

# Sensing and Controlling with Markov Process for Locally Independent Fractal Image

Li-liang Zhang,\* Ting-cheng Chang, and Yan-ming Mao

Computer and Information Engineering, Ningde Normal University, Ningde, Fujian 352100, China

(Received December 31, 2019; accepted April 29, 2020)

**Keywords:** iterated function system, Markov model, fractal image, local control

We used the Markov process to impose parameter perturbation on affine transformations to overcome the self-similarity limitations of the fractal attractor of the classic iterated function system (IFS) and construct a fractal image with irregular shapes and features. Then, a control method for the IFS system for fractal image generation was proposed. This method decomposes the original IFS into several independent local subsystems. Then, we defined a transition probability matrix of the Markov process for each of the different local subsystems, and carried out image shape operation of a perturbation function to increase the control of the fractal image. This method can construct colorful fractal images effectively through computer image generation.

## 1. Introduction

Fractal geometry uses its unique self-similarity to construct protean fractal images with an arbitrary high resolution, which has attracted great attention from the public.<sup>(1)</sup> Fractal image integrates art with science. It also has various technological applications, including sensor technologies.<sup>(2,3)</sup> However, fractal images have mainly been used for their artistic value in a broad range of industries, including packaging and printing, fashion and fabrics, embroidery, ceramics, architecture, and advertisements.<sup>(4–6)</sup> Nowadays, science and aesthetic impressions are generated by numerical calculation by computers.<sup>(7,8)</sup> Advanced computer graphics technology creates designs for various types of artistic patterns automatically. This has become the mainstream of modern art pattern design.

Research on fractal images for art pattern designs has shown significant progress.<sup>(9,10)</sup> However, it is not easy to control the generation of fractal art images. In order to improve the classic iterated function system (IFS) used to generate fractal art images, we applied the Markov stochastic process theory to fractal modeling using the IFS. On the basis of the multilevel decomposition of the IFS affine transformation system, the IFS is decomposed into multiple independent subsystems to obtain IFS iterated codes of different local IFSs and desired depths.

In accordance with preset control rules, shape changes of different local fractal images are controlled to realize the shape evolution of each subsystem image. This is based on a local independent image operation that carries out deformation processing on a specific local

---

\*Corresponding author: e-mail: 13859603598@139.com  
<https://doi.org/10.18494/SAM.2020.2792>

subimage without affecting other local subimages of the image. We use a Markov random sequence to adjust the affine transformation family of the IFS local subsystem and combine the parameter perturbation method. In this way, we can generate a large number of changeable fractal images by designing the shape modeling rules in an image drawing process.

## 2. Relevant Works

The IFS is a representative system that can be used as a basis to simulate the scenery of a fractal nature to describe common fractal shapes, and its theory and practice have been widely applied in many fields.<sup>(11,12)</sup> As the shape of an ordinary IFS fractal attractor is very regular, the attractor often seems to be “rigid” and does not significantly change when the IFS attractor is directly used to simulate a natural landscape. Garcla introduced a mutually recursive function system (MRFS), a popular form of the IFS, to obtain a balance between order and chaos in the IFS attractor image.<sup>(13)</sup> Its basic idea is to use the affine transformation of IFS variables during the computation process of successive iterations. As different changes are used for each recursion, attractors with different rules or irregular forms can be obtained.

Several enhanced attractor control methodologies have been proposed through the development of the IFS since the MRFS was introduced. A modified method was introduced in which the parameters of the transformation depend on the number of iterations. This method combined the IFS with a new approach to model compound objects.<sup>(14)</sup> Sherman and Hart, and Radu introduced a Markov process into the study of an IFS to strengthen the control of the fractal attractor image, which has great practical value.<sup>(15,16)</sup> An IFS with Markov characteristics has also been proposed, which overcomes the limitations of the traditional IFS methods to some extent. Various studies have proposed a more generalized iterated function system (MIFS), and many concrete applications have been reported. Sharif Ullah *et al.* proposed the design for manufacturing (DFM) of fractals created by a random walk, that is, the DFM of an IFS-created fractal called Barnsley’s fern leaf. The IFS for creating virtual models of a fern leaf used a set of four strictly contracting affine mappings.<sup>(17)</sup>

Stepien used a modular method to construct a fractal model of natural objects with spiral structures, such as taken horns and seashells. The model was composed of a base module and an IFS with a compressed set (IFSc), where the IFSc attractor comprised a gradually shrinking copy sequence.<sup>(18)</sup> Zhang put forward a method of modeling the fractal attractor of a Markov IFS based on a transition probability matrix that involved the multiple decomposition of the MIFS affine transformation. Then, he used a Markov transition probability matrix to conduct deformation processing of the local subimages of the attractor.<sup>(19)</sup> Zhang and Pan, and Flavian and Radu proposed an IFS based on the hidden Markov model. They used the hidden Markov random process to control the modeling and coloring of fractal images, and then drew them with an irregular shape and color.<sup>(20,21)</sup> Zhuang *et al.* proposed a texture-based IFS (TIFS) that introduced a method of texture synthesis into fractal image designs as a color rendering rule of fractal images to achieve a better rendering effect.<sup>(22)</sup>

### 3. Local Control Algorithm for IFS Fractal Image

#### 3.1 Overview of Markov iterative function system

**Definition 1:** The mapping family  $\{X; \omega_1, \omega_2, \dots, \omega_N\}$  composed of complete metric space  $(X, d)$  and  $N$  compressed mappings  $\omega_i: X \rightarrow X$  is called an IFS. For each mapping  $\omega_i$ , there is a probability  $p_i > 0$ ,  $\sum_{i=1}^N p_i = 1$ .  $x_0 \in X$  is obtained independently by recursion  $x_n \in \{\omega_1(x_{n-1}), \omega_2(x_{n-1}), \dots, \omega_N(x_{n-1})\}$ ,  $n = 1, 2, \dots$ . If a sufficiently large integer  $N_{max}$  is selected, the sequence  $\{x_n, n \geq N_{max}\}$  converges to the attractor of the IFS.

The compressed mapping  $\omega_i$  usually selects the affine transformation in the following form:

$$\omega_i \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a_i & b_i \\ c_i & d_i \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \end{pmatrix} = \begin{pmatrix} r_i \cos \theta_i & -s_i \sin \varphi_i \\ r_i \sin \theta_i & s_i \cos \varphi_i \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \end{pmatrix} = A_i X + B_i, i = 1, 2, \dots, N. \quad (1)$$

Scaling parameters  $r_i$  and  $s_i$  of the above formula are the compression scaling factors in the  $x$  and  $y$  directions, respectively, which determine the scaling down and scaling up of the fractal attractor along these two directions. Rotation parameters  $\theta_i$  and  $\varphi_i$  determine the rotation angles of the fractal attractor around the  $x$  and  $y$  directions, respectively. Displacement parameters  $e_i$  and  $f_i$  represent the translation along the  $x$  and  $y$  directions, respectively.

**Definition 2:** The random process  $\{x_n, n = 0, 1, 2, \dots\}$  is called the Markov chain, and it only takes a finite or countable number of values  $e_1, e_2, \dots$ . Here,  $\{1, 2, \dots\}$  is used to mark  $e_1, e_2, \dots$ , which are called states of the process, and its subset is denoted as  $S$  and is called a state space of the process. For any  $n > 0$  and states  $i, j, i_1, \dots, i_{n-1}$ , we have the following formula:

$$P\{x_n = j \mid x_1 = i_1, \dots, x_{n-1} = i\} = P\{x_n = j \mid x_{n-1} = i\}. \quad (2)$$

**Definition 3:** The conditional probability  $P\{x_n = j \mid x_{n-1} = i\}$  in Eq. (2) is called the transition probability of the Markov chain. If  $p_{ij} = P\{x_n = j \mid x_{n-1} = i\}$  ( $n > 0$ ), the matrix  $P = (p_{ij})_{N \times N}$  is called the transition probability matrix, which meets the following two conditions: (1)  $p_{ij} \geq 0, i, j \in S$ ; (2)  $\sum_{j \in S} p_{ij} = 1, \forall i \in S$ .

**Definition 4:** If the compressed mapping of the entire space  $(X, d)$  is  $\omega_i: X \rightarrow X, i = 1, 2, \dots, N$ , letting  $P = (p_{ij})_{N \times N}$  be the Markov transition probability matrix, which satisfies  $p_{ij} \geq 0, \sum_{j=1}^N p_{ij} = 1, i = 1, 2, \dots, N$  where  $p_{ij}$  represents the probability of mapping  $\omega_j$  at this time when the mapping  $\omega_i$  is selected the last time, then  $\{X; \omega_1, \omega_2, \dots, \omega_N, P\}$  is a Markov IFS.

#### 3.2 Affine transformation decomposition theorem

**Theorem 1:** Let  $\omega_1, \omega_2, \dots, \omega_N$  be the IFS codes of the fractal image  $f$  and  $\omega_i^{-1}$  be the inverse of  $\omega_i$ . The operator “ $\circ$ ” is defined as  $(\omega_i \circ \omega_j)(x) = \omega_i(\omega_j(x))$ , and then  $\omega_i \circ \omega_j \circ \omega_i^{-1}, j = 1, 2, \dots, N$  is the IFS code of subimage  $\omega_j(f)$ .

The transformation formula of a subimage  $\omega_j(f)$  transformation matrix is as follows:

$$(\omega_i \circ \omega_j \circ \omega_i^{-1})(x) = A_i A_j A_i^{-1} x + (B_i + A_i B_j - A_i A_j A_i^{-1} B_i), j = 1, 2, \dots, N. \quad (3)$$

According to Eq. (3), the generation can be decomposed into several subimages, and the decomposed subimages can be further decomposed into smaller subimages, which can be gradually decomposed to the required level. By applying the decomposition theorem of the affine transformation, the IFS is decomposed layer by layer to generate multiple independent subsystems to obtain the IFS decomposition codes of local subimages.

### 3.3 Construction of probability distribution sequence

**Definition 5:** Supposing that probability set  $P = \{p_1, p_2, \dots, p_N\}$  and subscript set  $I = \{1, 2, \dots, N\}$  satisfy  $\sum_{i=1}^N p_i = 1, p_i > 0, i = 1, 2, \dots, N$ , then each determined set  $p_i (i = 1, 2, \dots, N)$ ,  $\{p_1, p_2, \dots, p_N\}$  is a probability distribution in the probability set.

**Definition 6:** Supposing that  $R$  is the real number field and set subscripts  $\sigma, \eta \in I$ , with  $\zeta(\sigma, \eta) = a \cdot \sigma + b \cdot \eta + c$  (real parameters  $a, b, c \in R, a > 0, b > 0, c \geq 0$ ), then  $\zeta(\sigma, \eta)$  is called the probability variable in  $I$ . The corresponding function  $F(x) > 0$  defined on the real number field is the distribution function of  $\zeta(\sigma, \eta)$ , and  $p_{\sigma, \eta}^*(\zeta(\sigma, \eta) = x) = F(x), x \in R$  is the likelihood probability of probability variable  $\zeta(\sigma, \eta)$ .

When selecting an appropriate distribution function  $F(x)$  to calculate the likelihood probability  $p_{\sigma, \eta}^* (\sigma, \eta \in I)$ , the likelihood probability set is normalized as  $p_{\sigma, \eta}^* / \sum_{\sigma} \sum_{\eta} p_{\sigma, \eta}^*$  ( $\sigma, \eta = 1, 2, \dots, N$ ), and the probability distribution  $\{p_{\sigma, \eta}, 1 \leq \sigma, \eta \leq N\}$  can be obtained.

The factors that determine the probability distribution of an IFS are (1) the distribution function and (2) the probability variables  $\zeta(\sigma, \eta) = a \cdot \sigma + b \cdot \eta + c$ . Different probability distribution sequences can be obtained by adjusting the values of parameters  $a, b, c$ , or selecting different forms of the function  $F(x)$ .

### 3.4 Perturbation of affine transformation parameters

To further improve the controllability of the IFS attractor image shape change, we used a random number sequence as the control factor to perturb the parameters of the affine transformation randomly within a specific range of numerical values to realize the evolution control of the fractal local shape structure of the IFS.

The perturbation function is set as  $D_k(p, q, r, k) = p + (q + r \times rand) \times g(k)$ , where  $\{p, q, r\}$  are the perturbation variables;  $k$  is the iterative cyclic variable of the calculation procedure;  $rand$  is a random factor with a uniform distribution in the interval  $[0, 1]$ ; function  $g(k)$  is the allocation function of  $k$ . Its purpose is to allocate the form of the perturbation function of the system. By using the perturbation function  $D_k(\cdot)$ , the proportion parameters of the affine transformation of the IFS adopt multiplication perturbation, and the rotation parameters and displacement parameters use additive perturbation. That is,

$$A_i = \begin{pmatrix} r_i D_k(\cdot) \cos(\theta_i + D_k(\cdot)) & -s_i D_k(\cdot) \sin(\varphi_i + D_k(\cdot)) \\ r_i D_k(\cdot) \sin(\theta_i + D_k(\cdot)) & s_i D_k(\cdot) \cos(\varphi_i + D_k(\cdot)) \end{pmatrix}, B_i = \begin{pmatrix} e_i + D_k(\cdot) \\ f_i + D_k(\cdot) \end{pmatrix}. \quad (4)$$

### 3.5 Proposed algorithm

The IFS is a classical system of fractals. In the iterative process, the transfer method of an iterative calculation between various parts of an image is associated with a certain probability. The method uses the processing mode of “oneness” for various parts composing the image, so it lacks local controllability. To implement a corresponding control mode at different parts of the image and construct the fractal image with a changeable shape and features, we proposed a method that separates various parts of the image for independent operation. The method can process only a particular local subimage without affecting the other local subimages. First of all, we used the decomposition theorem of the affine transformation to decompose the IFS into several mutually independent subsystems. Then, the images of each subsystem were constructed by applying the generation method of different rules. Finally, we collaged the local subimages generated by each subsystem to form the whole image.

When using the IFS to construct a fractal image, we applied the Markov double random process to control the form evolution of the local fractal image of the IFS, in addition to using the parameters of the affine transformation controlled by the defined perturbation function. Among the two processes, one Markov process participated in the calculation of the affine transformation family, while the other Markov process determined the presentation state of pixels obtained by the IFS iteration calculation. These two random processes are achieved by using two corresponding transition probability matrices.

It is assumed that  $P = (p_{ij})_{N \times N}$ ,  $1 \leq i, j \leq N$  represents the transition probability matrix that controls the selection state of the affine transformation family  $\{\omega_1, \omega_2, \dots, \omega_N\}$  in the IFS and  $Q = (q_{jk})_{N \times M}$ ,  $1 \leq j \leq N$ ,  $1 \leq k \leq M$  denotes the transition probability matrix. The condition  $\{O_m: m = 1, 2, \dots, M\}$  of the corresponding iterative pixel when the transformation is selected as  $\omega_j$ , and  $P, Q$  satisfies

$$\begin{aligned} p_{ij} &\geq 0, \sum_{j=1}^N p_{ij} = 1, \quad i = 1, 2, \dots, N, \\ q_{jk} &\geq 0, \sum_{k=1}^M q_{jk} = 1, \quad j = 1, 2, \dots, N. \end{aligned} \quad (5)$$

The presentation condition of the iteration point of each affine transformation is defined as one of the following five cases.

$O_1$  Represents “growth”: The iterative calculation is performed and coordinate points are drawn. Its probability value is denoted by  $q_{j1}$ .

$O_2$  Represents “death”: The points generated last time do not participate in the iterative calculation this time. Its probability value is denoted by  $q_{j2}$ .

$O_3$  Represents “pause”: It only participates in the iterative calculation and does not draw the coordinate points. Its probability value is denoted by  $q_{j3}$ .

$O_4$  Represents “repeat”: The iteration calculation is carried out and the coordinate points are drawn, then the previous step is returned to. Its probability value is denoted by  $q_{j4}$ .

$O_5$  Represents “branch”: The iterative calculation is carried out without drawing the coordinate points, and other graphs are drawn instead of drawing points. Its probability value is denoted by  $q_{j5}$ .

The local control algorithm is described as follows:

Step 1: We set an IFS  $\{X; \omega_1, \omega_2, \dots, \omega_N\}$  and its associated probability vector  $(p_1, p_2, \dots, p_N)$  satisfying  $p_i > 0, \sum_{i=1}^N p_i = 1$ .

Step 2: We select some affine transformations from the affine transformation family  $\{\omega_1, \omega_2, \dots, \omega_N\}$ , decompose them step by step to the set orders (set as  $n$ ) according to transformation formula Eq. (3), and then obtain the IFS decomposition code  $IFS_i^{(n)} \triangleq \{\omega_{i_1 i_2 \dots i_n j}, j = 1, 2, \dots, N\}$  of subimage  $\omega_i(f)$ .

Step 3: For the first  $n - 1$  levels of the affine transformation sequence obtained by decomposition,  $\omega_{i_1}, \omega_{i_1 i_2}, \dots, \omega_{i_1 i_2 \dots i_{n-1}}$ , these affine transformations only participate in the iteration without drawing points.

Step 4: Using the method in Sect. 3.3, set and select the transition probability matrix of affine transformation family  $\{\omega_{i_1 i_2 \dots i_n j}, j = 1, 2, \dots, N\}$  as  $P(p_{ij})$  and the transition probability matrix of the presentation condition of iterative pixel points as  $Q(q_{jk})$ . These satisfy the following conditions:

$$\begin{aligned}
 p_{ij} &\geq 0, \sum_{j=1}^N p_{ij} = 1, \quad i = 1, 2, \dots, N, \\
 q_{jk} &\geq 0, \sum_{k=1}^M q_{jk} = 1, \quad j = 1, 2, \dots, N.
 \end{aligned}
 \tag{6}$$

Step 5: Using the method in Sect. 3.4, we determine the perturbation function  $D_k(p, \mu, v, k) = p + (\mu + v \times rand) \times g(k)$  and multiply or add the affine transformation family of the system  $IFS_i^{(n)} \triangleq \{\omega_{i_1 i_2 \dots i_n j}, j = 1, 2, \dots, N\}$ :

$$r_i \times D_k(\cdot), s_i \times D_k(\cdot), \theta_i + D_k(\cdot), \phi_i + D_k(\cdot), e_i + D_k(\cdot), f_i + D_k(\cdot).
 \tag{7}$$

Step 6: We select affine transformation  $\omega_{i_1 i_2 \dots i_n j}$  according to the row probability distribution  $\{p_{i1}, q_{i2}, \dots, p_{iN}\}$  of the matrix  $P(p_{ij})$  and determine the presentation condition  $\{O_1, O_2, O_3, O_4, O_5\}$  of affine transformation points according to the row probability distribution  $\{q_{j1}, q_{j2}, \dots, q_{jM}\}, M = 5$  of the matrix  $Q(q_{jk})$ .

Step 7: The local subimage of each part of the iteration output system is produced.

## 4. Experimental Results and Discussion

### 4.1 Basic experimental setting

We chose the Poisson distribution or the Gaussian distribution for the probability distribution function  $F(x)$  in the experiment.

Poisson probability distribution formula:

$$\pi(\lambda) = \frac{\lambda^x}{x!} e^{-\lambda} \quad (\lambda \in R, \lambda > 0) \quad (8)$$

Gaussian probability density function:

$$N(\mu, \sigma) = \frac{1}{\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (\mu, \sigma \in R, \sigma > 0) \quad (9)$$

The parameter space formed by the two distributions combined with the value of the probability variable  $\xi(\sigma, \eta) = a \cdot \sigma + b \cdot \eta + c$  is as follows:

$$\{(a, b, c, \lambda) \mid a, b, c, \lambda \in R, \lambda > 0\}, \quad \{(a, b, c, \mu, \sigma) \mid a, b, c, \mu, \sigma \in R, \sigma > 0\}. \quad (10)$$

In the perturbation function  $D_k(p, q, r, k) = p + (q + r \times rand) \times g(k)$ , variables  $\{(p, q, r, k) \mid p, q, r \in R, k \in N, k > 0\}$  and random factors  $rand \in [0, 1]$  are random numbers with a uniform distribution.

The transformation of IFS affine transformation parameters  $(a, b, c, d)$  and  $(r, s, \theta, \varphi)$  is as follows:

$$\begin{cases} r = a / \cos[a \tan(c / a)], \\ s = d / \cos[a \tan(-b / d)], \\ \theta = a \tan(c / a), \\ \varphi = a \tan(-b / d). \end{cases} \quad (11)$$

The fern and tree fractals are taken as test examples of the algorithm in the experiment. The initial IFS codes of these three fractal images are listed in Table 1, and schematic diagrams of the generators of these three fractal images are shown in Fig. 1.

### 4.2 Algorithm implementation results

In the following algorithm implementation process, the Gaussian distribution function is adopted to define the transition probability matrix  $P(p_{ij})$  with its parameter values  $(a, b, c, \mu, \sigma) = (2, 1, 8, 6, 2)$ , while the Poisson distribution function is adopted to define the

Table 1  
IFS initial codes of fern fractal and tree fractals.

(a) Fern					
<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0	0	0	0.16	0	0
0.85	0.04	-0.04	0.85	0	1.6
0.2	-0.26	0.23	0.22	0	1.6
-0.15	0.28	0.26	0.24	0	0.44
(b) Tree 1					
<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
-0.058	-0.07	0.453	-0.111	0.598	0.097
-0.035	0.07	-0.469	-0.022	0.488	0.509
0.195	-0.488	0.344	0.443	0.443	0.245
-0.637	0	0	0.501	0.856	0.251
(c) Tree 2					
<i>a</i>	<i>b</i>	<i>c</i>	<i>d</i>	<i>e</i>	<i>f</i>
0.05	0.6	0	0	0	0
0.05	-0.5	0	0	0	1.0
0.55	0.4	-0.698	-0.698	0	0.7
0.5	0.55	-0.524	-0.524	0	1.0
0.5	0.45	0.349	0.349	0	1.1
0.6	0.5	0.698	0.698	0	0.6

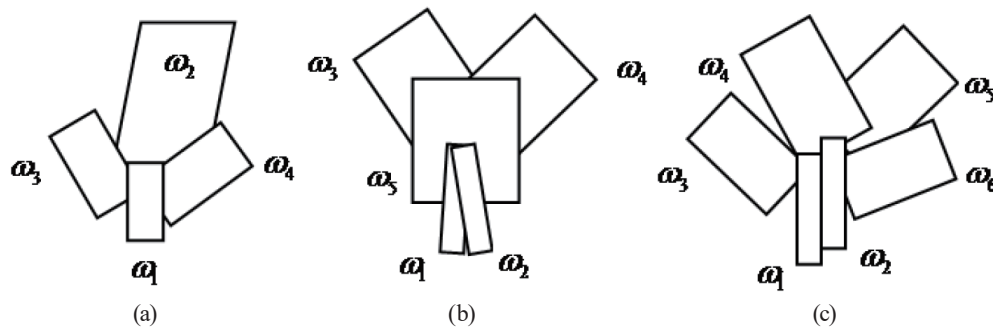


Fig. 1. Generators of three fractal images. (a) fern, (b) tree 1, and (c) tree 2.

transition probability matrix  $Q(q_{jk})$  with its parameter values  $(a, b, c, \lambda) = (2, 3, 6, 7)$ . However, the perturbation of each local affine transformation parameter adopts different combinations.

Figure 2 shows the evolution of the fern fractal. Figure 2(a) is the original image, whose generator is shown in Fig. 1(a). Figure 2(b) is the result obtained by the local subimage  $\omega_2(f)$  by using multiplication perturbation of the scale parameter and additive perturbation of the rotation angle. With increasing compression processing carried out by the ratio parameter, the local feature changes to a thin line. Figure 2(c) is the image formed by carrying out the perturbation of the subimages  $\omega_2(f)$ ,  $\omega_3(f)$ , and  $\omega_4(f)$ . In the perturbation process, if the rotation angle remains unchanged and multiplication perturbation is carried out on the scale parameters, then the random factor *rand* is used in the perturbation function. Figures 2(d)–2(f) are the images generated by carrying out different perturbations of the subimages  $\omega_2(f)$ ,  $\omega_3(f)$ , and  $\omega_4(f)$ ,



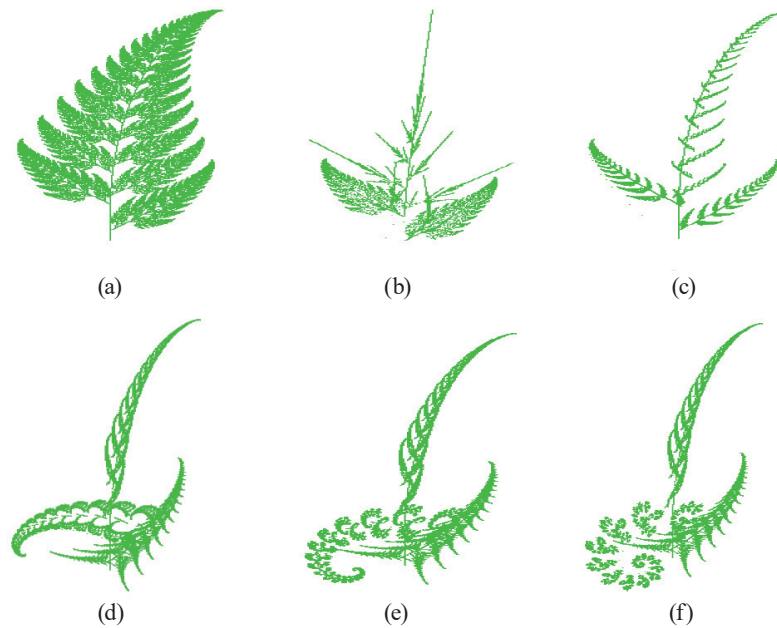


Fig. 2. (Color online) Local control of fern fractal image.

respectively. The left and right branches of subimage  $\omega_2(f)$  are subjected to multiplication perturbation of the scale parameters and additive perturbation of the rotation angle. Meanwhile, the left, right, and upper branches of subimages  $\omega_3(f)$  and  $\omega_4(f)$  are subjected to multiplication perturbation of the scale parameters and additive perturbation of the rotation angle. With increasing angle of rotation used for subimage  $\omega_2(f)$ , it can be seen that the local image becomes deformed.

Figure 3 shows the evolution of fractal image tree 1. Figure 3(a) is the original image, whose generator is shown in Fig. 1(b). Figures 3(b) and 3(c) show the results obtained by local subimage  $\omega_3(f)$  by using multiplication perturbation of the scale parameter and additive perturbation of the rotation angle. Meanwhile, the random factor *rand* is used in the multiplication perturbation, while only additive perturbation of the rotation angle is carried out for local subimage  $\omega_4(f)$ . Figures 3(d) and 3(e) are the results of applying multiplication perturbation of the scale parameter and additive perturbation of the rotation angle to local subimage  $\omega_5(f)$ , and the remaining parts are unchanged. Figure 3(f) is obtained when the parameters of the affine transformation of the local subimage are kept unchanged. In addition, the transition probability matrix  $Q(q_{jk})$  is used to control the presentation condition of affine transformation point  $\omega_i(x)$ , namely, among five conditions: “growth,” “death,” “pause,” “repeat”, and “branch.” The “branch” effect is to replace the drawing dots with red circles to represent the fruits on the tree and simulate a tree laden with fruits.

Figure 4 shows the evolution of fractal image tree 2. Figure 4(a) is the original image, and its generator is shown in Fig. 1(c). Figure 4(b) is the result obtained by local subimages  $\omega_4(f)$  and  $\omega_5(f)$  by multiplication perturbation of the scale parameter and additive perturbation of

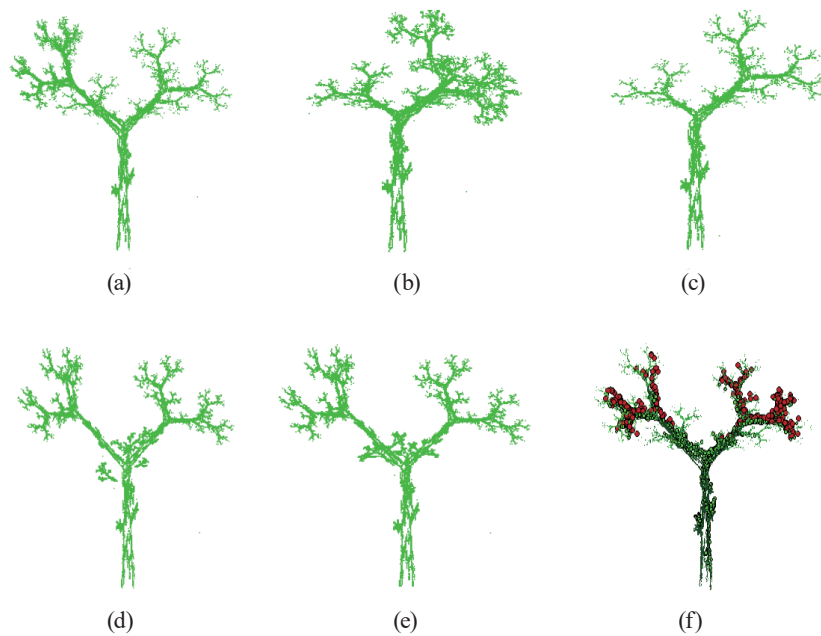


Fig. 3. (Color online) Local control of tree 1 fractal image.

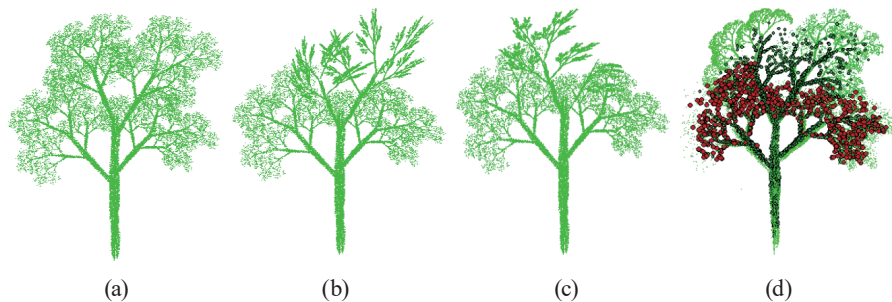


Fig. 4. (Color online) Local control of tree 2 fractal image.

the rotation angle. Changing the scale parameter only causes compression in the horizontal direction. Figure 4(c) is the result obtained by the multiplication perturbation of the scale parameter and the additive perturbation of the rotation angle on local subimages  $\omega_4(f)$  and  $\omega_5(f)$ . Meanwhile, the perturbation of the scale parameter for subimage  $\omega_4(f)$  only causes compression in the horizontal direction, while the scale parameter of subimage  $\omega_5(f)$  causes compression in both the horizontal and vertical directions simultaneously. Therefore, it can be seen that the compression degree of the subimage is more significant than  $\omega_4(f)$  from the experimental results. For Fig. 4(d), local subimages  $\omega_4(f)$  and  $\omega_5(f)$  use the transition probability matrix  $P(p_{ij})$  to control the deformation, while local subimages  $\omega_3(f)$  and  $\omega_6(f)$  adopt the “branch” treatment, as shown in Fig. 3(d), which also simulates a tree laden with fruits.

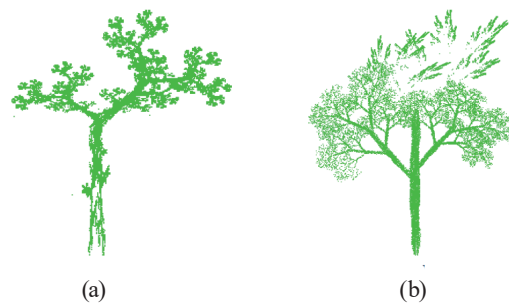


Fig. 5. (Color online) Effect of the distribution function.

If the parameters of the Gaussian and Poisson distribution functions are adjusted, Fig. 3(b) becomes Fig. 5(a) with  $(a, b, c, \mu, \sigma) = (2, 1, 0, 8, 2)$  as the parameter of the Gaussian distribution function and  $(a, b, c, \lambda) = (2, 3, 1, 6)$  as the parameter of the Poisson distribution function. When the parameters of the Gaussian and Poisson distribution functions are  $(a, b, c, \mu, \sigma) = (3, 1, 0, 2, 3)$  and  $(a, b, c, \lambda) = (5, 2, 1, 2)$ , respectively, Fig. 4(b) becomes Fig. 5(b).

Compared with the standard fractal iteration systems referred to previously, the new method in this research can modify the local shapes of fractal images more easily. It can not only maintain the basic characteristics of the same image but also develop them into new shapes. In addition, this method can give different colors to local images, enabling local color rendering.

## 5. Conclusions

Although the IFS has become an essential method in the computer generation of fractal images, the difficulty in the coding of the IFS affine transformation still remains. An effective way to overcome this difficulty is to add other processes to the existing IFS to control and produce fractal images with more desirable forms. We introduced the Markov random process into the fractal structure of the IFS to decompose the affine transformation of the IFS and to obtain the IFS decomposition code of a local subimage. By defining the transition probability matrix and the perturbation function for the evolution control of the local subimage's shape modeling, various images with irregular local forms are generated from the basic geometric structures of the original image. The proposed system further increases the ability to construct IFS fractal images. The new ideas in this research are as follows:

- (1) The Markov double random process is used to control the morphology of a fractal image.
- (2) Each local deformation operates independently.
- (3) Localized interference does not affect the overall shape.

Even with the improved results shown in this research, the new method lacks the effective regulation of fractal image color changes as the random number sequence cycle is not long enough. We discussed the fractal structure of a single IFS, but the ability of a single IFS to construct complex images is still limited. Therefore, it must be discussed in future research whether a multimodal IFS can be merged to construct a multibranch system and generate a fractal image with a more complex structure.

## Acknowledgments

The authors would like to thank the following for support in this research: Ningde Normal University Innovation Team (grant number 2017T02), Ningde Normal University Major Project Development Scheme (grant number 2017ZDK05), Recruiting High-Level Talent Program of Ningde Normal University (2017Y005), Project of Fujian Provincial Natural Science Fund (2017I0016), and Fujian Key Laboratory of Intelligent Ecotourism and Leisure Agriculture.

## References

- 1 M. Cho and Y. Yoo: *J. Korean Soc. Costume* **68** (2018) 76. <https://doi.org/10.7233/jksc.2018.68.4.076>
- 2 T. Yang, F. Tian, J. A. Covington, F. Xu, Y. Xu, A. Jiang, J. Qian, R. Liu, Z. Wang, and Y. Huang: *Chemosens.* **31** (2019) 11. <https://doi.org/10.3390/chemosensors7030031>
- 3 G. Danik, N. N. Gorobets, A. A. Tolstaya, and V. A. Timonyk: *Proc. 14th Int. Crimean Conf. Microwave and Telecommunication Technology (IEEE 2004)* 570–571. <https://doi.org/10.1109/CRMICO.2004.183356>
- 4 S. J. Guastello: *Nonlinear Dyn. Psychol. Life Sci.* **19** (2015) 81. <http://europepmc.org/article/med/25575560>
- 5 I. Dierking: *Liq. Cryst. Today* **21** (2012) 54. <https://doi.org/10.1080/1358314x.2012.720515>
- 6 A. Samper and B. Herrera: *Nexus Network J.* **16** (2014) 251. <https://doi.org/10.1007/s00004-014-0187-7>
- 7 T. Xingyan, D. Jiyuan, and D. Hongyan: *Proc. 3rd Int. Conf. E-Business and E-Government (IEEE Computer Society, 2012)* 510–513. <https://doi.org/10.1109/ICEE.2012.1143>
- 8 K. Young and Y. Yang: *J. Korean Soc. Design Culture* **22** (2016) 453. <https://doi.org/10.17548/ksaf.2016.12.26.201>
- 9 S. Salcedo-Sanz, A. Aybar-Ruíz, and C. Camacho-Gomez: *Commun. Nonlinear Sci. Numer. Simul.* **56** (2017) 434. <https://doi.org/10.1016/j.cnsns.2017.08.010>
- 10 D. I. I. Attia: *Int. J. Des. Objects* **13** (2019) 19. <https://doi.org/10.18848/2325-1379/CGP/v13i01/19-39>
- 11 A. D. Potgantwar and S. G. Bhirud: *Int. J. Comput. Appl.* **1** (2011) 30. <https://doi.org/10.5120/57-159>
- 12 P. Jaros, L. Maslanka, and F. Strobin: *Numer. Algorithms* **73** (2016) 477. <https://doi.org/10.1007/s11075-016-0104-0>
- 13 G. Garcla: *Mediterr. J. Math.* **14** (2017) 14. <https://doi.org/10.1007/s00009-017-0845-6>
- 14 F. Georgescu, R. Miculescu, and A. Mihail: *J. Fixed Point Theory Appl.* **20** (2018) 125. <https://doi.org/10.1007/s11784-018-0497-6>
- 15 P. Sherman and J. C. Hart: *Comput. Graph.* **27** (2007) 143. [https://doi.org/10.1016/S0097-8493\(02\)00252-2](https://doi.org/10.1016/S0097-8493(02)00252-2)
- 16 M. Radu: *Acta Appl. Math.* **130** (2014) 135. <https://doi.org/10.1007/s10440-013-9841-4>
- 17 A. M. M. Sharif Ullah, Y. Sato, A. Kubo, and J. Tamaki: *Comput.-Aided Des. Appl.* **12** (2015) 241. <https://doi.org/10.1080/16864360.2014.981452>
- 18 C. Stepien: *Comput. Graph.* **33** (2009) 576. <https://doi.org/10.1016/j.cag.2009.02.003>
- 19 L. L. Zhang: *Comput. Appl. Software* **27** (2010) 115. [http://en.cnki.com.cn/Article\\_en/CJFDTOTAL-JYRJ201012037.htm](http://en.cnki.com.cn/Article_en/CJFDTOTAL-JYRJ201012037.htm)
- 20 L. L. Zhang and Z. G. Pan: *Trans. Edutainment, LNCS* **7220** (2012) 178. [https://doi.org/10.1007/978-3-642-31439-1\\_16](https://doi.org/10.1007/978-3-642-31439-1_16)
- 21 G. Flavian and M. Radu: *Indag. Math.* **30** (2019) 214. <https://doi.org/10.1016/j.indag.2018.10.001>
- 22 Y. X. Zhuang, Y. S. Xiong, and K. Tan: *Comput. Eng. Sci.* **35** (2013) 142. [http://en.cnki.com.cn/Article\\_en/CJFDTOTAL-JSJK201306027.htm](http://en.cnki.com.cn/Article_en/CJFDTOTAL-JSJK201306027.htm)

## About the Authors



**Li-liang Zhang** received his B.A. degree in mathematics in 1983 from Fujian Education University, China. He is currently a professor of computer and information engineering at Ningde Normal University, Fujian, China. His research interests lie in digital image processing and big data. (13859603598@139.com)



**Ting-cheng Chang** received his M.S. and Ph.D. degrees in process control and mechanical engineering in 1992 and 1996 from University of Houston and University of Texas at Arlington, Texas, USA, respectively. He is currently a professor of computer and information engineering at Ningde Normal University, Fujian, China. His research interests lie in the Internet of Things, data mining, big data, and optimal theory. (18250922163@163.com)



**Yan-ming Mao** received his M.S. degree in computer software and theory in 2009 from Yunnan Normal University, China. He is now a lecturer of information and mechatronic engineering at Ningde Normal University, Fujian, China. His research interests lie in computer vision, artificial intelligence, and big data. (hydrone@163.com)