A FAST ENCODING METHOD WITHOUT SEARCH FOR FRACTAL IMAGE COMPRESSION

In Kwon Kim, Rae-Hong Park

Dept. of EE, Sogang Univ., C.P.O. Box 1142, Seoul 100-611, Korea

ABSTRACT

A fast coding algorithm for images using vector quantization (VQ) and pixelwise fractal approximation is proposed. The low frequency component of an input image is approximated and its residual is used to calculate the scaling factor of fractal transform. The scaling factor is compressed by transform VQ (TVQ). In the proposed method, to encode a digital image by an iterated function system (IFS), we use the pixel-based IFS (PIFS) rather than the block-based IFS: the scaling factor is computed for each pixel. In the proposed method, the scaling factor of each pixel is calculated with the constraint of contraction mapping and it is transformed by wavelet and quantized by VQ. For approximation of an original image, the variable block-size segmentation using quadtree is employed. Because the proposed method calculates the scaling factor using the PIFS, the encoding time is faster than the conventional algorithm using block-based IFS with search.

1. INTRODUCTION

Since Barnsley [1] first presented the idea of fractal image coding, various fractal image coding approaches have been developed [2]-[4]. Because the quality of images reconstructed by a fractal coding algorithm depends on the way of describing self-similarity [1], various research has been conducted to effectively present the concept of self-similarity existing in an image. To represent the self-similarity, the transformation between two image blocks should be described.

Algorithms using an iterated function system (IFS) [1, 2] are generally divided into two categories depending on the fractal model employed. One is a self-transformation system (STS) which maps a block to each subblock of it [3]. The other is a piecewise transformation system (PTS) which maps a block of an image to other block [2].

Since Jacquin [2] first introduced the practical image coding algorithm based on the PTS, the PTS has been extensively studied for encoding of natural images. Coding algorithms based on the STS were also presented [3], where a partitioned block rather than the whole image was used as an entire function. In Jacquin's method, the computational requirement for calculating transform coefficients and for determining a domain block is high. Whereas the computational complexity of Monro's method [3] is much lower than that of Jacquin's. Monro's method showed worse performance because it searched in the limited region.

The PTS is more flexible than the STS. However the PTS requires extensive search time to find a domain block that shows the minimum distortion. In this paper, a fractal coding algorithm using the pixel-based IFS is proposed. In the proposed algorithm, the scaling factor of each pixel is calculated with the constraint. To calculate the scaling factor, the input image is approximated using quadtree segmentation, then an input image and its residual image are used to calculate the scaling factor of each pixel. Fractal approximation which uses the self-similarity between a pixel and its neighbor pixels is employed.

The rest of the paper is organized as follows. In Section 2 the proposed pixel-based IFS fractal coding method is described. In Section 3 we then discuss the parameter quantization and explain the proposed quantization method based on wavelet transformation and vector quantization (VQ). In Section 4, computer simulation results of the proposed method are shown for the 512×512 Lena and Boat images. Finally, the properties of the proposed method and future works are presented.

2. PROPOSED PIXEL-BASED FRACTAL CODING METHOD

The proposed algorithm doesn't need the search process to select the optimal parameters. In the conventional block-based methods, the image A is divided into a number of $B \times B$ range blocks and is encoded by the IFS W defined as

$$W(A) = \bigcup_{n=1}^{N} w_n(A) \qquad (1)$$

where N denotes the number of blocks and w_n represents the nth contraction mapping with the contractivity s_n. In

This work was supported in part by the Ministry of Information and Communication, Korea.

general, the following form of

$$w_n = \alpha_n(D_n(i,j) - \bar{D}_n) + \bar{R}_n \tag{2}$$

is adopted for w_n, where α_n and $\bar{R}_n$ represent the scaling factor and the mean value of the nth range block, respectively. D_n and D_n denote the resized domain block corresponding to the nth range block R_n and its mean value, respectively. The size of the domain block is set to double the size of the range block. α_n is selected so that it minimizes the distortion function e_n defined by

$$e_n = \sum_{i=0}^{B-1}\sum_{j=0}^{B-1}\{R_n(i,j) - \alpha_n(D_n(i,j) - \bar{D}_n) - \bar{R}_n\}^2. \tag{3}$$

Fig. 1 shows the block diagram of the proposed method. First, an input image is segmented by quadtree and each block is approximated by its mean value. Then, the scaling factor is calculated using the domain and the residual image. The domain image is generated by 3×3 mean filtering of an input image. The scaling factor is segmented using quadtree and quantized by transformation VQ (TVQ) using wavelet. In the proposed pixel-based method, contraction mapping w_{ij} defined by

$$w_{ij} = \alpha_{ij} \cdot f_d(i,j) + \hat{f}(i,j) \tag{4}$$

is used, where w_{ij} represents the pixelwise contraction mapping for the (i,j) pixel of an input image and it must satisfy the contractivity to reconstruct the image at a receiver. f_d and $\hat{f}$ denote the pre-generated domain image and approximated image, respectively. Vector quantization (VQ) and the variable block size method are used to encode the scaling factor. α_{ij} is computed by

$$\alpha_{ij} = \frac{1}{f_d(i,j)}\{f(i,j) - \hat{f}(i,j)\} \tag{5}$$

where f represents the input image and $\hat{f}$ denotes its approximated version.

For contractivity, the constraint, $|\alpha_{ij}| < 1$, should be satisfied, i.e.,

$$|\hat{f}(i,j) - f(i,j)| < f_d(i,j) \tag{6}$$

where f_d, $\hat{f}$, and f have the dynamic range between 0 and $G-1$, with G denoting the maximum gray level. Thus α_{ij} is less than one except when the gray value of f_d is very small. For that case, α_{ij} is set to one to guarantee the convergence. In the proposed method, because $\alpha_{i,j}$ is calculated for each pixel, i.e., the optimal value is known for each pixel, the reconstructed image is degraded only by the quantization error. Thus the search process that spends the expensive encoding time is not employed in the proposed method, and the address of domain is not transmitted.

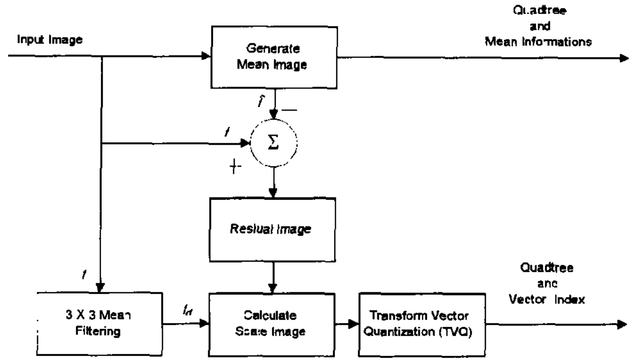


Figure 1: Block diagram of the proposed pixel-based IFS.

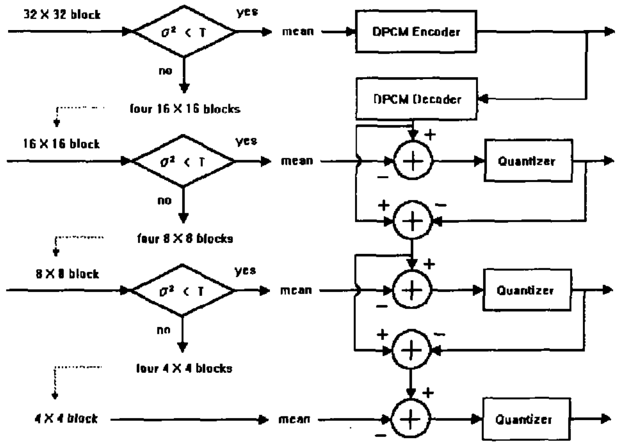
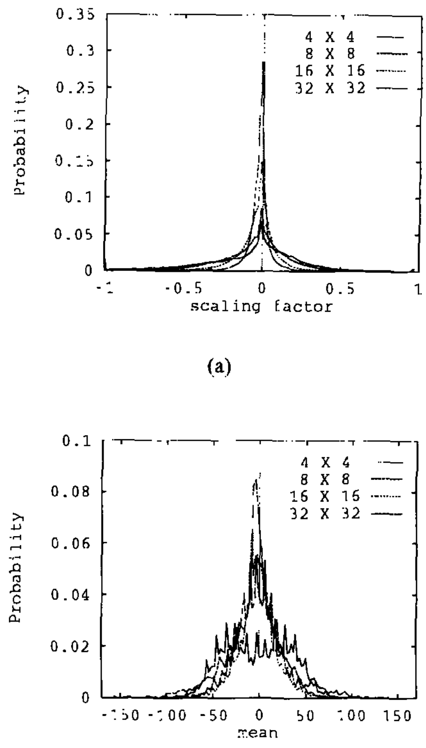


Figure 2: Flowchart of variable block size segmentation.

3. PARAMETER QUANTIZATION

The parameters α_{ij} and $\hat{f}$ are to be encoded. The proposed method consists of two stages: approximation and residual compensation. In the approximation stage, $\hat{f}$ is quantized and transmitted, and in the residual compensation stage, α_{ij} is quantized. To quantize $\hat{f}$, the variable block size segmentation method using quadtree [4],[5] is used. In the proposed method, the variance of blocks is used as the threshold value to determine whether or not we split the given block further. The largest block size is set to 32×32 and the mean values of 32×32 blocks are quantized by differential pulse code modulation (DPCM). For blocks with the size smaller than 32×32, the difference value between the mean value of the current block and that of the parent block is quantized, as shown in Fig. 2. Because those difference values of each block size have the different probabilities as shown in Fig. 3(a), different coders are used.

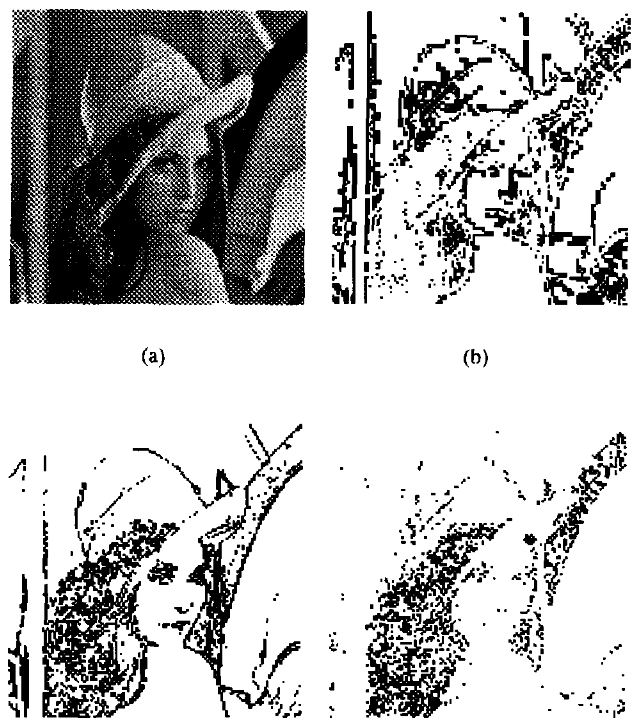
Next, the scaling factor $\alpha_{i,j}$ for the residual signal $f_d - \hat{f}$ is quantized. Because the scaling factor is selected for each



(a)

(b)

Figure 3: Distributions of $\hat{f}$ and α_{ij} for each block size. (a) $\hat{f}$, (b) α_{ij}.



(a) (b)

(c) (d)

Figure 4: Classification results based on the fBm for the Lena image. (a) Original Lena image, (b) shade region, (c) dynamic region, (d) texture region.

pixel, TVQ is employed for each block size to reduce the number of bits needed. Also, the quadtree segmentation used in the approximation stage is applied to quantize the scaling factor. Each segmented block is transformed by wavelet and quantized by VQ. Note that the dynamic range of the residual signal is smaller than that of f_d which is a part of an input image: most scaling factors are distributed around zero and different distribution patterns are generated for different block sizes. Fig. 3(b) shows the distribution patterns of the scaling factors for each block size.

The blocks larger than 4×4 are transformed by wavelet transformation, yielding 4×4 low-low (LL) bands. The transformed 4×4 blocks are quantized by a codebook. The 4×4 blocks are classified into three types such as shade, dynamic, and texture. Each block is quantized by VQ with shade, dynamic, and texture codebooks. To classify a block, the fractional Brownian motion (fBm) [6] is used, where the fBm is a two-dimensional process $X(i,j)$ with the properties:

1. The increments $X(i,j) - X(k,l)$ are Gaussian with zero mean.
2. The variance of the increments $X(i,j) - X(k,l)$ is proportional to the $2H$-th power of the distance between (i,j) and (k,l), where H satisfies $0 < H < 1$.

In the proposed method, the variance σ_1^2 and $\sigma_{\sqrt{2}}^2$ of the increments α_{ij} with distance 1 and $\sqrt{2}$ are calculated. The block types are classified by thresholding σ_1^2 and $\sigma_{\sqrt{2}}^2$. If σ_1^2 and $\sigma_{\sqrt{2}}^2$ are lower than the pre-determined threshold value, the block corresponds to shade type. Otherwise if $\sigma_1^2/\sigma_{\sqrt{2}}^2$ is close to 1, the block has the random texture type, otherwise dynamic type. Fig. 4 shows the classified 4×4 blocks of the Lena image by the proposed method.

4. SIMULATION RESULTS

512×512 Lena and Boat images, uniformly quantized to 8 bits, were used as test images. Three 720×480 and six 512×512 images were used for construction of the VQ codebook, in which the LBG algorithm was employed. Note that two test images are not included in the set of images used for codebook generation. The codebook indexes were coded with Huffman coding. The performance of the proposed

Table 1: PSNR and Bit Rate of the Proposed Pixel-Based IFS for Two Test Images.

	Block Size	Lena mean approx.	Lena final result	Boat mean approx.	Boat final result
PSNR (dB)	32×32	32.27	41.70	32.29	41.86
	16×16	27.81	31.30	26.75	31.11
	8×8	22.19	26.90	20.80	26.44
	4×4	18.51	27.82	16.74	29.77
	Total	24.74	29.75	22.08	28.37
Bit Rate (bpp)		0.070	0.167	0.096	0.236

method is evaluated in terms of the bit rate and the peak signal to noise ratio (PSNR). The PSNR is defined by

$$\mathrm{PSNR} = 10 \log_{10} \frac{255^2}{\frac{1}{IJ} \sum_{i=0}^{I-1} \sum_{j=0}^{J-1} \left[f(i,j) - \tilde{f}(i,j) \right]^2} \quad (7)$$

where $I \times J$ represents the size of the image, and $\tilde{f}(i,j)$ denotes the reconstructed image.

Table 1 shows the blockwise PSNR, total PSNR, and the bit rate of the proposed method, with 'mean approx.' and 'final result' representing the reconstructed images by $(\bar{f})$ and $(\alpha_{ij}, \bar{f})$, respectively. The PSNR increment of 32×32 blocks by the residual compensation stage with α_{ij} is higher than that of other block sizes, where the PSNR increment is defined by the difference between PSNR's of the reconstructed images $\tilde{f}$ (with residual compensation) and $\bar{f}$ (without residual compensation). 16×16 and 8×8 blocks show the lower PSNR increment. At low bit rates, the image reconstructed by JPEG shows noticeable artifact such as the blocking effect, but that by the proposed method yields less blocking effect. Because each pixel is updated and smoothed iteratively, the PIFS shows more natural reconstructed image than JPEG at low bit rates.

The proposed method does not require a search process and calculates the scaling factor by a simple equation, resulting in lower computational load compared to the conventional fractal coding methods. In the proposed method, if the computational complexity for scaling factor α_{ij} of each pixel is the same, the difference results from the search process. If the proposed method without the search process takes the $O(1)$ complexity, the conventional fractal coding requires $O(M^2)$ complexity with the $M \times M$ search region assumed.

5. CONCLUSIONS

This paper proposes a fast coding algorithm using the pixelwise IFS. Fractal approximation uses the shape information of gray surfaces, i.e., the IFS encodes an input image based on the idea of self-similarity. Thus, the existence of similar gray surface patterns is important to encode an image by fractal coding. In the proposed method, because the scaling factor is calculated pixelwise, an input image having the weak self-similarity is effectively encoded by fractal mapping. In addition, the proposed method can find optimal parameters without the search process. Thus it is faster than other methods requiring expensive search time.

Further research will focus on the investigation of the quantization problem, optimization, and applications of the PIFS to moving image sequences.

6. REFERENCES

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

[2] A. E. Jacquin, "Fractal image coding: A review," *Proc. IEEE*, vol. 81, pp. 1451-1465, Oct. 1993.

[3] D. M. Monro, "A hybrid fractal transform," in *Proc. Int. Conf. Acoust., Speech, Signal Processing'93*, vol. V, pp. 169-172, Minneapolis, MN, Apr. 1993.

[4] I. K. Kim and R.-H. Park, "Still image coding based on vector quantization and fractal approximation," *IEEE Trans. Image Processing*, vol. IP-5, pp. 587-597, Apr. 1996.

[5] J. Vaisey and A. Gersho, "Image compression with variable block size segmentation," *IEEE Trans. Signal Processing*, vol. SP-40, pp. 2040-2060, Aug. 1992.

[6] D. Saupe, "Algorithms for random fractals," in *The Science of Fractal Images*. Eds. H.-O. Peitgen and D. Saupe, New York, NY: Springer-Verlag, 1988.