2012 International Workshop on Nonlinear Circuits, Communications and Signal Processing
NCSP'12, Honolulu, Hawaii, USA. March 4-6, 2012

On Investigation of the Intensity Scaling Factor in Fractal Image Coding

Yoshinori Kitano[1], Tetsushi Ueta[1], Yoshifumi Nishio[1], Guoan Yang[2], and Yang Yang[2]

[1]Tokushima University [2]Xi'an Jiaotong University
Email: kitano-yoshinori@is.tokushima-u.ac.jp

Abstract

Fractal image coding uses geometrical self-similarity properties in the given image and it shows a high compression ratio. There are many research results on gray-scale or color natural images with this coding system, in this paper we specify that there is a trouble with black/white images because of the scaling factor value of the affine mapping for pixel values. To overcome this problem, we examine the scaling factors versus S/N ratio for various black and white images, and investigate the scaling factor values distribution resulting the maximum S/N ratio. With these values, the decoded image looks very natural.

1. Introduction

According to diffusion of Internet and multimedia, image compressing techniques become very important for improve speed and saving storage media. Today, some techniques of image compression such as JPEG, GIF and PNG method that are using techniques such as statistical redundancy of the image frequency are researched and they are the mainstream of image compression method. In contrast, there is a fractal compression that has a compression technique using geometric redundancy in the image. It has features such as high compression and high-speed decoding of compressed pixel data from the current mainstream method. Therefore, this one has an attracted attention as a next generation image compression method.

Fractal coding that using fractal compression has been proposed by Barnsley et in 1988[2]. This coding is using the region self similarity that the global structure used the similarity in local structure. Then, Jacquin et proposed partial self similarity of fractal coding that search in blocks[2] and the latter approach has become mainstream now. This approach is divided into blocks without overlapping images and is that encoding using similarity blocks of different sizes. When we decode the image, decoded image is generated by repeating any blocks for affine mapping between the blocks stored in compressed data. while fractal coding takes much time for coding computations, there are features of fractal coding that is better than compression ratio of the other methods and allows fast decoding. Today, research of the image processing has been studied using geometric information of the compressed data.

Pen drawings, cartoons, sketches are often composed by a set of sharp black lines, dots, screen tones. When the original fractal coding scheme is applied to a binary valued image (only black and white pixels are stored), we noticed the decompressed (decoded) images tend to be remarkably degraded since the scaling parameter of the intensity direction in the affine mapping is fixed. In fact, this parameter was appeared in the Barnsley's original algorithm[1], and it has been fixed for a long time. In this paper, we explain this phenomenon, and propose its work around.

2. Fractal Coding

A practical scheme of the image compression is described below. At first, we divide an original image into some small blocks not to be overlapped one another. Here we call them domain blocks. Next, we make a $2N \times 2N$ size of domain block from the original image. Here we call this a range block. When we fix a domain block, a fractal code is a set of parameters for affine transformation with the best match range block in the image:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} \mapsto \begin{pmatrix} a_i & b_i & 0 \\ c_i & d_i & 0 \\ 0 & 0 & p_i \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} + \begin{pmatrix} s_i \\ t_i \\ q_i \end{pmatrix} \quad (1)$$

where, x, y are the positions of the domain block. z is the pixel value of (x, y). a_i, b_i, c_i, d_i, p_i, q_i, s_i, t_i are parameters. Fractal image codes are these sets of position and parameters. Note that these codes are usually encoded again by an entropy encoding algorithm to remove the redundancy of the parameters.

The decompressing scheme is described as follows. At first, we create an image of the same size as the original image and divide it by the domain block. we obtain the original image by replacing all domain blocks with corresponding range blocks with iterating the affine mapping, see Fig.1. Parameters of the affine mapping are restored from the compressed data.

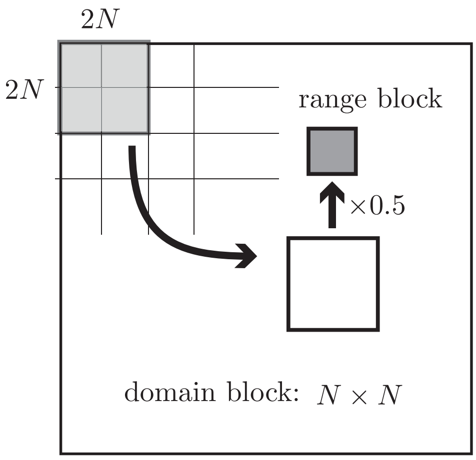
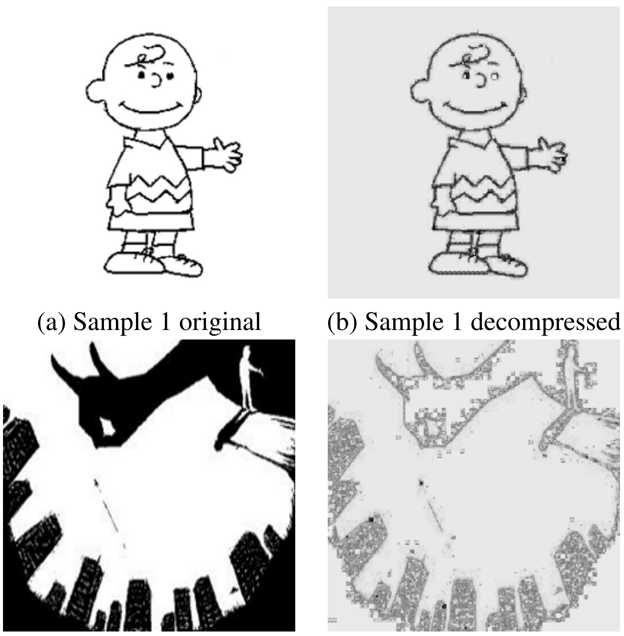


Figure 1: Definition of the domain block and the range block.

3. Conventional Method

When one tries the fractal coding for a binary valued image with the original Barnsley's code, an undesired results are happened, see Fig. 2.



(a) Sample 1 original (b) Sample 1 decompressed

(a) Sample 2 original (b) Sample 2 decompressed

Figure 2: Examples: undesired degraded decoding. (a) and (c): the original images. (b) and (d): decoded images with Barnsley's original algorithm.

For natural gray-scale images, the fractal coding works well since distribution of pixel values are not appropriately spread. This means that the decoding error does not specifically change impression of the whole image. However, for binary valued images, drastic degrading results are obtained. In Fig. 2(b), a white background is filled by a gray pixel. In Fig 2(d), it seems to be a completely false image. The reason of these defeats is the fixed value of p_i in Eq. (1). Historically, 0.9 is chosen as p_i, because the first version of Barnsley's algorithm, this value is specified without any explanation. For gray scaled images, a certain range of p_i, say, $0.5 < p_i < 0.9$ is allowed to retain enough quality of the target image. For the binary valued images, however, this 0.9 is not a suitable number for coding. In the pixel value direction, a contract mapping $p_i z + q_i$ (gray scale reduction) does not prevent undesirable mixture of gray pixels enough.

4. Proposed Method

If we take the other value for p_i, what is happened ? Figure **??** shows decompressed images for Fig. 2. For Charley Brown, $p_i = 0.9$ leaves the best quality. Now we propose a modified fractal compression for binary valued images, i. e., we adaptively choose p_i for the given image.

To evaluate the quality of decompressed image, we define the peak signal-to-noise ratio $PSNR$ as follows:

$$PSNR = 10 \log \left\{ \frac{255 \times 255}{MSE} \right\} \quad (2)$$

$$MSE = \frac{\sum_{i=0}^{m} \sum_{j=0}^{n} f(i,j) - f'(i,j)}{m \times n} \quad (3)$$

where, we assume the image is $m \times n$ size. f and f' are the original and the decompressed image pixel value, respectively and they range 0 to 255. If the decompressed image is loss-less, the PSNR becomes infinity. Generally, if PSNR is around 40 dB, people could not recognize the difference between original and decompressed image. We use $m = n = 256$, and 4 for the domain block.

The strategy to obtain p_i generating the best quality decoded image, we simply check brute-force computation, i.e., simply scan all values of $p_i < 1$. A p_i marking the largest value of PSNR in this scan is the best value.

5. Experimental Result

We use Apple Mac Mini, 2.4 GHz Intel Core 2 Duo, 2MB memory for computer simulations. The average coding time for all figures is 200 seconds. Tables **??**–**??** show computation comparison with the conventional method (Barnsley's fixed parameter) and our best quality p_i. We do not change any algorithm, the computation time and compression ratio are not changed.

We summarize the features of these results below:

- It seems that there is an extreme value for Sample 1 and 4. Beyond the extreme value, the PSNR suddenly declines like a threshold effect. Anyway in these samples, Barnsley's p_i can only leave poor decoded images. We should choose smaller value for p_i, e.g., $0.65 < p_i < 0.7$.

- The fixed value of p_i used in Barnsley's algorithm does not gain the best PSNR for any binary valued images.

- Oscillatory PSNRs are found in Sample 2 and 3. These images contain thin and complex structures inside. Probably in the domain and range block matching stage, the size of the domain block affects PSNR too much since a slight difference of shapes between domain and range block makes completely different referenced locations.

- It is difficult to find out generic tendency of these examples. Some normalized images (e.g., parameterized circles, squares, rectangles and other objects) should be examined.

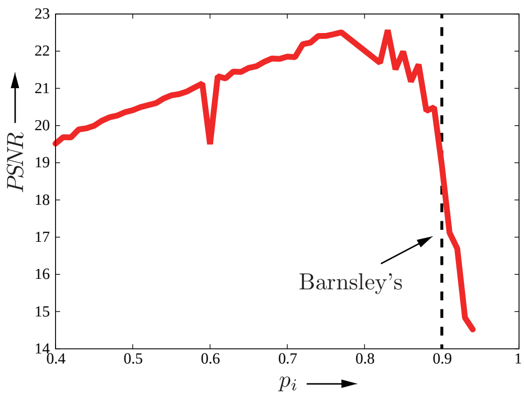


Figure 3: p_i-PSNR graph for Sample 1.

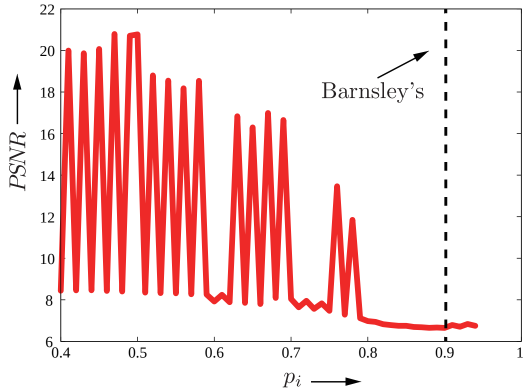


Figure 4: p_i-PSNR graph for Sample 2.

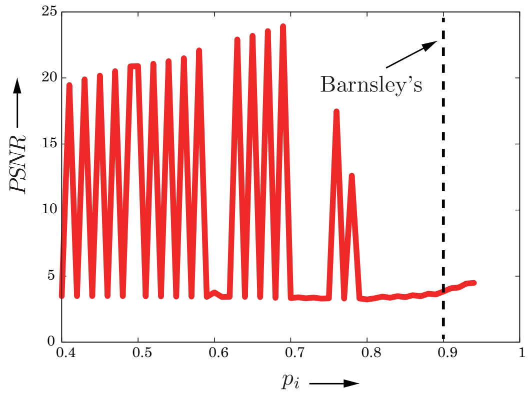


Figure 5: p_i-PSNR graph for Sample 3.

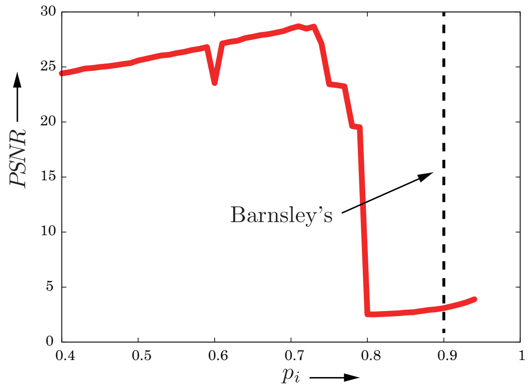
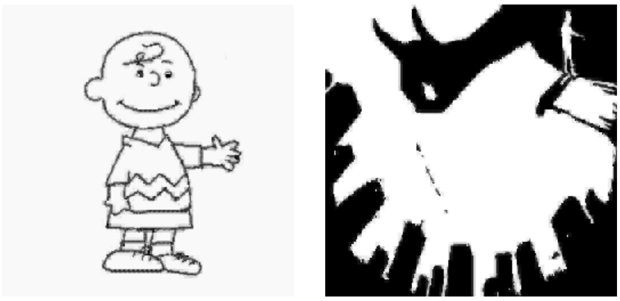
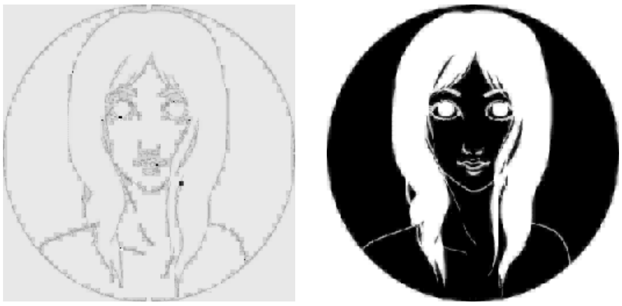


Figure 6: p_i-PSNR graph for Sample 4.



(a) (b)

Figure 7: The best images for Sample 1 and 2 with modified value of p_i. (a): $p_i = 0.82$, (b): $p_i = 0.49$.



(a) Barnsley's (b) $p_i = 0.69$.

Figure 8: Sample 3

(a) Barnsley's (b) $p_i = 0.71$.

Figure 9: Sample 4

6. Conclusion

We investigated the role of p_i used in the fractal coding. We clarified that the fixed value of p_i introduced in Barnsley's first issue do not have a reasonable evidence for its determination. Since the brute force method consume much computational costs and time to obtain the best value of p_i, we would like to derive an adaptive design method, i.e., an evaluation function and its optimizing method should be developed.

References

[1] M. Barnsley and L. Hurd, Fractal Image Compression, AK Peters Ltd., 1993.

[2] A. Jacquin, "A Novel Fractal Block-Coding Technique For Digital Images," IEEE ICASSP, pp.22–25, 1990.

[3] A. Takeuchi, T. Ueta, and K. Terada, "Improvements of Image Segmentation by using Fractal Coding," NOLTA'99, pp.347–350, Hawaii, 1999.

[4] T. Masaki, T. Ueta, and K. Terada, "Fingerprint Identification using Fractal Coding," Proc. NOLTA'99, Vol. 1, pp. 143–146, Hawaii, Dec. 1999.

[5] T. Nakanishi, T. Ueta, "Loss-less Fractal Compression Method for Line Drawings and Illustrations," Proc. NCSP'10, pp. 309–312, Hawaii, Mar. 2010.