Resolution enhancement of images using fractal coding

Mohammad Gharavi-Alkhansari†
Robert DeNardo†
Yoichi Tenda‡
Thomas S. Huang†

†Beckman Institute and Coordinated Science Laboratory
University of Illinois at Urbana-Champaign, Urbana, Illinois, U.S.A. 61801

‡Information Technology R&D Center
Mitsubishi Electric Corporation, Ofuna, Kamakura, Kanagawa, JAPAN

ABSTRACT

The code generated by fractal coding of a digital image provides a resolution-independent representation of the image as this code can be decoded to generate a digital image at any resolution. When the image is decoded at a size larger than the original encoded image, image details beyond the resolution of the original image are predicted by assuming local self-similarity in image at different scales. In this paper, we (1) present a formulation of how decoding may be done at a higher resolution, (2) evaluate the accuracy of the predicted details using a frequency analysis of fractally enlarged test images, and (3) propose a method for fractal resolution enhancement without the low-frequency loss of information due to fractal coding.

Keywords: Fractal, Image, Coding, Compression, Resolution Enhancement, Interpolation

1 CLASSICAL IMAGE INTERPOLATION

Image Interpolation is an important operation in image processing and is used for image enlargement. One classical method for image interpolation is based on sampling theory and is done by using the frequency components of the smaller image for the low frequency components of the enlarged image, and by assuming that the high-frequency components of the enlarged image are all zero.[1] This principle is shown in Figure 1. In this approach, it is basically assumed that the enlarged image is bandlimited. The high-frequency components of the enlarged image are assumed to be zero because these information of the original image have been lost in the sampling process. Although bandlimitedness is usually assumed in the Fourier or DCT domain, it may also be assumed in the wavelet domain. For example, interpolation assuming bandlimitedness in the Haar wavelet domain is equivalent to image enlargement by pixel repetition. The bandlimitedness assumption, however, is not generally true, and if the enlarged image had been obtained by a direct sampling of a continuous image, it would have had non-zero high-frequency components.

2 FRACTAL CODING

The essence of most fractal-based image coding methods is to approximate each segment of the image by applying a (contractive) transformation on some bigger segment(s) in the image. One can then reconstruct the image (with some

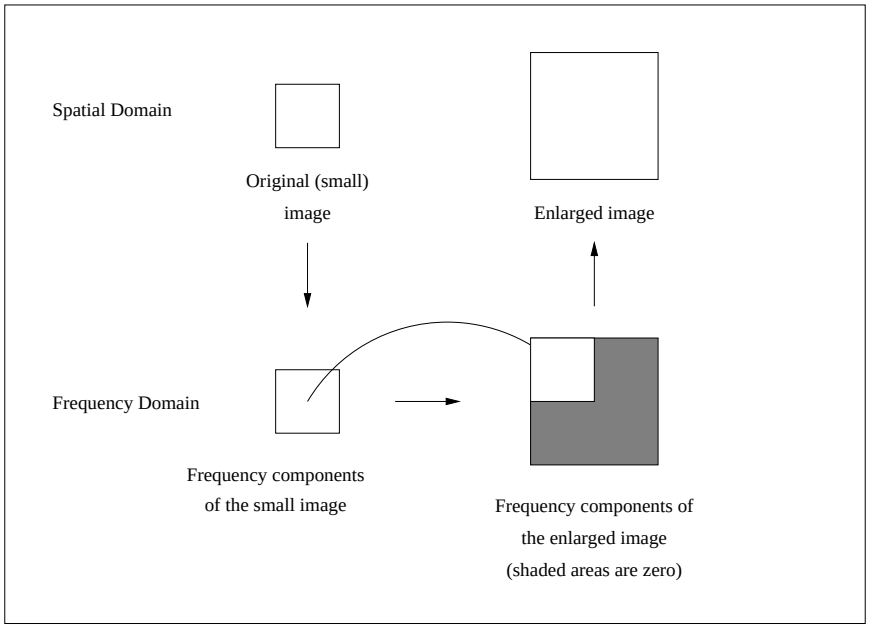


Figure 1: Classical image interpolation

error) by using only the parameters of the transformations.[2] In these methods, most of the information in the image is encoded by coding relationships among different segments (of different sizes) of the image. These methods are based on *Local Iterated Function Systems*.[3] The first automatic image compression algorithm based on this theory was implemented by Jacquin.[2]

Fractal coders typically encode an image by representing it with the parameters of a dynamical system whose attractor is close to the given image. The analysis provided in this paper is for the 1-D case. For the 2-D case the analysis and the properties are very similar to that of the 1-D case.

We begin with real discrete-time signals of finite length N. These signals may be represented by N-dimensional vectors in R^N. Let

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix}$$

be such a signal. (In the 2-D case, $\mathbf{x}$ may, for example, be generated by scanning an image row-by-row.) Then $\mathbf{x}$ may be coded by designing a transformation $\mathcal{T}$ such that the dynamical system described by the equation

$$\mathbf{x}_{n+1} = \mathcal{T}(\mathbf{x}_n) \tag{1}$$

has the property that for any initial vector $\mathbf{x}_0 \in R^N$,

$$\mathcal{T}^{\infty}(\mathbf{x}_0) \approx \mathbf{x},$$

and the number of bits needed for representing $\mathcal{T}$ is smaller than that of $\mathbf{x}$. The Contraction Mapping Theorem[4] provides a sufficient condition for convergence of (1).

In practice, linear transformations are usually used, so $\mathcal{T}$ may be represented by matrices $\mathbf{A}$ and $\mathbf{B}$ such that

$$\mathcal{T}(\mathbf{x}) = \mathbf{A}\mathbf{x} + \mathbf{B}$$

and (1) may be written as

$$\mathbf{x}_{n+1} = \mathbf{A}\mathbf{x}_n + \mathbf{B}.$$

In fractal image compression, each signal is typically partitioned into K range blocks $\mathbf{x}_i$, $i = 1, 2, \ldots, K$, each of size M, i.e.,

$$\begin{aligned} N &= KM, \\ \mathbf{x} &= \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \vdots \\ \mathbf{x}_K \end{bmatrix}, \\ \mathbf{x}_i &= \begin{bmatrix} x_{(i-1)M+1} \\ x_{(i-1)M+2} \\ \vdots \\ x_{(i-1)M+M} \end{bmatrix}, \end{aligned}$$

and for each $\mathbf{x}_i$, a domain block $\mathbf{d}_i$ of size LM (with $L > 1$) and an operator $\mathcal{T}_i : R^{LM} \mapsto R^M$ is found such that

$$\mathbf{x}_i \approx \mathcal{T}_i(\mathbf{d}_i) = \alpha_i \mathbf{P}_i \mathbf{G} \mathbf{d}_i + \mathbf{B}_i, \qquad i = 1, 2, \ldots, K.$$

$\mathbf{B}_i$ is a shift in grayscale

$$\mathbf{B}_i = \begin{bmatrix} b_i \\ b_i \\ \vdots \\ b_i \end{bmatrix}_{M \times 1},$$

and α_i is a scalar factor. $\mathbf{P}_i$ is an $M \times M$ matrix generated by a permutation of the rows of the identity matrix. It corresponds to a shuffling of the elements of the domain blocks, and in the 2-D case can, among other things, apply rotation of blocks by multiples of 90 degrees, and/or reflection against vertical, horizontal, or diagonal axis.

$\mathbf{G}$ is a *shrinking* matrix, which in the 1-D case typically has the form

$$\mathbf{G} = \begin{bmatrix} \mathbf{w} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{w} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{w} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{w} \end{bmatrix}_{M \times LM},$$

where

$$\mathbf{w} = \begin{bmatrix} \frac{1}{L} & \frac{1}{L} & \cdots & \frac{1}{L} \end{bmatrix}_{1 \times L}.$$

However, $\mathbf{G}$ may be generalized to represent decimation, i.e., lowpass filtering followed by subsampling, in which case it may be written as

$$\mathbf{G}_{M \times LM} = \mathbf{S}_{M \times LM} \mathbf{F}_{LM \times LM}$$

where $\mathbf{F}$ is the lowpass filtering operator and $\mathbf{S}$ is the subsampling matrix and may be written as:

$$\mathbf{S} = \begin{bmatrix} \mathbf{s} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{s} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{s} \end{bmatrix}_{M \times LM},$$

where

$$\mathbf{s} = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \end{bmatrix}_{1 \times L}.$$

In the 2-D case, $\mathbf{F}$ should represent a 2-D filtering operator and $\mathbf{S}$ should represent a 2-D subsampling by a factor L. The operator $\mathcal{T}_i$ transforms a domain vector $\mathbf{d}_i$ to an approximation of a range vector $\mathbf{x}_i$. All the operators $\mathcal{T}_i$, $i = 1, 2, \ldots, K$, together define a transformation $\mathcal{T}$ for $\mathbf{x}$ such that

$$\mathbf{x} \approx \mathcal{T}(\mathbf{x}) \tag{2}$$

where

$$\begin{aligned}
\mathcal{T}(\mathbf{x}) &= \sum_{i=1}^{K} \mathbf{H}_i \left(\mathcal{T}_i \left(\mathbf{d}_i \right) \right) \\
&= \sum_{i=1}^{K} \mathbf{H}_i \left(\left(\alpha_i \mathbf{P}_i \mathbf{G} \mathbf{d}_i \right) + \mathbf{B}_i \right) \\
&= \sum_{i=1}^{K} \alpha_i \mathbf{H}_i \mathbf{P}_i \mathbf{G} \mathbf{d}_i + \sum_{i=1}^{K} \mathbf{H}_i \mathbf{B}_i \\
&= \left(\sum_{i=1}^{K} \alpha_i \mathbf{H}_i \mathbf{P}_i \mathbf{G} \mathbf{K}_i \right) \mathbf{x} + \left(\sum_{i=1}^{K} \mathbf{H}_i \mathbf{B}_i \right) . \qquad (3)
\end{aligned}$$

$\mathbf{K}_i$, $i = 1, 2, \ldots, K$ are *fetch* operators, generating $\mathbf{d}_i$ from $\mathbf{x}$, i.e., $\mathbf{d}_i = \mathbf{K}_i \mathbf{x}$. If we denote the index of the first element of the domain block corresponding to $\mathbf{x}_i$ by I_i, then

$$\mathbf{K}_i = \left[\begin{array}{ccc} \mathbf{0}_{LM \times (I_i - 1)} & \mathbf{I}_{LM \times LM} & \mathbf{0}_{LM \times (N - (I_i - 1) - LM)} \end{array} \right]_{LM \times N} .$$

$\mathbf{H}_i$, $i = 1, 2, \ldots, K$ are *put* operators generating a component of $\mathbf{x}$ from $\mathbf{x}_i$, i.e.,

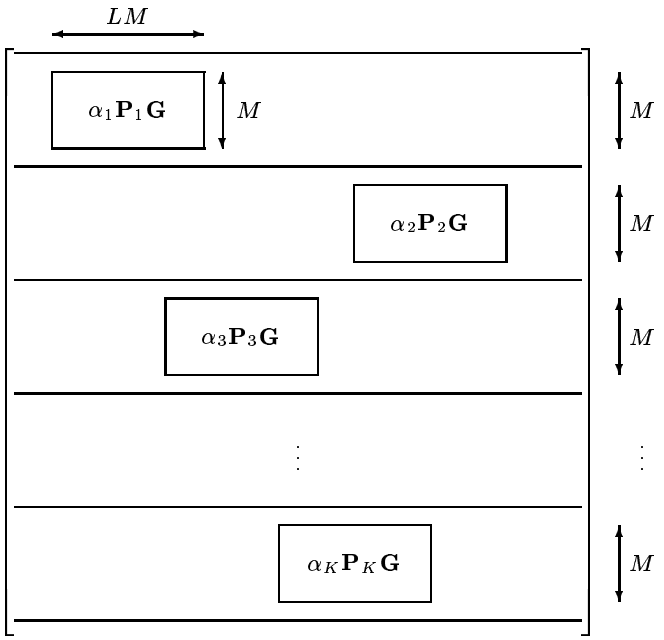
$$\begin{aligned}
\mathbf{H}_i \mathbf{x}_i &= \left[\begin{array}{c} \mathbf{0}_{(i-1)M \times 1} \\ \mathbf{x}_i \\ \mathbf{0}_{(K-i)M \times 1} \end{array} \right]_{N \times 1} , \\
\mathbf{H}_i &= \left[\begin{array}{c} \mathbf{0}_{(i-1)M \times M} \\ \mathbf{I}_{M \times M} \\ \mathbf{0}_{(K-i)M \times M} \end{array} \right]_{N \times M} .
\end{aligned}$$

Note that (3) has the form

$$\mathcal{T}(\mathbf{x}) = \mathbf{A}_{N \times N} \mathbf{x} + \mathbf{B}_{N \times 1} \qquad (4)$$

where $\mathbf{A}$ is of the form

$\mathbf{A} =$



and the horizontal location of each matrix $\alpha_i \mathbf{P}_i \mathbf{G}$ is determined by I_i. $\mathbf{B}$ has the form

$$\mathbf{B} = \left[\begin{array}{c} \mathbf{B}_1 \\ \mathbf{B}_2 \\ \vdots \\ \mathbf{B}_K \end{array} \right]_{N \times 1} .$$

In fractal coders, the matrices $\mathbf{G}$ and $\mathbf{H}_i$, $i = 1\ ,\ 2, \ldots, K$ are typically independent of $\mathbf{x}$ and are fixed both at the encoder and the decoder, and are not part of the code. But matrices $\mathbf{B}_i$, $\mathbf{K}_i$, $\mathbf{P}_i$ and values α_i, for $i = 1,\ 2, \ldots, K$, need to be specified for the decoder through the code.

Due to the structure of $\mathbf{B}_i$, it can be specified by sending b_i, $i = 1,\ 2, \ldots, K$, i.e., for each range block specify the offset of the mean of the range block with respect to the mean of the domain block. To specify $\mathbf{K}_i$, one needs only to specify I_i, $i = 1,\ 2, \ldots, K$, i.e., for each range block specify the address of its domain block. $\mathbf{P}_i$ typically represents one of a few types of permutations, and one needs to specify which one is used for each range block. Many researchers use only $\mathbf{P}_i = \mathbf{I}$, and do not need to specify it in the code. α_i is quantized and included in the code.

So the information needed to fully define $\mathcal{T}$ for the decoder, are $\{(\alpha_i,\ b_i,\ I_i,\ \mathbf{P}_i),\ \ i = 1,\ 2, \ldots, K\}$.

3 PREDICTION OF HIGHER RESOLUTION INFORMATION FROM LOWER RESOLUTION

Multiresolution analysis has many applications in image coding and analysis. Some wavelet-based multiresolution image compression methods, like zerotree,[5] implement some type of prediction of higher resolution information from lower resolution information. Fractal coders also do this type of prediction, though in a different way.[6,7]

The fractal code generated by fractal encoding of a digital image describes relationships (in the form of affine functions) between various segments of the image and is independent of the resolution of the original image. In other words, the fractal code is a *resolution independent* representation of the image and theoretically represents an approximation of the original image inthe continious-space domain. A decoder may decode this code to generate a digital image at any resolution. The resolution of the decoded image may be higher than the resolution of the original image. This increase of resolution is sometimes referred to as *fractal zoom*. This concept must be distinguished from the *fractal interpolation* that is studied in the mathematical literature.

The higher resolution obtained is not created by a simplistic technique such as repeating the pixels of the image, but by actually generating more detail. In fact, the additional higher resolution information is generated using information from the lower resolution image. When an image is reconstructed at the same resolution as the originally encoded image, in the decoding process the domain blocks of the image are shrunk (lowpass filtering followed by subsampling), which eliminates some of the details of the domain blocks. However, if the image is reconstructed at a higher resolution, in the shrinking of the domain block, the details of the domain block are only shrunk to generate the extra resolution in the range block. In fact, details of the domain blocks are used for missing details of the range block. The details in the domain block are also generated to some extent from details of other domain blocks, used for encoding each part of it. In other words, it is implicitly assumed that if the range block is similar to its corresponding domain block, then the details of the range block (which are *beyond* the resolution of the originally encoded image) are also similar to the details of the domain block (which are *within* the resolution of the encoded image). This principle is demonstrated in Figure 2. This assumption is a typical property of self-similarity of fractal sets at different scales, and the resolution independence is a property of the code generated by fractal-based methods.

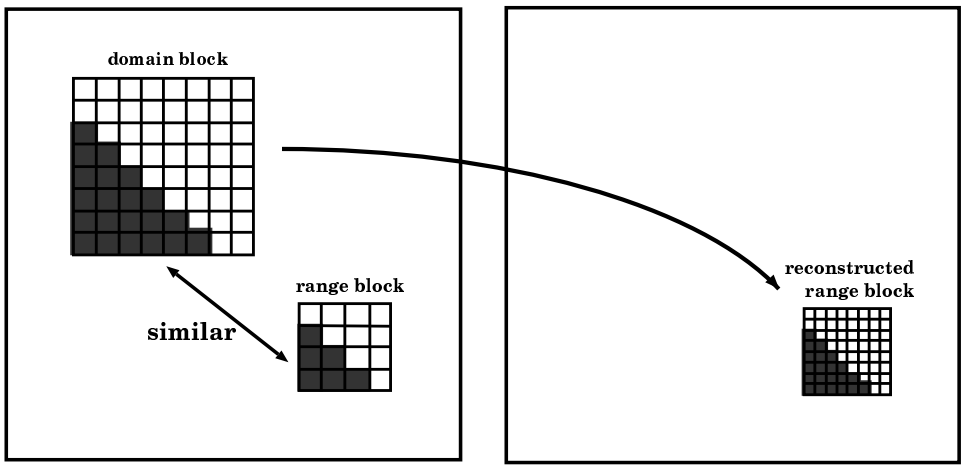
For fractal-based resolution enhancement, an $N_1 \times N_2$ image is first coded using the fractal method to generate a code. Then the decoder generates an $sN_1 \times sN_2$ image using this code. This process is shown in Figure 3. In this system, the encoder part is that of a fractal image compression system. However, the decoder needs special provisions. The decoder has a priori knowledge of $\mathbf{G}, \mathbf{H}_i$, $i = 1,\ 2, \ldots, K$, and the code is made up of information that specifies $\{\alpha_i, b_i, I_i, \mathbf{P}_i, i = 1,\ 2, \ldots, K\}$. This information specifies the transformation $\mathcal{T} : R^N \mapsto R^N$. The decoder uses the information to construct another transformation $\bar{\mathcal{T}} : R^{sN} \mapsto R^{sN}$, where $\bar{\mathcal{T}}$ can be written similar to (3) and (4) as

$$\bar{\mathcal{T}} = \bar{\mathbf{A}}_{sN\times sN}\mathbf{x} + \bar{\mathbf{B}}_{sN\times 1} \tag{5}$$

$$= \left(\sum_{i=1}^{K} \bar{\alpha_i}\bar{\mathbf{H}}_i\bar{\mathbf{P}}_i\bar{\mathbf{G}}\bar{\mathbf{K}}_i\right)\mathbf{x} + \left(\sum_{i=1}^{K} \bar{\mathbf{H}}_i\bar{\mathbf{B}}_i\right) \tag{6}$$

where

$$\bar{\alpha}_i = \alpha_i$$



(a) Low resolution image (b) High resolution image

Figure 2: A demonstration of a simplified version of the resolution enhancement method: (a) approximate each range block in the original image with a decimated domain block in the same image, and (b) use the domain block in place of the range block for a higher resolution.

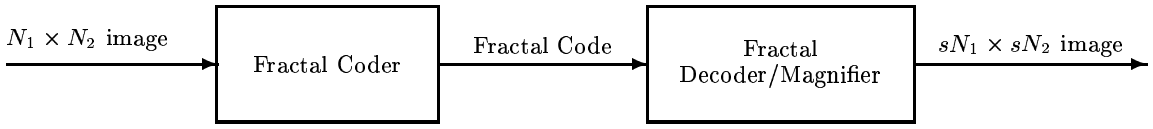


Figure 3: Block-diagram of a fractal-based resolution enhancer

$$
\begin{aligned}
\bar{\mathbf{H}}_i &= \begin{bmatrix} \mathbf{0}_{s(i-1)M \times sM} \\ \mathbf{I}_{sM \times sM} \\ \mathbf{0}_{s(K-i)M \times sM} \end{bmatrix}_{sN \times sM}, \\
\bar{\mathbf{G}} &= \begin{bmatrix} \bar{\mathbf{w}} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \bar{\mathbf{w}} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \bar{\mathbf{w}} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \bar{\mathbf{w}} \end{bmatrix}_{sM \times sLM}, \\
\bar{\mathbf{w}} &= \begin{bmatrix} \frac{1}{sL} & \frac{1}{sL} & \cdots & \frac{1}{sL} \end{bmatrix}_{1 \times sL}, \\
\bar{I}_i &= s(I_i - 1) + 1, \\
\bar{\mathbf{K}}_i &= \begin{bmatrix} \mathbf{0}_{sLM \times (\bar{I}_i - 1)} & \mathbf{I}_{sLM \times sLM} & \mathbf{0}_{sLM \times (sN - (\bar{I}_i - 1) - sLM)} \end{bmatrix}_{sLM \times sN}, \\
\bar{\mathbf{B}}_i &= \begin{bmatrix} b_i \\ b_i \\ \vdots \\ b_i \end{bmatrix}_{sM \times 1}.
\end{aligned}
$$

Regarding $\bar{\mathbf{P}}_i$, it must represent the same operation that $\mathbf{P}_i$ does, but for a $sM \times sM$ block. In the 1-D case, for example,

$$\mathbf{P}_i = \mathbf{I}_{M \times M}$$

gives

$$\bar{\mathbf{P}}_i = \mathbf{I}_{sM \times sM},$$

and

$$\mathbf{P}_i = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 1 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 1 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix}_{M \times M}$$

gives

$$\bar{\mathbf{P}}_i = \begin{bmatrix} 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 1 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 1 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \end{bmatrix}_{sM \times sM}.$$

It can be shown that decimating $\bar{\mathbf{x}}_\infty$ by applying an averaging filter of length s followed by subsampling converts $\bar{\mathbf{x}}_\infty$ to $\mathbf{x}_\infty$. The operator for this decimation has the form

$$\mathbf{V} = \begin{bmatrix} \mathbf{v} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{v} & \cdots & \mathbf{0} & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{v} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{v} \end{bmatrix}_{N \times sN},$$

$$\mathbf{v} = \begin{bmatrix} \frac{1}{s} & \frac{1}{s} & \cdots & \frac{1}{s} \end{bmatrix}_{1 \times s}.$$

In other words

$$\mathbf{V}\bar{\mathbf{x}}_\infty = \mathbf{x}_\infty.$$

4 PROPOSED METHOD

There have been many publications on fractal image coding, but there have not been any published studies on the resolution enhancement feature of the fractal coders. Given a digital image, in order to evaluate the quality of an interpolated version of this image, we need to know what the larger image would be if it was obtained directly by sampling of a continuous-space image at the higher rate of the larger image. Therefore, we first take an image O_n of size $n \times n$, assuming it is the real larger image. We then decimate (shrink) it using the classical method of lowpass filtering (to avoid aliasing) followed by subsampling to obtain the smaller $m \times m$ image. Note that the decimation process removes the high-frequency components of the larger image. This decimation may be done using ideal lowpass filters, for example, in the DCT domain or in the Haar domain. In the DCT case, we denote the resulting image by $C_{m \leftarrow n} O_n$, and in the Haar case by $H_{m \leftarrow n} O_n$. We then take $C_{m \leftarrow n} O_n$ or $H_{m \leftarrow n} O_n$ as the initial image, encode it using a fractal coder to obtain a fractal code, and decode the code using the fractal decoder/magnifier back to size $n \times n$. We denote the enlarged images by $F_{n \leftarrow m} C_{m \leftarrow n} O_n$ and $F_{n \leftarrow m} H_{m \leftarrow n} O_n$.

We can also enlarge $C_{m \leftarrow n} O_n$ and $H_{m \leftarrow n} O_n$ by classical interpolation where we denote the results by $C_{n \leftarrow m} C_{m \leftarrow n} O_n$ and $H_{n \leftarrow m} H_{m \leftarrow n} O_n$. Note that $C_{n \leftarrow m} C_{m \leftarrow n} O_n$ is an ideally lowpassed version of O_n in the DCT domain, and $H_{n \leftarrow m} H_{m \leftarrow n} O_n$ is an ideally lowpassed version of O_n in the Haar domain. We may then compare O_n with $F_{n \leftarrow m} C_{m \leftarrow n} O_n$ and $C_{n \leftarrow m} C_{m \leftarrow n} O_n$ at different DCT frequency bands, or compare O_n with $F_{n \leftarrow m} H_{m \leftarrow n} O_n$ and $H_{n \leftarrow m} H_{m \leftarrow n} O_n$ at different Haar wavelet bands. Note that $C_{n \leftarrow m} C_{m \leftarrow n} O_n$ and O_n are exactly the same at low DCT frequencies and at high frequency bands $C_{n \leftarrow m} C_{m \leftarrow n} O_n$ is all zero, and similarly for the Haar case. However, this is not true between $F_{n \leftarrow m} C_{m \leftarrow n} O_n$ and O_n or between $F_{n \leftarrow m} H_{m \leftarrow n} O_n$ and O_n.

With a coding application in mind, the final result of the final interpolation is $F_{n \leftarrow m} C_{m \leftarrow n} O_n$ or $F_{n \leftarrow m} H_{m \leftarrow n} O_n$. The fractal coding is typically lossy, which means that in contrast to $C_{n \leftarrow m} C_{m \leftarrow n} O_n$ and $H_{n \leftarrow m} H_{m \leftarrow n} O_n$, which have exactly the same low frequency components as in O_n, the low-frequency components of the $F_{n \leftarrow m} C_{m \leftarrow n} O_n$ and $F_{n \leftarrow m} H_{m \leftarrow n} O_n$ are different from that of O_n.[1] This means that decimation of $F_{n \leftarrow m} C_{m \leftarrow n} O_n$ or $F_{n \leftarrow m} H_{m \leftarrow n} O_n$ does not result in $C_{m \leftarrow n} O_n$ and $H_{m \leftarrow n} O_n$.

[1] However, for the typical fractal coders, the Haar transform has the property that low-frequency components of $F_{n \leftarrow m} H_{m \leftarrow n} O_n$ are the same as the whole frequency components of $F_{m \leftarrow m} H_{m \leftarrow n} O_n$, but this is not true in DCT domain.

Table 1: Rms of the Original 512 × 512 image O_{512} and the rms error of its approximations obtained from interpolation by a factor of 2 (i.e. rms of $O_{512} - H_{512\leftarrow 256}H_{256\leftarrow 512}O_{512}$ and $O_{512} - F_{512\leftarrow 256}H_{256\leftarrow 512}O_{512}$), and by a factor of 4 (i.e., rms of $O_{512} - H_{512\leftarrow 128}H_{128\leftarrow 512}O_{512}$ and $O_{512} - F_{512\leftarrow 128}H_{128\leftarrow 512}O_{512}$) at different Haar frequency bands.

			a_0	d_1	v_1	h_1	a_1	d_2	v_2	h_2	a_2	d_3	v_3	h_3	a_3
rms of original O_{512}			133.	4.0	10.6	7.3	266.	12.1	29.6	19.1	530.	32.9	77.7	48.2	1055.
Factor 2	rms error	classical	6.7	4.0	10.6	7.3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		G&H	6.8	4.9	9.6	6.4	5.3	5.0	6.0	5.1	5.2	5.5	6.2	5.2	3.3
		ISI	6.3	3.8	7.7	5.3	7.5	5.1	7.7	5.0	10.6	7.8	8.7	6.0	16.6
		Fisher	6.9	4.5	8.7	6.3	7.5	7.3	8.4	7.6	6.7	7.8	7.3	7.3	3.4
		Fisher pp	6.7	4.4	8.2	5.9	7.5	7.0	8.2	7.2	7.3	8.0	7.8	7.3	6.0
		N&G	8.2	4.4	9.2	6.3	11.1	9.5	12.4	10.5	11.9	11.5	10.3	11.3	14.3
Factor 4	rms error	classical	11.5	4.0	10.6	7.3	18.6	12.1	29.6	19.1	0.0	0.0	0.0	0.0	0.0
		ISI	9.6	4.2	8.9	6.3	15.2	11.6	22.4	14.7	8.1	5.5	5.9	5.8	12.7
		Fisher	12.1	4.8	10.6	7.7	19.8	13.0	25.3	17.1	21.5	21.0	24.4	21.0	19.5
		Fisher pp	11.8	4.8	10.3	7.5	19.2	12.8	24.3	16.4	21.2	20.7	23.9	20.4	19.6
		N&G	13.7	4.6	11.0	7.5	23.5	13.1	26.9	17.4	31.9	27.8	35.4	30.1	33.9

Table 2: Rms of the Original 512 × 512 image O_{512} and the rms error of its approximations obtained from interpolation by a factor of 2 (i.e. rms of $O_{512} - C_{512\leftarrow 256}C_{256\leftarrow 512}O_{512}$ and $O_{512} - F_{512\leftarrow 256}C_{256\leftarrow 512}O_{512}$), and by a factor of 4 (i.e., rms of $O_{512} - C_{512\leftarrow 128}C_{128\leftarrow 512}O_{512}$ and $O_{512} - F_{512\leftarrow 128}C_{128\leftarrow 512}O_{512}$) at different DCT frequency bands.

			a_0	d_1	v_1	h_1	a_1	d_2	v_2	h_2	a_2	d_3	v_3	h_3	a_3
rms of original O_{512}			133.	3.0	3.9	6.4	266.	11.4	13.2	22.5	531.	35.8	34.8	63.7	1059.
Factor 2	rms error	classical	4.0	3.0	3.9	6.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
		G&H	7.2	5.1	6.5	9.6	7.0	6.9	6.3	8.4	6.3	6.5	6.1	6.7	5.7
		ISI	5.9	3.4	4.3	6.8	8.2	6.6	5.5	9.2	10.6	8.1	5.1	8.3	17.0
		Fisher	7.4	4.4	6.3	9.0	8.9	8.5	8.6	10.2	8.0	9.2	8.1	8.8	5.4
		N&G	8.4	4.2	6.1	8.8	12.4	10.2	10.9	13.6	14.2	14.1	12.4	13.0	16.9
Factor 4	rms error	classical	11.5	4.0	10.6	7.3	18.6	12.1	29.6	19.1	0.0	0.0	0.0	0.0	0.0
		ISI	10.1	4.6	4.9	7.2	17.6	12.3	16.3	26.7	10.4	10.9	6.7	9.5	13.3
		Fisher	13.0	4.7	7.2	10.2	22.3	14.5	17.6	27.8	26.3	25.5	23.7	31.7	23.5
		N&G	14.3	4.2	6.6	10.0	25.7	14.0	17.3	28.4	36.6	32.8	31.2	41.2	40.3

However, when an interpolation-only application is in mind, the fractal coder may still be used without loss in low-frequency components by replacing the low-frequency components of $F_{n\leftarrow m}C_{m\leftarrow n}O_n$ or $F_{n\leftarrow m}H_{m\leftarrow n}O_n$ with frequency components of $C_{m\leftarrow n}O_n$ and $H_{m\leftarrow n}O_n$, which is the frequency decomposition of the smaller image that is known by the interpolator. However, this method cannot be used in coding applications as the $C_{m\leftarrow n}O_n$ and $H_{m\leftarrow n}O_n$ are not available to the decoder.

5 EXPERIMENTAL RESULTS

The 512 × 512 test image Lena was used as O_n with $n = 512$. Then $C_{m\leftarrow n}O_n$, $H_{m\leftarrow n}O_n$, $C_{n\leftarrow m}C_{m\leftarrow n}O_n$, and $H_{n\leftarrow m}H_{m\leftarrow n}O_n$ were computed for $m = 256$ and $m = 128$. Four different fractal coding algorithms[8–11] were used and for each of them $F_{n\leftarrow m}C_{m\leftarrow n}O_n$ and $F_{n\leftarrow m}H_{m\leftarrow n}O_n$ were computed. The resulting images were then transformed into frequency domain (DCT domain for C cases and Haar wavelet domain for H cases). The rms error at different frequency bands between the results and O_n were computed. These results are shown in Tables 1 and 2. The frequency bands a_i, v_i, h_i, and d_i are shown in Figure 4. In these tables, the "rms of original" is rms of the original O_{512} image signal, while the rms error of the methods is the rms of the difference between the original image and the interpolated images. The method "G&H" refers to,[8] "Fisher" refers to[10] ("pp" stands for "with postprocessing"), "ISI" refers to,[9] and "N&G" refers to.[11] The original image O_{512}, and the interpolated image by classical method $C_{512\leftarrow 256}C_{256\leftarrow 512}O_{512}$ are shown in Figure 5. The fractally interpolated images $F_{512\leftarrow 256}C_{256\leftarrow 512}O_{512}$ from the 256 × 256 image $C_{256\leftarrow 512}O_{512}$ using four

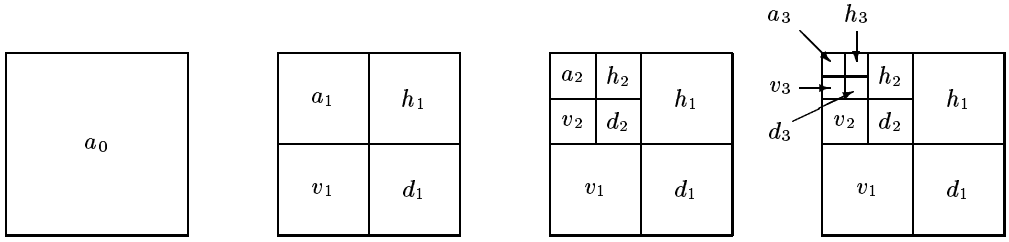


Figure 4: Frequency bands of Tables 1 and 2

different fractal coding methods are shown in Figure 6. These images after subband replacement from the $C_{256\leftarrow 512}O_{512}$ image are shown in Figure 7.

The results show that for all the fractal enhancement methods, the energies of error in the predicted bands (bands d_1, v_1, and h_1, for the $F_{512\leftarrow 256}C_{256\leftarrow 512}O_{512}$ and $F_{512\leftarrow 256}H_{256\leftarrow 512}O_{512}$images, and bands d_1, v_1, h_1, d_2, v_2, and h_2 for the $F_{512\leftarrow 128}C_{128\leftarrow 512}O_{512}$, and $F_{512\leftarrow 128}C_{128\leftarrow 512}O_{512}$ images) are nearly as strong as the energies of the original O_{512} signal in those bands. However, by definition, this is also true for the classical interpolation methods.

The prediction of the high-frequency bands by the $F_{512\leftarrow m}H_{m\leftarrow 512}O_{512}$ is slightly better than $H_{512\leftarrow m}H_{m\leftarrow 512}O_{512}$ for the case of $m = 256$, but worse for $m = 128$. In the case of DCT domain, $F_{512\leftarrow m}C_{m\leftarrow 512}O_{512}$ is slightly worse than $C_{512\leftarrow m}C_{m\leftarrow 512}O_{512}$ for both $m = 256$, but clearly worse for $m = 128$.

In the Haar domain, the results suggest that prediction of high frequency bands from one band lower in frequency is much better than prediction from several bands lower. This implies that blocks in frequency bands are more similar in closer bands and suggests that a better fractal resolution enhancer may be obtained if blocks in high-frequency bands are approximated by blocks in only one band above them. This is in contrast to matching blocks of different sizes in spatial domain which corresponds to matching trees of blocks in the Haar domain.

6 CONCLUSIONS

The resolution enhancement feature of fractal coders is analyzed and evaluated against the classical interpolation method using four different fractal coding methods. In the Haar wavelet domain, the fractal enhancement shows slightly better results when change of scale is by a factor of 2. However, in other cases, the classical interpolation method performs better. Our study suggests a new type of fractal coder operating directly in the frequency domain.

7 ACKNOWLEDGEMENTS

This work was supported in part by Army Research Laboratory under Cooperative Agreement No. DAAL01-96-2-0003, and in part by a Grant from Mitsubishi Electric Inc.

8 REFERENCES

[1] A. V. Oppenheim and R. W. Schafer, *Discrete-Time Signal Processing.* Englewood Cliffs, NJ: Prentice Hall, 1989.

[2] A. E. Jacquin, *A Fractal Theory of Iterated Markov Operators with Applications to Digital Image Coding.* PhD thesis, Georgia Institute of Technology, Atlanta, GA, Aug. 1989.

[3] M. F. Barnsley and L. P. Hurd, *Fractal Image Compression.* Wellesley, MA: AK Peters, Ltd., 1993.

[4] M. F. Barnsley, *Fractals Everywhere.* San Diego, CA: Academic Press, Inc., 1988.

O_{512} $C_{512\leftarrow 256}C_{256\leftarrow 512}O_{512}$

Figure 5: Original 512×512 Lena image O_{512}, and classically interpolated image $C_{512\leftarrow 256}C_{256\leftarrow 512}O_{512}$.

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

[6] R. Rinaldo and G. Calvagno, "An image coding scheme using block prediction of the pyramid subband decomposition," in *Proceedings of IEEE International Conference on Image Processing*, vol. 2, (Austin, TX), pp. 878–882, Nov. 13–16, 1994.

[7] G. Davis, "Self-quantized wavelet subtrees: A wavelet-based theory for fractal image compression," in *DCC'95: Data Compression Conference*, (Snowbird, UT), Mar. 28–30, 1995.

[8] M. Gharavi-Alkhansari and T. S. Huang, "Fractal image coding using rate-distortion optimized matching pursuit," in *Proceedings of the SPIE, Visual Communications and Image Processing*, vol. 2727, (Orlando, FL), pp. 1386–1393, Mar. 17–20, 1996.

[9] Iterated Systems, Inc., "Fractal Imager." Coding software available from http://www.iterated.com.

[10] Y. Fisher, "Fractal image compression with quadtrees," in *Fractal Image Compression: Theory and Application* (Y. Fisher, ed.), pp. 55–77, New York: Springer-Verlag, 1995.

[11] M. Nelson and J.-L. Gailly, "Fractal image compression," in *The Data Compression Book*, ch. 13, pp. 457–510, New York: M&T Books, 2nd ed., 1996.

G&H

ISI

Fisher

N&G

Figure 6: $F_{512\leftarrow 256}C_{256\leftarrow 512}O_{512}$: 512×512 images interpolated from the 256×256 image, using four different fractal methods.

G&H

ISI

Fisher

N&G

Figure 7: Subband replaced $F_{512\leftarrow 256}C_{256\leftarrow 512}O_{512}$: 512×512 images interpolated from the 256×256 image, using four different fractal methods, with subband replacement.