

A NOVEL FAST FRACTAL IMAGE COMPRESSION METHOD BASED ON DISTANCE CLUSTERING IN HIGH DIMENSIONAL SPHERE SURFACE

SHUAI LIU,^{*,†} ZHENG PAN^{*,†} and XIAOCHUN CHENG^{‡,§}

**College of Computer Science*

†Key Laboratory of Data Mining and Knowledge Engineering

Inner Mongolia University

Inner Mongolia, Hohhot 010021, P. R. China

‡Department of Computer Science

Middlesex University

London NW4 4BT, UK

§X.Cheng@mdx.ac.uk

Received January 5, 2017

Revised March 1, 2017

Accepted March 4, 2017

Published June 2, 2017

Abstract

Fractal encoding method becomes an effective image compression method because of its high compression ratio and short decompressing time. But one problem of known fractal compression method is its high computational complexity and consequent long compressing time. To address this issue, in this paper, distance clustering in high dimensional sphere surface is applied to speed up the fractal compression method. Firstly, as a preprocessing strategy, an image is divided into blocks, which are mapped on high dimensional sphere surface. Secondly, a novel image matching method is presented based on distance clustering on high dimensional sphere surface. Then,

[§]Corresponding author.

This is an Open Access article published by World Scientific Publishing Company. It is distributed under the terms of the Creative Commons Attribution 4.0 (CC-BY) License. Further distribution of this work is permitted, provided the original work is properly cited.

the correctness and effectiveness properties of the mentioned method are analyzed. Finally, experimental results validate the positive performance gain of the method.

Keywords: Fractal Image Compression; Sphere Surface; Distance Clustering.

1. INTRODUCTION

Image compression is important for storing and communicating image data, which is the basis of video and other multimedia services. With the increment of image quality and resolution ratio, image size increases quickly and becomes the bottleneck in real-world applications. The increment of image size needs larger space to store and larger bandwidth to transfer. How to decrease image size becomes an important research subject in image processing.

Two fundamental ways of compression are redundancy and irrelevancy reduction.¹ Redundancy reduction aims at removing duplication from the signal source. Irrelevancy reduction omits parts of the signal that will not be noted by the signal receiver, such as the Human Visual System (HVS).

There are many classic methods and international standards of image compression and encoding. As we know, discrete cosine transformation (DCT),² Huffman coding,³ wavelet image coding,⁴ etc., are all classical image compression and encoding methods. BIG, JPEG, H.263, MPEG, etc., are all classical international standards.

The above-mentioned classical methods analyze the correlation between pixels to remove spatial redundancy or spectral redundancy in order to compress images. However, the compression ratio reaches the bottleneck now. Thus, in order to reach higher compression ratio, it needs new information redundancy reduction method of image.

Fractal is researched for the non-classical geometry since 1980s.^{5,6} As we know, artificial object usually has a regular and smooth form that can be processed by traditional geometry and geometrical form of natural object is usually rough and anomalous. However, the self-similarity may be found in natural object when research is conducted on its rough and anomalous form.

For years, self-similarity is used to study the properties of natural objects like galaxies, lasers and waves.⁷⁻⁹ Then, fractal thinking applies to image compression by contractive affine transformation

(CAT), which constrains iteration uncompressing convergent by Banach fixed point theorem.^{10,11} The fractal compression applies self-similarity of blocks with different sizes in compressing image and stores the parameters of transformation instead of classical compression method, which stores color, position and other information of pixels. In fractal compression, only quantization parameters of CAT needs to be stored. This is because Banach theorem proves the convergence of iteration of CAT. In other words, all blocks of original image can be reconstructed with some black blocks that need not be stored.

This novel image compression method is named as fractal image compression and it is also an image encoding method. In fractal compression, it first extracts the self-similarity between large and small parts of images. Then it stores the quantization parameters of CAT as a compressing file. So the compressing file is much smaller than the original image. In this case, it can reach higher compressing ratio.

Fractal method has many benefits in the domain of image compression. For example, fractal compression method with same parameters reaches fixed compression ratio for all images with same size. Also, because affine iteration is existent and unique, iteration will reach the original image better with more iteration times. The original image is the ultimate limit of the iteration.

But fractal compression method is restricted by its high computational time. The high computational time mainly appears in the iterating comparison between range and domain blocks. Thus, if comparison results can be recycled, computational time will be decreased.

This paper presents a novel fractal compression method. Firstly, we pre-process using the classical fractal compression method. In this case, a distributed fractal compression method is proposed. Then, we proved its correctness and analyze its effectiveness in theory. Finally, experiments using several classical images are performed to validate our conclusion.

The remainder of the paper is organized as follows. Related works are presented in Sec. 2. Then, we present pre-processing method and a novel fractal compression method in Sec. 3. Further, strict proof of correctness and effectiveness are presented in Sec. 4. Furthermore, experiment and analysis are presented in Sec. 5. Finally, Sec. 6 summarizes the whole paper.

2. RELATED WORKS

Thinking of fractal compression was first presented by Barnsley and Jacquin.¹¹ They provided iterated function systems (IFS). Then, Jacquin fused CAT and IFS, the first presented fractal compression method.¹² Meantime, Bedford *et al.* applied fractal compression method into monochrome images.¹³ Further, Kim *et al.* applied vector quantization into fractal compression for image and Kim *et al.* applied fractal compression into video sequence processing.^{14,15} These were all the basic researches and applications for fractal compression.

Afterwards, fractal compression research was widely improved for image processing. Chang *et al.* designed domain pool for an iteration-free fractal compression method.¹⁶ Tong *et al.* presented a fractal compression method with adaptive search of domain and range blocks.¹⁷ Lai *et al.* presented a kick-out model for fractal image compression method in zero contrast condition.¹⁸ Distasi *et al.* introduced a fractal image compression method by applied error approach in range and domain blocks.¹⁹ Wang *et al.* provided no-search fractal image compression method for fitting plane and gray-level transform.^{20,21}

Recently, there were more researches in the domain of fractal compression. Respectively, Jeng *et al.* and Lu *et al.* presented fractal image compression method based on Huber loss function.^{22,23} Wang *et al.* provided fractal image compression scheme based on correlation between range and domain blocks.²⁴ Bhavani *et al.* applied fractal compression into medical image compression.²⁵ Wang presented a fractal image compression method by using applied discrete wavelet to classify range blocks.²⁶ Our team also researched on fractal and fractal compression theory and application.²⁷⁻³¹

This paper extends classical fractal image compression method of Refs. 22–24 with novel thinking of distance in high dimension. References 22–24 applies statistical theory into fractal compression method to decrease the comparison between range

and domain blocks. This paper transfers fractal compression method into high dimension space. Then the comparison is researched by analyzing distances between range and domain blocks. This decreases more computational time than the above mentioned references.

3. A NOVEL FRACTAL COMPRESSION METHOD BASED ON DISTANCE CLUSTERING

3.1. Theoretical Basis

First, Banach fixed point theorem is provided in Lemma 1 as a theoretical foundation for fractal compression method. It can be applied to research CAT in Definition 2, which is directly applied into fractal image compression method. In Lemma 1 and Definition 2, we have that R is the set of elements and ρ is a defined distance (computation) in R , T is a transform (mapping from R to R), $r_i^1, r_i^2, d_i^1, d_i^2, s, a_{1i}, b_{1i}, bi \in R$, A^T means transposition of matrix A .

Lemma 1. Assume that (R, ρ) is a complete metric space, there exists only one $r \in R$ that makes $T_r = r$ if in Eq. (1) is tautology.

$$\rho(T_x, T_y) \leq \theta \rho(x, y), \quad s.t. \quad 0 \leq \theta \leq 1, \quad x, y \in R. \quad (1)$$

Definition 2. $W = \left(\begin{bmatrix} a_{11} & a_{12} \\ b_{11} & b_{12} \end{bmatrix}, \begin{bmatrix} b_1 \\ b_2 \end{bmatrix} \right)$ is called a CAT in a Euclid space (R, ρ) , where In Eq. (2) is tautology.

$$\begin{aligned} \rho(R_{n+1}, D_{n+1}) &\leq s \cdot \rho(R_n, D_n), \text{ where} \\ R_i &= (r_{i+1}^1, r_{i+1}^2)^T, \quad D_i = (d_{i+1}^1, d_{i+1}^2)^T, \\ R_{n+1} &= W \cdot (R_n, 1)^T, \quad D_{n+1} = W \cdot (D_n, 1)^T \\ \text{and } 0 &\leq s < 1. \end{aligned} \quad (2)$$

In this way, the compression problem can be described in Eq. (3), where E means a matrix with all elements 1, $\| \cdot \|$ means the vector norm (usually 2-norm in fractal compression), R_i means a determined range block R , D_j means a determined domain block D , $best_i$ means serial number of the best D_j of R_i , s means scaling of R , o means offset of R in an affine mapping and T means transformation of D which contract size of D to size of R . In addition, the compression file consists of s , o and

the corresponding position (x, y) .

$$\begin{aligned} & \|R - (s \cdot T(D_{\text{best}_i}) + o \cdot E)\|_k \\ &= \min_j \left\{ \min_{s, o \in R, |s| < 1} \|R_i - (T(s \cdot d_j) + o \cdot E)\|_k \right\}. \end{aligned} \tag{3}$$

But in fact, Eq. (3) needs large computational time that cannot be applied. In real application, Eq. (4) is applied to instead of Eq. (3). So, in application, Eq. (4) is usually applied to find affine parameters of best mapping from domain block to each R_i with minimum loss.

$$\begin{aligned} & \min_j \left\{ \min_{s, o \in R} \|R_i - (s \cdot T(D_j) + o \cdot E)\|_k \right\} \\ &= \min_{s, o \in R} \|R_i - (s_i \cdot T(D_{m_i}) + o_i \cdot E)\|_k. \end{aligned} \tag{4}$$

The transformation iterates convergently when this transformation conforms to Definition 2, which means the final iteration can be iterated by any original input. In this way, only parameters of CAT are needed to be recorded as final compression file when CAT can be found in an image. When a series of CAT parameters stored the transformation parameters of all pixels of compressing image, the original image can be constituted with sub-images iterated by all CAT parameters in the uncompressing process. This is fractal image compression method. The process of fractal image compression is presented in Fig. 1 and Algorithm 1. Fractal uncompressing method is presented in Algorithm 2.

In Algorithms 1 and 2, I means original image, n means the size of I is $n \times n$, r means the size of range image is $r \times r$ ($r|n$), $p = n/r$, d means the size of domain image is $d \times d$ ($d > r$), F means the compressing file, s means the scaling of luminance, o means offset of luminance, (x, y) mean the starting position of affine transformation, direction means the eight types of equiangular transformations (with transformation angle $\frac{n\pi}{4}$ $n = 0, \dots, 8$), T means the iterating time, D means the uncompressing images.

From Fig. 1 and Algorithm 1, we find that in the fractal image compression method, compressing

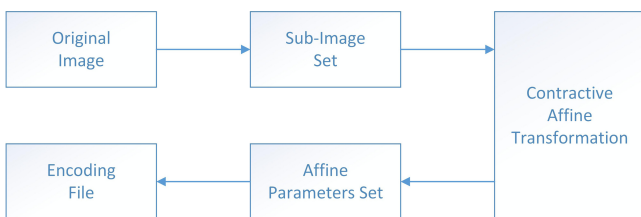


Fig. 1 Processes of fractal image encoding.

Algorithm 1 Classical Fractal Compression Method.

Input: $I(n \times n); r; d$

Output: F

- 1: To find every segmented block R_k of I ($k = 1, 2, \dots, p^2$), where $\forall i, j | i \neq j \rightarrow R_i \cap R_j = \emptyset$ and $\cup_{1 \leq i \leq p} R_i = I$, Meanwhile, $\forall i, j | \text{scale of } R_i = \text{scale of } R_j = r \times r$.
 - 2: Find all part-repeatable D_p in I . The size of all D_p is d .
 - 3: **for** every R_k **do**
 - 4: Find the best affine transformation of R_k in D_p , and store it as a vector in the affine transformation table.
 - 5: Output r, d together affine transforming table as compression file F .
 - 6: **return** F
-

Algorithm 2 Classical Fractal Uncompressing Method.

Input: F

Output: D

- 1: To extract r, d and affine transforming table from F . Apply a blank image as R .
 - 2: **while** The affine transforming table is not finished **do**
 - 3: To extract the k^{th} vector from affine transforming table, which is the best affine transformation of R_k . To locate D_k with position (x, y) in R . To iterate R_k by D_k with size $d \times d$ by stored corresponding s_k and o_k . Then, to process equiangular transformation by stored direction.
 - 4: To collage all rectangular areas to an image D . Then, let $R = D$, iterating step 2 until iterating time = T .
 - 5: **return** D
-

image is first segmented to non-overlapping range blocks with the same size. Then, in order to reach the best match, all domain blocks with larger size are contracted to the same size with every range block. Usually, domain blocks are contracted by means of neighbor pixels to match the size of range blocks because domain blocks have double size to range blocks.

Though fractal compression method has high compression ratio and well uncompressing result, the bottleneck is high compressing time. So, in this

paper, a novel fractal compression method is provided to speed up the compressing time.

3.2. Distance Clustering in High Dimension

In one space with high dimension S^n , where n denotes dimension of the space, an original circle C is a circle where the distance is unit 1 between original point O and every point P in its surface C^* . It means that we can construct n -tuple $(\alpha_1, \dots, \alpha_n)$ for every point P in C^* where α_i denotes angle between OP and the i th axis of dimension. Radius is not needed because C^* is surface of original circle.

Then, a clustering method is presented based on a given threshold T . Defining a point on surface $P(\alpha_1, \dots, \alpha_l)$ and another point $P'(\alpha_1 + \Delta\alpha_1, \dots, \alpha_l + \Delta\alpha_l)$, distance $D(P, P')$ between these two points are computed as follows:

$$D(P, P') = \sqrt{\sum_{i=1}^l (\cos(\alpha_i + \Delta\alpha_i) - \cos \alpha_i)^2}. \quad (5)$$

Further, applying Eq. (6) in Eq. (5), we reach Eq. (7).

$$\begin{aligned} &\cos(\alpha_i + \Delta\alpha_i) - \cos \alpha_i \\ &= -2 \sin\left(\alpha_i + \frac{\Delta\alpha_i}{2}\right) \sin\left(\frac{\Delta\alpha_i}{2}\right), \end{aligned} \quad (6)$$

$$D(P, P') = 2 \sqrt{\sum_{i=1}^l \sin^2\left(\alpha_i + \frac{\Delta\alpha_i}{2}\right) \sin^2\left(\frac{\Delta\alpha_i}{2}\right)}. \quad (7)$$

Without any constraint, we compute partial derivative $\frac{\partial D}{\partial \alpha_i} = 0$ of D to reach its extremum, which implies the relationship between $\Delta\alpha_i$ and D .

$$\frac{\partial D}{\partial \alpha_i} = \frac{\sin^2\left(\frac{\Delta\alpha_i}{2}\right) \sin(2\alpha_i + \Delta\alpha_i)}{\sqrt{\sum_{i=1}^n \sin^2\left(\alpha_i + \frac{\Delta\alpha_i}{2}\right) \sin^2\left(\frac{\Delta\alpha_i}{2}\right)}} = 0. \quad (8)$$

Then, assuming that $\Delta\alpha_i = \epsilon \neq 0$ for all $i = 1, \dots, n$, it is admittedly that $\sin\left(\frac{\Delta\alpha_i}{2}\right) \neq 0$ and $\sin(2\alpha_i + \epsilon) = 0$. In this way, we have that $\alpha_i = \frac{k\pi - \epsilon}{2}$. Further, with consideration of $\frac{\partial^2 D}{\partial \alpha_i^2}$, we have Eq. (9). Intersection between solutions of Eqs. (8) and (9) is the maximum of Eq. (5).

$$\frac{\partial^2 D}{\partial \alpha_i^2} = \sin^2\left(\frac{\epsilon}{2}\right) \cdot \frac{2 \cos A \cdot F(\alpha_i) - \sin A \cdot F'(\alpha_i)}{F^2(\alpha_i)}$$

$$\begin{aligned} &= \sin^2\left(\frac{\epsilon}{2}\right) \\ &\cdot \frac{4 \cos A \cdot F^2(\alpha_i) - \sin^2 A \cdot \sin^2\left(\frac{\epsilon}{2}\right)}{F^3(\alpha_i)} < 0, \\ &\text{where } A = 2\alpha_i + \epsilon. \end{aligned} \quad (9)$$

Solution of Eq. (9) can be solved by $\cos(2\alpha_i + \epsilon) < 0$ when assuming $\sin^2\left(\frac{\epsilon}{2}\right) < 0$. The solution is $|\alpha_i + \frac{\epsilon}{2}| \in (k\pi + \frac{\pi}{4}, k\pi + \frac{3\pi}{4})$. So intersection between solutions of Eqs. (8) and (9) is presented in Eq. (10).

$$\alpha_i = k\pi + \frac{\pi - \epsilon}{2}. \quad (10)$$

Applying Eq. (10) into Eq. (7), we have

$$\max \arg_{\alpha_i} D(P, P') = 2\sqrt{l} \sin\left(\frac{\Delta\alpha_i}{2}\right) \approx \sqrt{l}\Delta\alpha_i,$$

which means $D(P, P') < \sqrt{l}\Delta\alpha_i$ when $\Delta\alpha_i$ is positive for any α_i . Further, $D(P, P')$ approximately equals $\sqrt{l}\Delta\alpha_i$ when $\Delta\alpha_i$ is tiny enough.

3.3. A Novel Fractal Compression Method

In this case, classical fractal compression method can be preprocessed to speed up the compressing time. In fact, this novel method preprocesses the segmented range blocks in high dimension first.

After range blocks are segmented as classical fractal compression method, corresponding range blocks are linear transformed. In mathematical words, defining a range block $R = (r_1, r_2, \dots, r_k)$ where $r_i = (r_{1i}, r_{2i}, \dots, r_{ki})$ in linear transformation space, a linear transformation $R' = (r'_1, r'_2, \dots, r'_k)$ where $r'_i = (r'_{1i}, r'_{2i}, \dots, r'_{ki})$.

So if R' can be computed easily, R' can be applied instead of R when computing the distance between R and D . Identifying domain blocks D and range blocks R , Eq. (11) is applied to reach parameters of CAT between all D and R blocks.

$$\min_{s,o} E(R, D) = \|R - (s \cdot D + o \cdot I)\|_p. \quad (11)$$

In this paper, r'_{ij} is constructed by Eq. (12).

$$r'_{ij} = \begin{cases} r_{ij} & \text{if } i = j = k, \\ \frac{r_{ij} - r_{kk}}{\sqrt{\sum_{j=1}^{k-1} (r_{ij} - r_{kk})^2}} & \text{otherwise.} \end{cases} \quad (12)$$

In order to enhance further computation, one additional dimension is added in R' . The additional dimension $r^* = \frac{r_{ij}}{r'_{ij}}$, which applies to store parameter of linear transform.

So, after the preprocessing procedure, transformation of range block $R^* = (R', r^*)$. Then, defining $E'(R', D')$ as loss between R' and D' , where D' is transformation of D . The transformation is similar to mapping from R to R' .

In this way, we can provide the novel fractal compression method in Algorithm 3.

Algorithm 3 A Novel Fractal Compression Method.

Input: $I(n \times n); r; d$

Output: F

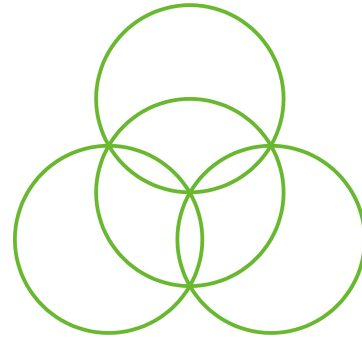
- 1: Find every segmented block R_k of I like Alg.1 where size of R_k is $r \times r$. Preprocess R_k to $R_k^* = (R'_k, r_k^*)$.
 - 2: Find all part-repeatable D_p in I . The size of all D_p is $d \times d$. Compress D_p to $D_{p'}$ by mean of every square (size 2×2) to preprocess $D_{p'}$ to $D_{p'}^* = (D'_{p'}, d_{p'}^*)$.
 - 3: **for** every R'_k **do**
 - 4: Hashing R'_k to exist unit cambered surfaces with radius $r \cdot \varepsilon$ on half spherical surface, where ε is a given minimum threshold. Then matching computation for each point is only for those points on a sub-sphere like Fig. 2a, where the matching point is in the center round.
 - 5: Add r and d to the affine transforming table and Output them as compressing file F .
 - 6: **return** F
-

Also, we can use sub-sphere like Fig. 2b. Both Figs. 2a and 2b show plane mapping of the spherical surface. It is admittedly that both Figs. 2a and 2b show covering of the plane. In this paper, we use Fig. 2a as our covering unit. A diagrammatic sketch of hashing structure is just like Fig. 2c, which shows mapping from cross-section to semi-sphere in the three-dimensional space.

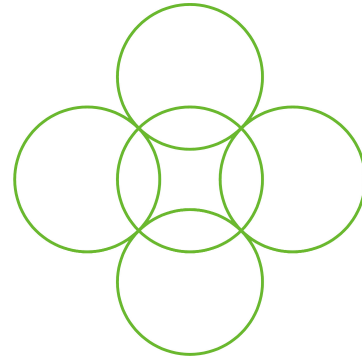
3.4. Properties of Presented Fractal Compression Method

Here, we have the following assumption to compute and prove properties of the proposed fractal compression method.

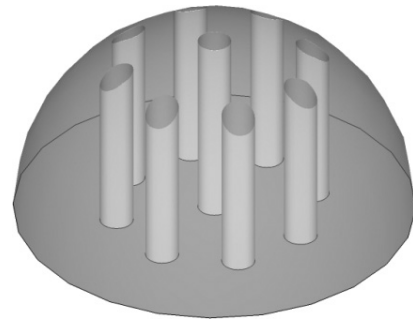
In this paper, $\alpha = (\alpha_1, \dots, \alpha_l)$, $\beta = (\beta_1, \dots, \beta_l)$ are two vectors in one-dimensional space ($\alpha_l \geq 0$, $\beta_l \geq 0$), $\alpha' = (\alpha'_1, \dots, \alpha'_l)$ and $\beta' = (\beta'_1, \dots, \beta'_l)$ are mappings of α and β on surface of original unit circle where $\alpha'_i = \frac{\alpha_i}{\alpha_{sum}}$, $\beta'_i = \frac{\beta_i}{\beta_{sum}}$, $\alpha_{sum} = \|\alpha\|_p$, $\beta_{sum} = \|\beta\|_p$.



(a) Matching area on surface.



(b) Another matching area on surface.



(c) A sample for three-dimensional space.

Fig. 2 Plane mapping of spherical surface.

Then, applying bivariate function $D(\alpha, \beta) = \alpha - (s \cdot \beta + o \cdot I)$, we have Theorem 3 to research relationship between solutions of $\min D(\alpha, \beta)$ and $\min D(\alpha', \beta')$.

Theorem 3. Solution of $\min_{s,o} D(\alpha', \beta') = \|\alpha' - (s \cdot \beta' + o \cdot I)\|_p$ is $\frac{s^* \cdot \beta_{sum}}{\alpha_{sum}}$ and $\frac{o^*}{\alpha_{sum}}$ if and only if solution of $\min_{s,o} D(\alpha, \beta) = \|\alpha - (s \cdot \beta + o \cdot I)\|_p$ is s^* and o^* .

Proof. First, it is assumed that solution of $\min_{s,o} D(\alpha', \beta') = \|\alpha' - (s \cdot \beta' + o \cdot I)\|_p$ is s_1, o_1 and solution of $\min_{s,o} D(\alpha, \beta) = \|\alpha - (s \cdot \beta + o \cdot I)\|_p$ is s_2, o_2 .

Then, applying reduction to absurdity, it is assumed that $s_1 \neq \frac{\beta_{\text{sum}}}{\alpha_{\text{sum}}} \cdot s_2$, $o_1 \neq \frac{1}{\alpha_{\text{sum}}} \cdot o_2$ and $\|\alpha' - (s_1 \cdot \beta' + o_1 \cdot I)\|_p < \|\alpha' - (\frac{\beta_{\text{sum}} \cdot s_2}{\alpha_{\text{sum}}} \cdot \beta' + \frac{o_2}{\alpha_{\text{sum}}} \cdot I)\|_p$.

Then, we have that $\|\alpha - (s_1 \cdot \frac{\alpha_{\text{sum}}}{\beta_{\text{sum}}} \cdot \beta + o_1 \cdot \alpha_{\text{sum}} \cdot I)\|_p = \alpha_{\text{sum}} \cdot \|\alpha' - (s_1 \cdot \beta' + o_1 \cdot I)\|_p < \alpha_{\text{sum}} \cdot \|\alpha' - (\frac{\beta_{\text{sum}} \cdot s_2}{\alpha_{\text{sum}}} \cdot \beta' + \frac{o_2}{\alpha_{\text{sum}}} \cdot I)\|_p = \|\alpha - (s_2 \cdot \beta + o_2 \cdot I)\|_p$, which means $s_1 \cdot \frac{\alpha_{\text{sum}}}{\beta_{\text{sum}}}$ and $o_1 \cdot \alpha_{\text{sum}}$ makes $D(\alpha, \beta)$ smaller than s_2 and o_2 . This contradicted the premise that s_2 and o_2 is solution of $\min_{s,o} D(\alpha, \beta) = \|\alpha - (s \cdot \beta + o \cdot I)\|_p$.

So $\frac{\beta_{\text{sum}}}{\alpha_{\text{sum}}} \cdot s_2$ and $\frac{1}{\alpha_{\text{sum}}} \cdot o_2$ are the solutions of $\min_{s,o} D(\alpha', \beta') = \|\alpha' - (s \cdot \beta' + o \cdot I)\|_p$, when s_2 and o_2 are the solutions of $\min_{s,o} D(\alpha, \beta) = \|\alpha - (s \cdot \beta + o \cdot I)\|_p$.

Then, with similar proof, we have that s_2 and o_2 are the solutions of $\min_{s,o} D(\alpha, \beta) = \|\alpha - (s \cdot \beta + o \cdot I)\|_p$ when $\frac{\beta_{\text{sum}}}{\alpha_{\text{sum}}} \cdot s_2$ and $\frac{1}{\alpha_{\text{sum}}} \cdot o_2$ are the solutions of $\min_{s,o} D(\alpha', \beta') = \|\alpha' - (s \cdot \beta' + o \cdot I)\|_p$.

Summarizing all the above, Theorem 3 is proved. \square

Theorem 3 provide the method to get s and o of minimum $D(\alpha, \beta)$ from s' and o' of minimum $D(\alpha', \beta')$. Then we have Theorem 5 to prove correctness of presented algorithms. First, Definition 4 is presented to describe the *nearest* between two vectors.

Definition 4. Vector α_i in vector set $\{\alpha_j\}_{j=1,2,\dots}$, is the nearest to vector α if there exists $k_i \geq 0$, $s_i, o_i \in \mathbf{R}$ that makes $D(\alpha, k_i \cdot \alpha_i)$ is minimum in set $\{D(\alpha, k_j \cdot \alpha_j)\}_{j=1,2,\dots}$.

Definition 4 defines the *nearest* where there exists positive real number k_i , real numbers s_i and o_i makes $D(\alpha, k_i \cdot \alpha_i)$ is minimum in all vectors. In this case, Theorem 5 is presented to prove the property *nearest* is static between vectors and unchanged by modulus of vectors, which confirms the correctness of presented algorithms.

Theorem 5. A vector α_i is nearest to α in set $\{\alpha_j\}_{j=1,2,\dots}$, if and only if there exist $k_i \in \mathbf{R}_+$ that makes $k_i \cdot \alpha_i$ is nearest to $k \cdot \alpha$ in vector set $\{k_j \cdot \alpha_j\}_{j=1,2,\dots}$, for every $k \in \mathbf{R}_+$.

Proof. Assuming α_i is nearest to α in vector set $\{\alpha_i\}_{i=1,2,\dots}$, we have s_i and o_i that makes $\|\alpha - (s_i \cdot \alpha_i + o_i \cdot I)\|_p < \|\alpha - (s_j \cdot \alpha_j + o_j \cdot I)\|_p$ for all $j \neq i$ where s_j and o_j are the solution of $\min_{s_j,o_j} D(\alpha, \alpha_j) = \|\alpha - (s_j \cdot \alpha_j + o_j \cdot I)\|_p$.

Applying reduction to absurdity, it is assumed that there exists s_j, o_j, k and k_j that make $\|k \cdot \alpha - (s_j \cdot k_j \cdot \alpha_j + o_j \cdot I)\|_p < \|k \cdot \alpha - (s'_i \cdot k'_i \cdot \alpha_i + o'_i \cdot I)\|_p$ for all k'_i, s'_i and o'_i .

Then, we have that $k \cdot \|\alpha - (s_j \cdot \frac{k_j}{k} \cdot \alpha_j + \frac{o_j}{k} \cdot I)\|_p = \|k \cdot \alpha - (s_j \cdot k_j \cdot \alpha_j + o_j \cdot I)\|_p < \|k \cdot \alpha - (s'_i \cdot k'_i \cdot \alpha_i + o'_i \cdot I)\|_p = k \cdot \|\alpha - (s'_i \cdot \frac{k'_i}{k} \cdot \alpha_i + \frac{o'_i}{k} \cdot I)\|_p = k \cdot \|\alpha - (s_i \cdot \alpha_i + o_i \cdot I)\|_p$ when let $s'_i \cdot \frac{k'_i}{k} = s_i$ and $\frac{o'_i}{k} = o_i$. This contradicted the premise that $\|\alpha - (s_i \cdot \alpha_i + o_i \cdot I)\|_p < \|\alpha - (s_j \cdot \alpha_j + o_j \cdot I)\|_p$ for all $j \neq i$.

So there exist $k_i \in \mathbf{R}_+$ that makes $k_i \cdot \alpha_i$ is the nearest to $k \cdot \alpha$ in vector set $\{k_j \cdot \alpha_j\}_{j=1,2,\dots}$ for every $k \in \mathbf{R}_+$.

Then, with similar proof, we have that α_i is the nearest to α when there exist $k_i \in \mathbf{R}_+$ that makes $k_i \cdot \alpha_i$ is the nearest to $k \cdot \alpha$ in vector set $\{k_j \cdot \alpha_j\}_{j=1,2,\dots}$ for every $k \in \mathbf{R}_+$.

Summarizing all the above, Theorem 5 is proved. \square

From Theorem 5, we have the correctness of the presented fractal compression method. Then, we have Theorem 7 to present the computation of the proposed fractal compression method. First, Lemma 6 is applied to present computational time complexity of the proposed fractal image compression method.

Lemma 6. Computational time complexity of proposed fractal image compression method is $O(n^4 \varepsilon^2 + 1/\varepsilon^2)$.

Proof. We have Fig. 2a to present our matching area in hashing mapping. It means one vector only needs to compare its matching vectors in its matching area.

Assuming radius of single circle is $r \cdot \varepsilon$, we have covering area of Fig. 2a is $(\pi + 3\sqrt{3})r^2 \cdot \varepsilon^2$. So, average covering area of each single circle is $(\frac{\pi}{3} + \sqrt{3})r^2 \cdot \varepsilon^2$.

So in Lemma 6, when assuming all vectors are well distributed in the whole half high dimension sphere, we have that the vectors' number in each unit is $\frac{n_d(\pi+3\sqrt{3})r^2 \cdot \varepsilon^2}{\pi R^2}$, where $R = 1$ is radius of sphere, $n_d = (n + 1 - 2r)^2$ is number of patterned vectors (domain blocks), r^2 is number of components in each vector. This means the total computational time is $\frac{n_d n_r (\pi+3\sqrt{3})r^2 \cdot \varepsilon^2}{\pi R^2}$ for all vectors, where $n_r = \frac{n^2}{r^2}$ is number of original vectors (range blocks). However, total computational time of the

data structure spent is $\frac{\pi R^2}{(\frac{\pi}{3} + \sqrt{3})r^2 \cdot \varepsilon^2}$ for the hashing structure.

Computational time complexity can be achieved by using the simplified computational time in Eqs. (13) and (14).

$$\begin{aligned} & O\left(\frac{n_d n_r (\pi + 3\sqrt{3})r^2 \cdot \varepsilon^2}{\pi R^2}\right) \\ &= O\left((n+1-2r)^2 \frac{n^2}{r^2} r^2 \varepsilon^2\right) \\ &= O(n^4 \varepsilon^2), \end{aligned} \quad (13)$$

$$O\left(\frac{\pi R^2}{(\frac{\pi}{3} + \sqrt{3})r^2 \cdot \varepsilon^2}\right) = O\left(\frac{1}{\varepsilon^2}\right). \quad (14)$$

So, total computational time complexity is $O(n^4 \varepsilon^2) + O(\frac{1}{\varepsilon^2}) = O(n^4 \varepsilon^2 + \frac{1}{\varepsilon^2})$.

Lemma 6 is proved. \square

Then, when assuming $\varepsilon = \lambda/n$ is inverse proportion of n , we know minimum of total computational complexity of proposed method is $O((n-1+r)^2 + n^2) = O(n^2)$. So, we have Theorem 7.

Theorem 7. *Optimal computational time complexity of proposed fractal image compression method is $O(n^2)$ if and only if ε is inverse proportion of n .*

Since computational time complexity of classical fractal compression method is $O(n^4)$, we know the proposed fractal compression method is a faster algorithm.

4. EXPERIMENTAL RESULTS AND ANALYSIS

Here, we present six figures with same size in Fig. 3 as the original images in our experiment, where Fig. 3a is Lena, Fig. 3b is bird, Fig. 3c is baboon, Fig. 3d is Barbara, Fig. 3e is pepper, Fig. 3f is fruit.

In our experiment, we use $n = 256$, $r = 4$, $d = 8$. Then, optimal $\lambda = 0.615$ can be computed.

Then, we use the proposed fractal compression method to compress these original figures. Some uncompressing results are presented in Fig. 4, where each left sub-figure is an uncompressing image with one iterating time, each middle figure is an uncompressing image with five iterating times and each right figure is an uncompressing image with 10 iterating times.

It is admittedly that the compression ratio of the proposed method is similar to the classical methods

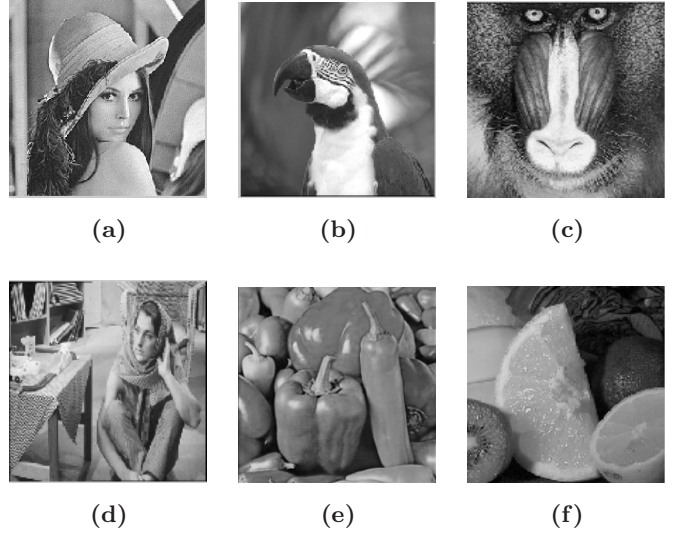


Fig. 3 Original experimental images.



(a) Uncompressing images for image Fig. 3a



(b) Uncompressing images for image Fig. 3b



(c) Uncompressing images for image Fig. 3c



(d) Uncompressing images for image Fig. 3d

Fig. 4 Some uncompressing images of Fig. 3 with one (left), five (middle) and ten (right) iterating time(s).

Table 1 Comparing Quality Between Fractal Compression Methods.

Image	PSNR (dB)					
	HFICFP	FIEPAE	FIENON	PFI	FICPQ	FIEHC
Lena	30.4	26.5	31.3	25.8	30.3	31.7
Bird	31.7	30.6	31.8	26.1	31.1	31.1
Baboon	23.1	22.5	23.3	20.9	21.7	24.3
Barbara	25.1	23.0	25.0	21.2	25.2	25.8
Pepper	29.7	25.6	26.9	27.4	28.9	29.1
Fruit	32.6	30.7	31.4	32.1	32.8	31.5
Average	28.8	26.5	28.3	25.6	28.3	28.9

Table 2 Comparison of Compression Time Between Fractal Compression Methods.

Image	Time (s)					
	HFICFP	FIEPAE	FIENON	PFI	FICPQ	FIEHC
Lena	668	659	692	657	665	632
Bird	676	641	711	669	671	617
Baboon	701	668	704	660	652	684
Barbara	685	634	699	641	700	625
Pepper	691	665	695	654	683	659
Fruit	680	657	708	650	646	601
Average	684	654	702	655	670	636

because the proposed method only has one additional storage unit in each compression unit to store ratio between original vector and vector in unit spherical surface. Also, compressing quality of the proposed method is similar to that of the classical methods because almost all vectors can be patterned in our experiment. However, the proposed method shows better performance in compressing time.

Then, we presented the comparison of compressing quality and time between other respected methods and our results in Tables 1 and 2, which are HFICFP,²³ FIEPAE,³⁰ FIENON,³² PFI,³³ FICPQ³⁴ and the proposed method (FIEHC).

In Table 1, we find that the FIEHC method shows best average quality for all six experiments. Especially, FIEHC has better PSNR in figures baboon and barbara, which are hard to encode with fractal compression method. It proves FIEHC is an effective fractal image compression method.

Also, we can find some compressing quality (PSNR) of our method is lower than others. We checked it and find it is because of calculation error in mapping, which is due to the experimental result not corresponding to theoretical analysis. But this problem can be solved in computers with higher computational precision.

In Table 2, we also find FIEHC shows best average time for all six experiments. However, FIEHC shows best performance in each figure except figures baboon and peppers. This is because there are many range and domain vectors mapping into a few constructed areas, which increases computational times.

Also, we find that the computational time is not lesser than the other methods like the theoretical analysis. This is because the structure method on high dimensional sphere surface spent much memory space and time. But this problem can be solved if GPU is used in the constructing process.

5. CONCLUSIONS

This paper provided a novel fractal image compression method by using distance clustering on high dimensional sphere surface. This method was provided by a novel transformation strategy for compressing image, which can be mapped in constructed structure on high dimensional sphere surface. Then, in order to ensure the correctness and effectiveness, theoretical proofs for this fractal image compression method were provided. Finally, experimental results also verified the conclusion of this paper.

Next, it will be necessary to research properties of image blocks, which can be used to define direction of segmentation and speed up the distributed algorithm. Also, further accelerating compression method will be investigated, such as the one based on quantum computing.

ACKNOWLEDGMENTS

The research is supported by Natural Science Foundation of Inner Mongolia (Grant No. 2014BS0606), National Natural Science Foundation of China (Grant No. 61502254) and EU FP7.

REFERENCES

1. S. Saha, Image compression from dct to wavelets: A review, *ACM Crossroads* **6**(3) (2000) 12–21.
2. K. R. Rao and P. Yip, *Discrete Cosine Transform: Algorithms, Advantages, Applications* (Academic Press, Cambridge, MA, 2014).
3. D. A. Huffman, A method for the construction of minimum-redundancy codes, *Proc. IRE* **40**(9) (1952) 1098–1101.
4. J. M. Shapiro, Embedded image coding using zerotrees of wavelet coefficients, *IEEE Trans. Signal Process.* **41**(12) (1993) 3445–3462.
5. B. B. Mandelbrot and R. Pignoni, *The fractal geometry of Nature* (W. H. Freeman, NY, 1983).
6. K. Falconer, *Fractal Geometry: Mathematical Foundations and Applications* (John Wiley & Sons, NY, 2004).
7. K. Goebel and W. Kirk, A fixed point theorem for asymptotically nonexpansive mappings, *Proc. Am. Math. Soc.* **35**(1) (1972) 171–174.
8. F. Ilday, J. Buckley, W. Clark and F. Wise, Self-similar evolution of parabolic pulses in a laser, *Phys. Rev. Lett.* **92**(21) (2004) 213902.
9. J.-R. He, L. Yi and H.-M. Li, Self-similar propagation and asymptotic optical waves in nonlinear waveguides, *Phys. Rev. E* **90**(1) (2014) 013202.
10. D. R. Smart, *Fixed Point Theorems* (CUP Archive, Cambridge, 1980).
11. M. F. Barnsley and A. E. Jacquin, Application of recurrent iterated function systems to images, *Visual Communications and Image Processing '88: Third in a Series* (International Society for Optics and Photonics, USA, 1988), pp. 122–131.
12. A. E. Jacquin, Image coding based on a fractal theory of iterated contractive image transformations, *IEEE Trans. Image Process.* **1**(1) (1992) 18–30.
13. T. Bedford, F. Dekking, M. Breeuwer, M. Keane and D. Van Schooneveld, Fractal coding of monochrome images, *Signal Process. Image Commun.* **6**(5) (1994) 405–419.
14. I. K. Kim and R.-H. Park, Still image coding based on vector quantization and fractal approximation, *IEEE Trans. Image Process.* **5**(4) (1996) 587–597.
15. C.-S. Kim, R.-C. Kim and S.-U. Lee, Fractal coding of video sequence using circular prediction mapping and noncontractive interframe mapping, *IEEE Trans. Image Process.* **7**(4) (1998) 601–605.
16. H. T. Chang and C. J. Kuo, Iteration-free fractal image coding based on efficient domain pool design, *IEEE Trans. Image Process.* **9**(3) (2000) 329–339.
17. C.-S. Tong and M. Pi, Fast fractal image encoding based on adaptive search, *IEEE Trans. Image Process.* **10**(9) (2001) 1269–1277.
18. C.-M. Lai, K.-M. Lam and W.-C. Siu, A fast fractal image coding based on kick-out and zero contrast conditions, *IEEE Trans. Image Process.* **12**(11) (2003) 1398–1403.
19. R. Distasi, M. Nappi and D. Riccio, A range/domain approximation error-based approach for fractal image compression, *IEEE Trans. Image Process.* **15**(1) (2006) 89–97.
20. X.-Y. Wang and S.-G. Wang, An improved no-search fractal image coding method based on a modified gray-level transform, *Comput. Graph.* **32**(4) (2008) 445–450.
21. X.-Y. Wang, Y.-X. Wang and J.-J. Yun, An improved no-search fractal image coding method based on a fitting plane, *Image Vis. Comput.* **28**(8) (2010) 1303–1308.
22. J.-H. Jeng, C.-C. Tseng and J.-G. Hsieh, Study on huber fractal image compression, *IEEE Trans. Image Process.* **18**(5) (2009) 995–1003.
23. J. Lu, Z. Ye and Y. Zou, Huber fractal image coding based on a fitting plane, *IEEE Trans. Image Process.* **22**(1) (2013) 134–145.
24. J. Wang and N. Zheng, A novel fractal image compression scheme with block classification and sorting based on pearson's correlation coefficient, *IEEE Trans. Image Process.* **22**(9) (2013) 3690–3702.
25. S. Bhavani and K. G. Thanushkodi, Comparison of fractal coding methods for medical image compression, *IET Image Process.* **7**(7) (2013) 686–693.
26. X.-Y. Wang and D.-D. Zhang, Discrete wavelet transform-based simple range classification strategies for fractal image coding, *Nonlinear Dyn.* **75**(3) (2014) 439–448.
27. S. Liu, X. Cheng, C. Lan, W. Fu, J. Zhou, Q. Li and G. Gao, Fractal property of generalized m-set with rational number exponent, *Appl. Math. Comput.* **220** (2013) 668–675.
28. S. Liu, W. Fu, W. Zhao, J. Zhou and Q. Li, A novel fusion method by static and moving facial capture, *Math. Probl. Eng.* **2013** (2013) 503924.
29. S. Liu, W. Fu, H. Deng, C. Lan and J. Zhou, Distributional fractal creating algorithm in parallel environment, *Int. J. Distrib. Sens. Netw.* **9**(9) (2013) 1–8.

30. S. Liu, W. Fu, L. He, J. Zhou and M. Ma, Distribution of primary additional errors in fractal encoding method, *Multimed. Tools Appl.* **76**(4) (2017) 5787–5802.
31. S. Liu, Z. Zhang, L. Qi and M. Ma, A fractal image encoding method based on statistical loss used in agricultural image compression, *Multimed. Tools Appl.* **75**(23) (2016) 15525–15536.
32. H.-N. Chen, K.-L. Chung and J.-E. Hung, Novel fractal image encoding algorithm using normalized one-norm and kick-out condition, *Image Vis. Comput.* **28**(3) (2010) 518–525.
33. Y.-L. Lin, Robust estimation of parameter for fractal inverse problem, *Comput. Math. Appl.* **60**(7) (2010) 2099–2108.
34. Q. Wang and S. Bi, Prediction of the psnr quality of decoded images in fractal image coding, *Math. Probl. Eng.* **2016** (2016) 1–13.