2004 International Symposium on Nonlinear
Theory and its Applications (NOLTA2004)
Fukuoka, Japan, Nov. 29 - Dec. 3, 2004

The Effectiveness of Discrete Voronoi Division in Fractal Image Compression

Takayasu FUCHIDA and Sadayuki MURASHIMA

Department of Information and Computer Science, Kagoshima University
1-21-40 Kohrimoto, Kagoshima-shi, Kagoshima 890-0065 Japan
Email: {fuchida,mura}@ics.kagoshima-u.ac.jp

Abstract—In this paper, we show the effectiveness of discrete Voronoi division as the shape of ranges in fractal image compression. Normally, a square shape is used for a range of fractal image compression. On the other hand, we've proposed the way to use a discrete Voronoi region as a range. It is thought that a Voronoi range has more flexible shape than a square range, but the effectiveness of discrete Voronoi division for fractal image compression has not been investigated yet. We compress some images using these two methods and compare the values of error between range and domain by using computer simulations.

1. Introduction

Fractal image compression is one of the way of image compression using self-similarity contained in the original image [1, 2]. It has attracted attentions as a method of high compression ratio as much as JPEG method or more. In the normal fractal image compression method, a square shape is used as a range. But it is hardly thought that the square range would become the best fitting shape to the natural image. A flexible shape — like a polygonal shape — has complex features and it was thought that is could not be used for the shape of range in fractal image compression.

Recently we have proposed a way to compress an image by using discrete Voronoi division as range division in fractal image compression [5]. Discrete Voronoi diagram can be constructed from only the information of kernel points. So we can realize flexible ranges by using discrete Voronoi diagram.

The Voronoi range may be more flexible than square range but it is not clear how the Voronoi range effective.

So we compare the two methods — square range and Voronoi range — and calculate the value of errors between range and domain. By comparing these values, we can know how the obtained range is worthy to extract the self-similarity from the given image.

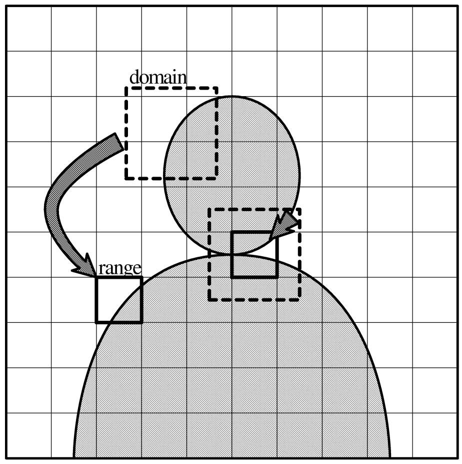


Figure 1: Relationship between domain and range

2. Fractal image compression using discrete Voronoi division

2.1. Fractal image compression

"Fractal" is a coined word by Mandelbrot and it means self-similarity [3]. The fractal image compression method compresses an image using the self-similarity contained in the original image. Concretely, a given image is first divided into the many areas called *range* which does not overlap with each other. Next, for each range, we search the *domain* which has (normally) twice the size of the range. At this time, the domain whose similarity to the range is the highest is searched. Here, the similarity is calculated by using RMS metric as shown later. Fig.1 shows the relationship between a range and a domain. The domain may overlap with each other.

Such a system like this is called *partitioned iterated function system* (PIFS). Former researches of fractal image compression, almost all cases were using a square range. This method is called *quad tree method.* In this method, if the value of RMS error of a range was not enough small, the range was divided into 4 square ranges. By repeating this process, a range is divided small gradually.

The contractive map from domain to range can be represented by using affine transformation. General affine transformation of ith range w_i is shown in formula (1).

$$w_i \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1/2 & 0 & 0 \\ 0 & 1/2 & 0 \\ 0 & 0 & s_i \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} a_i \\ b_i \\ t_i \end{bmatrix} \quad (1)$$

Where x, y are the coordinates of domain pixel and z is the brightness of this pixel. Normally, because the size of domain is twice the size of range, the affine parameters of resize, rotation and shear are fixed to $\begin{pmatrix} 1/2 & 0 \\ 0 & 1/2 \end{pmatrix}$. So a domain pixel means the block of 2×2 pixels and the position of this pixel is upper-left coordinates of this block, and the brightness means the sum of brightnesses of these 4 pixels. The parameters a_i and b_i represent parallel translation, and s_i and t_i decide the conversion of brightness value.

When the intensities of pixels in the range are $r_1, r_2, \cdots, r_n$, and the intensities of pixels in the domain are $d_i, d_2, \cdots, d_n$, the RMS error E_i can be calculated as follows.

$$E_i = \sum_{j=1}^{n} (s_i d_j + t_i - r_j)^2 \quad (2)$$

where n is the number of pixel contained in the range.

We can seek s_i and t_i to minimize this E_i. The minimum of E_i occurs when the partial derivatives with respect to s_i and t_i become zero as follows.

$$\frac{\partial E_i}{\partial s_i} = 0, \qquad \frac{\partial E_i}{\partial t_i} = 0 \quad (3)$$

This value E_i will become settled if any domain is specified. The domain where the value of E_i is possible the smallest is an optimum domain.

2.2. Fractal image compression using discrete Voronoi division

Voronoi diagram is a diagram which divide space into the influence areas of kernel points. Kernel point is a generator of Voronoi diagram. A point which exists in one Voronoi region is a point whose nearest kernel point is that region's kernel point.

A discrete Voronoi diagram is a Voronoi diagram which is made on the digitized space. The discrete Voronoi region is a set of pixels whose nearest kernel point is that region's kernel point. If we got a discrete Voronoi diagram, we could know which kernel point is the nearest one from any point of that image in an instant.

When an original image was given, we can construct a discrete Voronoi diagram on that image. And also we can use these discrete Voronoi regions as the ranges of fractal image compression.

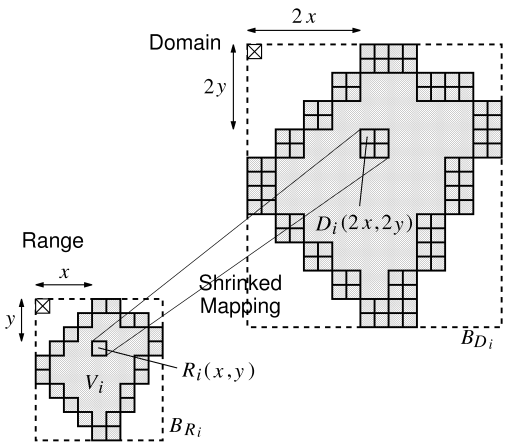


Figure 2: Contractive mapping from domain to range

In this case, we can use the affine transformation from a domain to a range like the case of the square range. Fig.2 shows the affine transformation in the case of using discrete Voronoi range. In this figure, $R_i(x, y)$ is a pixel of ith range, and $D_i(2x, 2y)$ is the corresponding domain to that range. B_{R_i} and B_{D_i} are the boundary rectangle of the range and domain respectively, and x, y are the coordinates in the boundary rectangle. We denote the sum of 4 pixels $D_i(2x, 2y), D_i(2x+1, 2y), D_i(2x, 2y+1)$ and $D_i(2x+1, 2y+1)$ as $D_i^*(x, y)$.

Then the RMS error E_i can be calculated as follows.

$$E_i = \sum_{x,y \in V_i} (s_i D_i^*(x, y) + t_i - R_i(x, y))^2 \quad (4)$$

In this formula, V_i is the discrete Voronoi region of ith range.

2.3. The way of range division

Generally, as a range becomes smaller, it becomes easier to find a good domain. So we will want to divide the image into small ranges but a lots of ranges causes bad compression ratio. Because the quantities of information which should be stored into the compressed file will also increase as the number of ranges increase.

So when we would divide one range, we should divide a range whose RMS error is larger than any other ranges.

The strategy of dividing range is as follows.

When the square range is used, we divide the original image into square ranges roughly, then we calculate the RMS error for all ranges. After this, we select n_w ranges and divide them into 4 ranges. We repeat this process until the number of ranges reached over n_m.

When the Voronoi range is used, range division is decided by the position of kernel points. And if we added a kernel point to arbitrary point on the image, the shape of several ranges which exist around this point would be changed by this addition. So after adding a kernel point, we will have to reconstruct the discrete Voronoi diagram to obtain new Voronoi ranges. Therefore we take the following strategy to decide the range division.

1. First, we distribute n_i kernel points on the original image randomly and divide the whole image into n_i initial ranges.
2. Optimum domains are searched for every ranges. At this time, the value of RMS error for each range will be calculated.
3. if the number of range reached to n_m then end.
4. We select n_w ranges whose RMS errors are larger than any other ranges and remove the kernel points of these ranges and add two kernel points in each removed range randomly.
5. Reconstruct the discrete Voronoi diagram by using new kernel points.
6. go to 2.

By taking this strategy, we can decide the number of range freely. But since this algorithm constructs discrete Voronoi diagram many times, it is required a effective construction method of discrete Voronoi diagram. In this research, we used the incremental method on digitized space to construct a discrete Voronoi diagram quickly [4].

3. Computer simulations

In order to compare the difference of two form of range, we performed several computer simulations. We used 3 images whose size is 512×512, — Lenna, Peppers and Baboon. These are the standard images for data compression.

Simulation environments are as follows. CPU is Pentium4 3G Hz and Memory is 1.5G Byte, Windows XP Professional. Programming language is Gnu C++.

Fig.3(a) and (b) show examples of range division by using square ranges and Voronoi ranges respectively. In both divisions, they have almost 10000 ranges. It turns out that the Voronoi ranges more concentrate on the detailed parts of Lenna than the square ranges. (For example, on the part of hair.)

Fig.4 shows the result of comparing the quality of images to the compression ratio in 3 cases. The horizontal axis shows the compression [bpp], and the vertical axis shows the PSNR [dB]. The original image is 8 [bpp]. PSNR is calculated by following formula,

(a) Range division using square range.

(b) Range division using Voronoi range.

Figure 3: Range division of both way

$$\mathrm{PSNR} = 20 \log_{10} \frac{256}{\mathrm{RMSE}}. \qquad (5)$$

From these figures, it is found that when the compression ratio is greater than 0.4 [bpp], the method using Voronoi range is better than conventional method. But as the compression ratio becomes smaller, the quality of restored image of conventional method becomes slightly better than the proposed method.

Since the proposed method is more complicated than the conventional method, it takes much compression time than the conventional one. The conventional method takes about 50 seconds when the compression ratio is 0.4 [bpp], but the proposed method takes about 500 seconds in same ratio. Voronoi method is slower than conventional method about 10 times in this case. Some reasons can be considered about this. One reason is the complexity of the form of Voronoi range. The flexibility also means the complexity. Therefore we have to more times of calculation to obtain the RMS error of Voronoi range than square range. In the case of square range, we can use pre-calculated cumulations to calculate the RMS error but it is impossible for Voronoi ranges. For these reasons, the proposed method takes much time.

4. Conclusions

In this paper, we compared two way of range division in fractal image compression. When the compression ratio is over 0.4 [bpp], it was found that the Voronoi range could realize higher quality than square range. But when the compression ratio becomes smaller than it, the quality of restored image becomes slightly worse. One of the reason is that since the Voronoi range is more flexible than the square range then it can be arranged at the complicated parts of original image.

On the other hand, the proposed method requires more calculation time to compress image. Since the Voronoi range has a complex form then we can not use the pre-calculated cumulations to obtain the RMS errors.

Acknowledgements

The portion of this study has been funded by the Grants-in-Aid for Scientific Research No.14780233 of the Japanese Ministry of Education, Culture, Sports, Science and Technology.

References

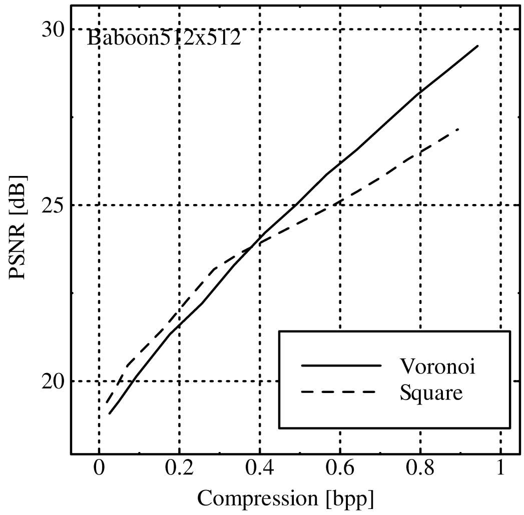
[1] Yuval Fisher Editor, *Fractal Image Compression — Theory and Application —*, Springer, 1994.

[2] M.Nelson, J.L.Gailly, *The Data Compression Book Second Edition*, M&T Books, 1995.

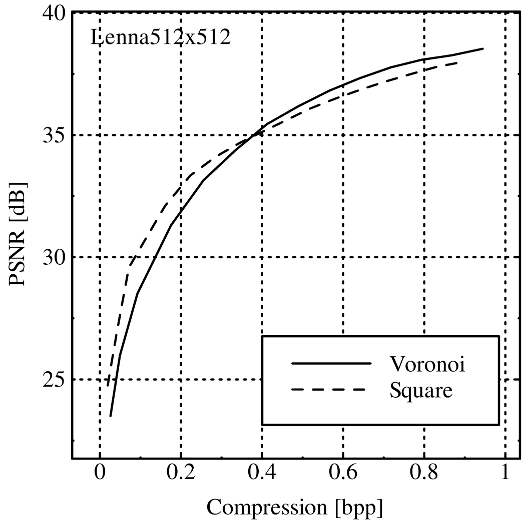
[3] B.B. Mandelbrot, *The Fractal Geometry of Nature, 2nd edition*, W. H. Freeman and Co., San Francisco, California, 1982.

[4] T.Fuchida, H.Nakamura, K.Mori and S.Murashima, *An Efficient Method to Construct a 2-Dimensional Discrete Voronoi Diagram by Adding Kernel Points One by One*, IEICE Trans A, Vol.J85-A No.5 pp.571–583, May 2002. (in Japanese)

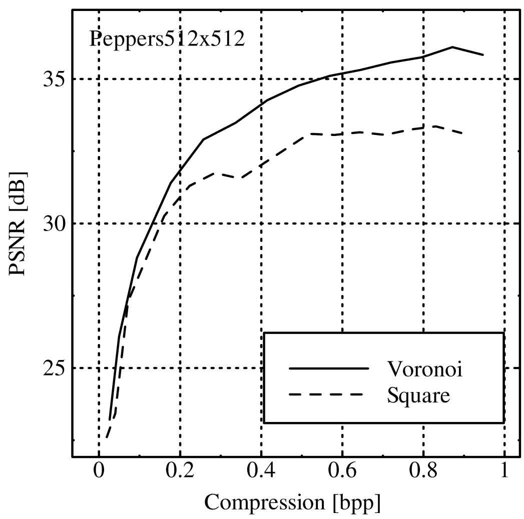
[5] T.Fuchida, H.Nakamura, K.Mori and S.Murashima, *Fractal Image Compression using Discrete Voronoi Tessellation*, Proceedings of International Symposium on NOLTA 2002, Vol.2 pp.711–714, Dec. 2002.



(a) result of 'Baboon'



(b) result of 'Lenna'



(c) result of 'Peppers'

Figure 4: Results of computer simulations