safranek, A perceptually tuned embedded zerotree image coder, in Proc. IEEE Int. Conf. Image, 1997.

[8] E. W. Jacobs, Y. Fisher, and R. D. Boss, Image compression: A study of the iterated transform method, Signal Processing, 29 (1992) 251-263.

[9] A. E. Jacquin, Image coding based on a fractal theory of iterated contractive image transformations, IEEE Trans. Image Processing, 1 (1) (January 1992) 18-30.

[10] A. E. Jacquin, Fractal image coding: A review, Proc. IEEE, 81 (October 1993) 1415-1465.

[11] R. L. Joshi, H. Hafarkhani, J. H. Kasner, T. R. Fisher, N. Farvardin, and M. W. Marcellin, Comparison of different methods of classification in subband coding of images, IEEE Trans. on Image Processing, 6 (11) (November 1997) 1473-1486.

[12] Taekon Kim and R. E. Van Dyck, Hybrid wavelet fractal image coding for MPEG-4, Proceedings of the 2000 Conference on Information Sciences and Systems, Princeton, NJ, Mar. 2000.

[13] Taekon Kim, R. E. Van Dyck, and D. J. Miller, Robust fractal zerotree wavelet image coding for noisy channel transport, In preparation for submission to IEEE Trans. on Image Processing.

[14] H. Krupnik, D. Malah, and E. Karnin, Fractal representation of images via the discrete wavelet transform, in IEEE 18th Conf. Of EE in Israel, Tel-Aviv, March 1995.

[15] J. Li and C. –C. Kuo, Image compression with a hybrid wavelet-fractal coder, IEEE Trans. Image Processing, 8 (6) (June 1999) 868-874.

[16] K. T. Lo, X. D. Zhang, J. Feng, and D. S. Wang, Universal perceptual weighted zerotree coding for image and video compression, IEE Proc.-Vis. Image Signal Process, 147 (3) (June 2000).

[17] S. M. LoProso, K. Ramchandran, and M. T. Orchard, Image coding based on mixture modeling of wavelet coefficients and a fast estimation-quantization framework, in Proc. Data Compression Conf., Snowbird, UT, 1997.

[18] Ning Lu, Fractal Imaging, Academic Press, 1997.

[19] T. N. Pappas, T. A. Michel, and R. O. Hinds, Supra-threshold perceptual image coding, in Proc. IEEE Int. Conf. Image Processing, 1996.

[20] A. Pentland and B. Horowitz, A practical approach to fractal-based image compression, in Proc. Data Compression Conference, Snowbird, UT, March 1991, pp. 176-185.

[21] R. Rinaldo and G. Calvagno, Image coding by block predition of multiresolution subimages, IEEE Trans. Image Processing, 4 (7) (July 1995) 909-920.

[22] A. Said , W. A. Pearlman, A new, fast, and efficient image codec based on set partitioning in hierarchical trees, IEEE Trans. on circuits and systems for video technology, 6 (3) (June 1996) 243-250.

[23] S. D. Servetto, K. Ramchandran, and M. T. Orchard, Image coding based on a morhpological representation of wavelet data, IEEE Trans. on Image Processing, 8 (9) (September 1999) 1161-1174.

[24] J. M. Shapiro, Embedded image coding using zerotrees of wavelet coefficients, IEEE Trans. on Signal Processing, 41 (December 1993) 3445-3462.

[25] D. Taubman and A. Zakhor, Multirate 3-D subband coding of video, IEEE Trans. on Image Processing, 3 (5) (September 1994) 572-588.

[26] I. H. Witten, M. Neal, and J. G. Cleary, Arithmetic coding for data compression, Commun. ACM, 30 (June 1987) 520-540.

[27] J. W. Woods and S. D. O'Neil, Subband coding of Images, IEEE Trans. Acoust, Speech, Signal Processing, vol. ASSP-34 (October 1986) 1278-1288.

[28] S. J. Woolley and D. M. Monro, Optimum parameters for hybrid fractal image coding, in Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, Detroit, MI, 1995.

[29] Z. Xiong, K. Ramchandran, and M. T. Orchard, Space-frequency quantization for wavelet image coding, IEEE Trans. on Image Processing, 6 (May 1997) 677-693.

[30] Y. Yoo, A. Ortegaod, and B. Yu, Image subband coding using context-based classification and adaptive quantization, IEEE Trans. on Image Processing, 8 (12) (December 1999).