

FRACTAL COLOR IMAGE COMPRESSION USING VECTOR DISTORTION MEASURE

Ying Zhang & Lai-Man Po

Department of Electronic Engineering
City University of Hong Kong
Kowloon Tong, Hong Kong

ABSTRACT

In this paper, fractal monochrome image compression technique proposed by Jacquin is investigated for 24-bit true color image by encoding the RGB components' images independently. To exploit the spectral redundancy in RGB components, the root mean square error distortion measure in grayscale space is extended to 3-dimensional color space for fractal-based color image coding. Experimental results show that 1.5 compression ratio improvement can be obtained using the vector distortion measure in fractal coding with fixed image partition as compared to separate fractal coding in RGB images. In addition, adaptive fractal color image coding based on quadtree partition is also proposed which can obtain a good trade-off between compression ratio and image fidelity.

1. INTRODUCTION

Since Jacquin introduced a practical block-based fractal image coding scheme [1] in 1990, fractal coding has aroused a great deal of attention [2,3,4] as a new promising image compression technique. This method is based on IFS (iterated function system) which exploits the self affine similarity between different parts of the image. Through affine transformation one sub-block of the original image can be approximated by another sub-block in the same image to some extent estimated by a distortion measure. If the approximation error is small enough, the reconstructed image will be very similar to the original image by iterative decoding process based on fractal transformation.

Fractal compression technique has been growing rapidly in recent years especially in monochrome image coding. While color image compression is required in most of the applications such as multimedia or photo storage. It is easily to apply monochrome fractal image coding technique in color image coding as a color image is usually represented in multi-channels such as R, G and B color components. Thus, each channel of the color image can be compressed as a gray-level image. However, such implementation need to repeat the same fractal image coding process multiple times and the spectral redundancy among channels cannot be exploited. In this paper, the 3-dimensional affine

transform for monochrome fractal image coding is extended to 5-dimensional affine transform for color image fractal coding in order to get a high compression ratio. The rest of this paper is organized as follows. Section 2 contains an overview of gray-level fractal image coding algorithm. In section 3, vector distortion measure for color pixel similarity measure in RGB space is first presented and then a generalization of fractal transform for 3-dimensional RGB color space is formulated for fractal color image compression. Adaptive fractal color image coding using vector distortion measure with quadtree partition for achieving high compression ratio and acceptable reconstructed image quality is presented in section 4. Experimental results and further consideration of the proposed fractal-based color image compression are discussed in section 5.

2. BASIC ALGORITHM FOR FRACTAL IMAGE CODING

In block-based fractal coding proposed by Jacquin, a digital monochrome image is first partitioned into non-overlapping square blocks R_k (range blocks) of size $B \times B$ and larger square blocks D_k (domain blocks) of size $D \times D$ (usually equal to $2B \times 2B$). The potential domain blocks can be obtained by sliding a $D \times D$ window across the original image with horizontal and vertical step size of Δh and Δv respectively, which typically equals to B or $B/2$. A search pool is formed by all the domain blocks. The major task of fractal image compression is to find the most suitable domain block in the search pool to approximate a range block through affine transformation such as scaling, translating and rotating.

Let $z = f(x, y)$ represents the grayscale value of a pixel in a domain block D_k , where (x, y) is the coordinate in x-y plan. There are two spatial dimensions and the gray level adds a third dimension. Likewise, $f'(x, y)$ denotes the pixel value in a range block R_k . The affine transformation in such 3-dimension (f): $f \rightarrow f'$ can be summarized by the following form

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & i \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} u_i \\ v_i \\ \Delta_i \end{bmatrix} \quad (1)$$

where the coefficients $a_i, b_i, c_i, d_i, u_i, v_i, i$ and Δ_i are all transformation parameters as called fractal codes. The most commonly used distortion measure in fractal image compression is root mean square error (**rms**). The **rms** between two $n \times n$ blocks R and \hat{R} is given by

$$d(R, \hat{R}) = \sqrt{\frac{1}{n^2} \sum_{(x,y) \in R, \hat{R}} (f(x,y) - \hat{f}(x,y))^2} \quad (2)$$

Conventionally, a domain block is double size of a range block and need to be scaled down by a factor two through subsampling with averaging of every four neighboring pixels. The fractal monochrome image coding is to find the minimum approximation error for a domain block through affine transformation to match a range block by minimizing the **rms** between them.

3. FRACTAL-BASED COLOR IMAGE CODING

The fractal coding algorithm for monochrome image can be easily employed in color image compression. A true color image is usually represented by 24 bits per pixel in RGB color space with each component R, G and B assigned 8 bits for the convenience to display colors on the monitor. The most straightforward method to encode a color image by gray-level fractal image coding is to split the RGB color image into 3 channels, red, green and blue, and compress the 3 channels separately by treating every color component's image as a grayscale image. The results of such separate fractal color image coding are shown in **Table 1**. This method, however, does not exploit the correlation between different color components (spectral redundancy) resulting in relatively low compression ratio.

In fact, the pixel in RGB color space can be considered as a 3-dimensional vector with elements of RGB components. Let $\mathbf{f}(x,y) = [r, g, b]^T$ denote a color pixel (3-dimensional vector) in a range block R where (x,y) is the coordinate in x-y plane and (r,g,b) are the three color components' values. In this case, there are two spatial dimensions with RGB color components add 3 dimensions. The square of Euclidean distance of two color vectors $\mathbf{f}(x,y), \hat{\mathbf{f}}(x,y)$ in RGB color space can be used as distortion measure and it is given by

$$d(\mathbf{f}, \hat{\mathbf{f}}) = \|\mathbf{f} - \hat{\mathbf{f}}\|^2 = (\mathbf{f} - \hat{\mathbf{f}})^T (\mathbf{f} - \hat{\mathbf{f}}) \quad (3)$$

Now, suppose R and \hat{R} are two $n \times n$ square blocks, the root mean square error (**rms**) in RGB color for similarity measure of two blocks R and \hat{R} need to be defined as

$$d(R, \hat{R}) = \sqrt{\frac{1}{3n^2} \sum_{(x,y) \in R, \hat{R}} [\mathbf{f}(x,y) - \hat{\mathbf{f}}(x,y)]^T [\mathbf{f}(x,y) - \hat{\mathbf{f}}(x,y)]} \quad (4)$$

Naturally, the affine transformation realized in RGB color space is required to change in the following form

$$\begin{bmatrix} x \\ y \\ r \\ g \\ b \end{bmatrix} = \begin{bmatrix} a_i & b_i & 0 & 0 & 0 \\ c_i & d_i & 0 & 0 & 0 \\ 0 & 0 & r_i & 0 & 0 \\ 0 & 0 & 0 & g_i & 0 \\ 0 & 0 & 0 & 0 & b_i \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ r \\ g \\ b \end{bmatrix} + \begin{bmatrix} u_i \\ v_i \\ \Delta_i^r \\ \Delta_i^g \\ \Delta_i^b \end{bmatrix} \quad (5)$$

where $a_i, b_i, c_i, d_i, u_i, v_i, r_i, g_i, b_i, \Delta_i^r, \Delta_i^g$ and Δ_i^b are all transformation parameters. With the above 5-dimensional affine transformation and vector distortion measure, it is easy to apply fractal image coding method to compress RGB color images simultaneously to exploit the spectral redundancy for increasing compression ratio. Simulation results for coding 512×512 test image *Lenna* using the proposed fractal color image coding technique with fixed partition are listed in **Table 1**. The results show that there is about 1.5 compression ratio improvement of the proposed technique when compared with the separate fractal coding.

4. ADAPTIVE QUADTREE-BASED FRACTAL COLOR IMAGE CODING

In **Table 1**, only fixed partition is used in the fractal color image coding. It is obvious that high compression ratio can be achieved if the image is divided into large range block such as 8×8 , but more block artifacts exist in the reconstructed image (see **Fig.2c**). In the contrary, if the image is divided into small range block such as 4×4 , the reconstructed image quality is very high as shown in **Fig.2b**, but the compression ratio is only 10.97. In order to achieve a good trade-off between compression ratio and image fidelity, an adaptive quadtree partition scheme with use of vector distortion can be adopted in fractal image coding.

Quadtree algorithm is often used in vector quantization (VQ) and then introduced into fractal image coding by Fisher [2]. The image is firstly segmented into non-overlapping maximum range block of size $B \times B$. For every range block, if all domain blocks of size $2B \times 2B$ in the search pool cannot find a good match

based on a predetermined error threshold, it will be divided into four quadrant sub-blocks of size $B/2 \times B/2$. To each of these four sub-blocks, the same searching process is done in the corresponding search pool with domain blocks of size $B \times B$. Once a sub-block can not find a good match, it will be further subdivided into four smaller quadrants of size $B/4 \times B/4$ until the minimum range block has been reached. When a range block cannot be partitioned again, the searching process will find the domain block which has the smallest approximation error, although this error maybe exceeds the preset error threshold. **Fig. 1** gives an example of 3 level quadtree partition of one range block.

For example, **Fig. 2d** gives a reconstructed image with use of quadtree partition starting from biggest range block 16×16 and ending at block size 4×4 . Such adaptive quadtree-based fractal coding using the vector distortion measure in RGB color space can obtain compression ratio as high as 27.79 with acceptable image quality.

5. DISCUSSIONS AND CONCLUSIONS

Computer simulation is done on standard 512×512 RGB color image *Lenna* with 24 bits per pixel and each color component R, G and B assigned 8 bits. The test image is first divided into sixteen 128×128 sub-images and then processed independently. The search pool is constructed by sliding a $D \times D$ window along horizontal and vertical direction at step size Δh and Δv , respectively. Obviously, the elements of the search pool is in inverse proportion to $\Delta h \times \Delta v$. Therefore, if the search step size is small, the search pool contains more potential domain blocks and a range block has a higher chance to find a better match domain block. Conversely, if the search step size is large, the search pool is small and the probability of a range block to match a domain block is low. Thus, the reconstructed image quality is relatively lower. To restrict the possible affine transformation to a reasonable level, only 8 canonical isometries of a square block are used in fractal coding. (see[1,5] for detail). The scale parameters $\begin{matrix} r \\ i \end{matrix}, \begin{matrix} g \\ i \end{matrix}, \begin{matrix} b \\ i \end{matrix}$, and shift parameters $\Delta_i^r, \Delta_i^g, \Delta_i^b$ can be calculated using least-square optimization [2] and must be quantized for efficient coding. The experimental results of fractal color image coding using the vector distortion measure are shown in **Table 1** as compared to separate fractal coding of RGB images. It is obvious that using vector distortion measure in fractal color image coding can achieve about 1.5 compression ratio improvement.

	range block	domain block	$\Delta h, \Delta v$	compression ratio	PSNR (dB)
Separate Fractal Coding in RGB Images	8×8	16×16	16	30.12	27.33
			8	26.95	28.01
			4	24.38	28.27
	4×4	8×8	16	7.53	30.85
			8	6.74	31.86
			4	6.10	32.44
Vector Distortion Fractal Coding	8×8	16×16	16	46.54	26.90
			8	43.88	27.46
			4	41.51	27.80
	4×4	8×8	16	11.64	30.17
			8	10.97	31.11
			4	10.38	31.64

Table 1. Results for coding 512×512 test image *Lenna* using two different fractal color image compression technique with fixed partition.

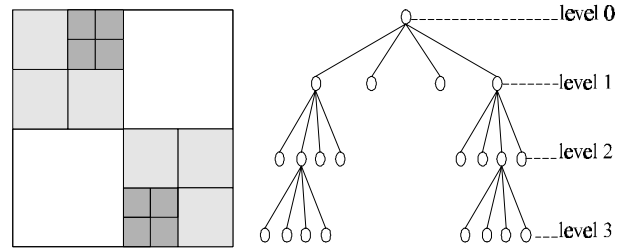


Fig.1. An example of 3 level quadtree structure partition of an image block

In order to evaluate the reconstructed image quality, the averaged *mse* (mean square error) of three RGB color components is used in the following form to compute the peak-signal-to-noise-ratio (PSNR) which is given by

$$\text{PSNR} = 10 \log_{10} \left(\frac{255 \times 255}{\text{mse}} \right) \quad (6)$$

Some reconstructed images using vector distortion measure in fractal coding with quadtree or fixed partition are illustrated in **Fig. 2b** to **Fig. 2d**. (Note : only the luminance component of the RGB color images are shown in **Fig.2**). Obviously, using quadtree partition in vector distortion fractal color image coding can obtain a good trade-off between compression ratio and image quality. (See **Fig.2d**).

As each color component in RGB has different contribution to color distortion perceived by human vision, it is more suitable using weighted Euclidean distance in vector distortion measure to obtain a better image quality or higher compression ratio. Further consideration includes compressing the color images in different color space, for example, YIQ, YUV, and $L^*a^*b^*$ etc., to find the best space which can achieve high compression ratio as well as good image fidelity.

References

- [1] Arnaud Jacquin, " Fractal image coding based on a theory of iterated contractive image transformations ", SPIE: Visual Communications and Image Processing, 1990, vol. 1360, pp.227-239.
- [2] E. W. Jacobs, Y. Fisher and R. D. Boss, " Fractal image compression using iterated transforms", in: *Image and Text Compression*, Kluwer Academic Publishers, Dordrecht, 1992, pp.35-61.
- [3] T. Bedford, F. M. Dekking, M. Breeuwer, M. S. Keane and D. Van Schooneveld , " Fractal coding of monochrome images ", Signal Processing: Image Communications, June, 1994, pp.405-419.
- [4] M. F. Barnsley and L. P. Hurd, *Fractal Image Compression*, AK Peter, Wellesley, 1992.
- [5] Arnaud E. Jacquin, " Fractal Image Coding : A Review ", Proceedings of The IEEE, Oct. 1993, pp.1451-1465.



Fig.2a. Original 512×512 *Lenna*



Fig.2b. Vector distortion fractal coding using fixed partition : range block 4×4, domain block 8×8, $\Delta h, \Delta v = 8$, compression ratio = 10.97, PSNR = 31.11 dB.



Fig.2c. Vector distortion fractal coding using fixed partition : range block 8×8, domain block 16×16, $\Delta h, \Delta v = 16$, compression ratio = 46.54, PSNR = 26.90 dB.



Fig.2d. Vector distortion fractal coding using quadtree partition : compression ratio = 27.79, PSNR = 29.69 dB.