PERGAMON

Computers & Graphics 27 (2003) 407–414

COMPUTERS & GRAPHICS

www.elsevier.com/locate/cag

Technical section

On bounding boxes of iterated function system attractors

Hsueh-Ting Chu, Chaur-Chin Chen*

Department of Computer Science, National Tsing Hua University, Hsinchu 300, Taiwan, ROC

Abstract

Before rendering 2D or 3D fractals with iterated function systems, it is necessary to calculate the bounding extent of fractals. We develop a new algorithm to compute the bounding box which closely contains the entire attractor of an iterated function system.

Keywords: Fractals; Iterated function system; IFS; Bounding box

1. Introduction

Barnsley [1] uses iterated function systems (IFS) to provide a framework for the generation of fractals. Fractals are seen as the attractors of iterated function systems. Based on the framework, there are many algorithms to generate fractal pictures [1–4]. However, in order to generate fractals, all of these algorithms have to estimate the bounding boxes of fractals in advance. For instance, in the program Fractint (http://spanky.triumf.ca/www/fractint/fractint.html), we have to guess the parameters of "image corners" before the beginning of drawing, which may not be practical.

For this reason, this paper is devoted to develop a practical bounding box algorithm. A good bounding algorithm is also a priori for rendering 3D fractals. Hart and DeFanti [5] use bounding spheres in their system of 3D fractal rendering. Sometimes the bounding spheres are very loose such that the rendering of common fractals may be inefficient. Most of 3D computer graphic systems employ bounding boxes for rendering objects. Thus, it is important to develop a tight bounding box algorithm for rendering fractal objects in a 3D system.

1.1. Iterated function systems

Definition 1. A transform $f: X \to X$ on a metric space (X, d) is called a contractive mapping if there is a constant $0 \leqslant s < 1$ such that

$$d(f(x), f(y)) \leqslant s \cdot d(x, y) \quad \forall x, y \in X, \tag{1}$$

where s is called a contractivity factor for f.

Definition 2. In a complete metric space (X, d), an iterated function system (IFS) [1] consists of a finite set of contractive mappings w_i, for $i = 1, 2, \ldots, n$, which is denoted as $W = \{X; w_1, w_2, \ldots, w_n\}$.

In this paper, we assume contractive affine transforms in an iterated function system. A contractive affine transform w in R^2 is

$$w\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} a & b \\ c & d \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} = \mathbf{A}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix} \tag{2}$$

and a contractive affine transform w in R^3 is

$$w\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) = \begin{bmatrix} a & b & c \\ d & e & f \\ i & j & k \end{bmatrix}\begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} t \\ u \\ v \end{bmatrix} = \mathbf{A}\begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} t \\ u \\ v \end{bmatrix}, \tag{3}$$

*Corresponding author. Tel.: +11-886-3-573-1078; fax: +11-886-3-572-3694.

E-mail address: cchen@cs.nthu.edu.tw (C.-C. Chen).

0097-8493/03/$ - see front matter © 2003 Elsevier Science Ltd. All rights reserved.
doi:10.1016/S0097-8493(03)00035-9

where the eigenvalues of the matrix A have magnitude less than 1.

Theorem 1 (Barnsley [1]). *Let $f: : X \to X$ be a contractive mapping in a complete metric space (X, d). Then f possesses a unique fixed-point $x_f \in X$, and for any $x \in X$, the Cauchy sequence $\{f^n(x)\}$ converges to x_f, where $f^n(x) = f^{n-1}(f(x))$, and $f^0(x) = x$.*

In a Hausdorff metric space $(H(X), h(d))$, an IFS, W, is also a contractive mapping, thus it possesses a unique fixed-point, called the attractor of the IFS. Tables 1–3 list the 2D IFS codes of fractals given by the program Fractint, which generate the "dragon", "coral", and "fern" fractal objects [1], respectively. Fig. 1 shows a picture of these fractal objects. We will compute the bounding boxes of these IFS codes later.

Lemma 1. *Let $W = \{X;\ w_1, w_2, \ldots, w_n\}$ be an iterated function system. Let A be the attractor associated with W on $(H(X), h(d))$, and let $z_1, z_2, \ldots, z_n$ be the fixed points of $w_1, w_2, \ldots, w_n$, respectively, on (X, d). Then $z_i \in A$ for $i = 1, 2, \ldots, n$.*

Proof. $z_i = w_i^\infty(z_i)$ since z_i is the fixed point of w_i, hence $z_i \in W^\infty(\{z_i\})$. By Theorem 1, $W^\infty(\{z_i\}) = A$. Hence $z_i \in W^\infty(\{z_i\}) = A$. □

Theorem 2. *Let $W = \{X;\ w_1, w_2, \ldots, w_n\}$ be an iterated function system. Let A be the attractor associated with W, and $z \in A$. Then the attractor A is equal to $\{z\} \cup \{W(z)\} \cup \{W^2(z)\} \cup \cdots \cup \{W^\infty(z)\}$.*

Proof. (a) By Theorem 1, $\{W^\infty(z)\} = A$. Hence,

$$A \subseteq \{z\} \cup \{W(z)\} \cup \{W^2(z)\} \cup \cdots \cup \{W^\infty(z)\}. \quad (4)$$

(b) We know $z \in A$, then all the sets $\{z\}, \{W(z)\}, \{W^2(z)\}, \ldots, \{W^\infty(z)\} \subseteq A$. Thus,

$$\{z\} \cup \{W(z)\} \cup \{W^2(z)\} \cup \cdots \cup \{W^\infty(z)\} \subseteq A. \quad (5)$$

From (a) and (b), $A = \{z\} \cup \{W(z)\} \cup \{W^2(z)\} \cup \cdots \cup \{W^\infty(z)\}$. □

Theorem 2 indicates that the entire attractor can be generated by applying the IFS on some point $z \in A$ iteratively. Thus, it provides a formula to represent all the points in the attractor as $A = \{p | p = t_1 t_2 \ldots t_c(z), t_i \in \{w_1, w_2, \ldots, w_n\}$ and z is the fixed-point of some mapping $w_j\}$.

1.2. Previous work

Let $W = \{X; w_1, w_2, \ldots, w_n\}$ be an iterated function system on X, and let A be the attractor associated with W. We try to find the bounding volume of the attractor A. In the literature, only bounding ball algorithms were developed [5–8] for IFSs. A bounding ball is also called a bounding circle in a 2D space or a bounding sphere in a 3D space.

Canright [6] uses multiple bounding balls to envelop the entire attractor of an IFS. Let $z_1, z_2, \ldots, z_n$ be the fixed points of $w_1, w_2, \ldots, w_n$ on (X, d), and let $s_1, s_2, \ldots, s_n$ be the contractivity factors of $w_1, w_2, \ldots, w_n$, respectively. They seek n balls C_i centered on z_i, $1 \leqslant i \leqslant n$ such that $w(A) \subset C_i$, then $A \subset \bigcup_{i=1}^n C_i$. The radii $\{r_i\}$ are chosen to satisfy [6]

$$r_i = s_i \max_{j \neq i}(r_j + \mathrm{d}(z_i, z_j)), \quad 1 \leqslant i \leqslant n. \quad (6)$$

Fig. 2 illustrates Canright's bounding balls for the fern. In order to enclose Canright's balls, a large bounding box $B \equiv Box(x_{min}, x_{max};\ y_{min}, y_{max}) = (-14.24, 19.16; -9.36, 26.70)$ is necessary. We mark the minimum bounding box of the fern with dotted lines in the figure, then the computed bounding box by Canright's algorithm is too loose in practice. However, Canright provided a very simple and fast solution to compute a bound for drawing fractals.

Hart and DeFanti [5] introduced another bounding ball algorithm. Only one bounding ball is computed to enclose the attractor of an IFS in their algorithm. Initially the ball C is the unit ball at the origin, then the ball is moved and enlarged to envelop the entire attractor iteratively. The next ball C^* in the sequence is found as

$$o^* = \frac{1}{n} \sum_{i=1}^{n} w_i(o) \quad (7)$$

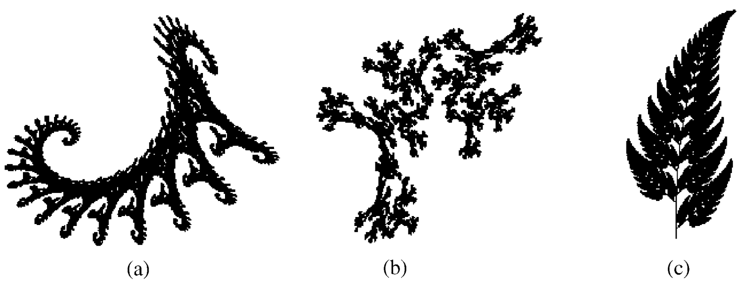

Fig. 1. The attractors of the 2D IFS codes in Tables 1–3.

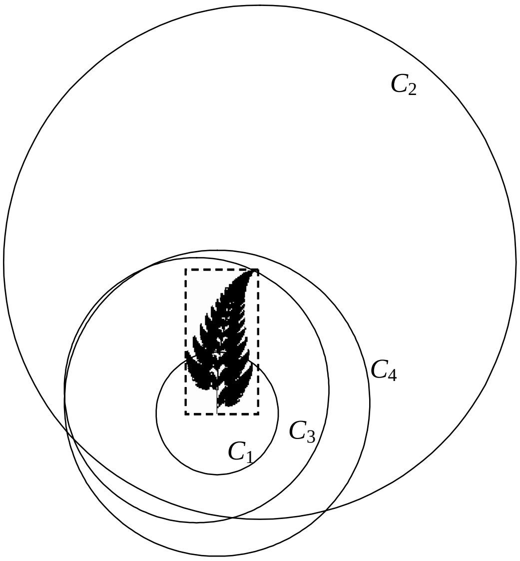

Fig. 2. Canright's bounding balls for fern.

and

$$r^* = \max_{i=1\ldots n} (s_i r + d(w_i(o), o^*)), \tag{8}$$

where o and r are the origin and radius of ball C and likewise for ball C^*.

We show the bounding ball for the fern computed by Hart's algorithm in Fig. 3. The origin and radius of the ball C^* are (0.03, 1.44) and 9.28, respectively. Compared with Canright's algorithm, Hart's algorithm obtains a better result in this case. Rice [8] tries to improve Hart's algorithm by optimizing the radius r. The effort made by Rice has little improvement in this case because the computed radius for the fern, by Hart's algorithm, is close to the optimal radius.

Though many researchers have developed bounding ball algorithms as shown above, the minimum bounding boxes of the fern in Figs. 2 and 3 are much smaller than the computed bounds. Thus, we try to develop a bounding box algorithm instead. We desire to acquire a very tight bounding volume instead of a loose one. However, there are many of problems we have to overcome. In the next section, we consider these problems and their solutions. For convenience, we consider the bounding problem in 2D cases. In Section 3, the new algorithm is presented. We show experimental results in Section 4. Finally, conclusions are drawn in Section 5.

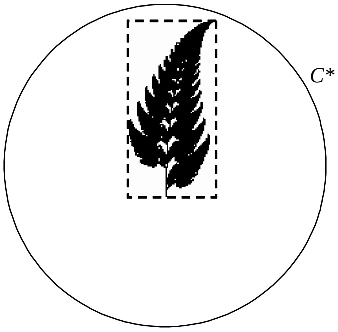

Fig. 3. Hart's bounding ball for fern.

2. Bounding boxes for IFS attractors

The advantage of a bounding ball algorithm is that it can handle rotations. It is a complex problem we have to resolve when developing a bounding box algorithm instead. An affine transformation is composed of translation, scaling, and rotation. Thus a 2D contractive affine transform of Eq. (2) can be rewritten as

$$w\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} s_x \cos\theta & s_y \sin\theta \\ -s_x \sin\theta & s_y \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}, \tag{9}$$

where $|s_x| < 1$ and $|s_y| < 1$.

For example, let $\theta = \pi/4$, $s_x = s_y = 0.9$, $o_x = o_y = 0$, then

$$w\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} 0.64 & 0.64 \\ -0.64 & 0.64 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e \\ f \end{bmatrix}. \tag{10}$$

In this case, we will always obtain a smaller ball after applying w. Here the new radius is 0.9 of the original, but we may need a larger box to enclose the transformed box. Fig. 4 shows the transformation of a box with w. A square with the length of sides 1.28 is necessary to enclose the transformed unit square. As a result, given an arbitrary initial box B_0, $W(B_0)$ may exceed the extent of B_0, for an IFS W. On the contrary, it would be more tractable if we could find an initial box B_0 such that $W(B_0) \subset B_0$. Assuming this hypothesis is true, B_0 will be a loose bounding box of the attractor of W. We can find a tight bounding box of the attractor of W by refining B_0 iteratively according to the following theorem.

2.1. Basic theory

Theorem 3. *Let* $W = \{R^2; w_1, w_2, \ldots, w_n\}$ *be an iterated function system of affine mappings in* R^2. *Let* $B_0 = Box(x_{min}, x_{max};\ y_{min}, y_{max}) \equiv \{(x, y) \in R^2 \mid x_{min} \leqslant x \leqslant x_{max}$ *and* $y_{min} \leqslant y \leqslant y_{max}\}$ *be an arbitrary box satisfying the following condition*

$$W(B_0) \subset B_0. \tag{11}$$

Then

$$W^i(B_0) \subset W^{i-1}(B_0), \quad \text{for } i \geqslant 1. \tag{12}$$

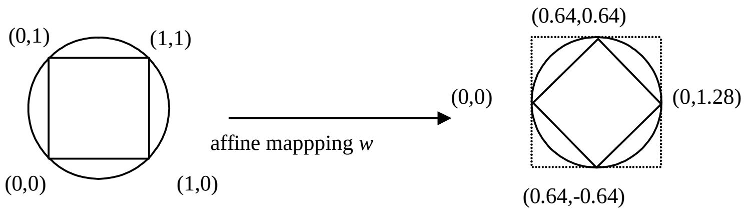

Fig. 4. An example of transforming a box with an affine mapping w.

Proof. We prove this by induction. When $k = 1$, $W^k(B_0) \subset W^{k-1}(B_0)$ by Eq. (10). Assume $W^k(B_0) \subset W^{k-1}(B_0)$ holds for some $k > 1$. We attempt to prove $W^{k+1}(B_0) \subset W^k(B_0)$. Because $W^k(B_0) \subset W^{k-1}(B_0)$, thus for any point $p \in W^k(B_0)$, it implies $p \in W^{k-1}(B_0)$. Therefore, $W(p) \in W^k(B_0)$. Hence $W^{k+1}(B_0) \subset W^k(B_0)$. In consequence, the theorem is proved. □

Definition 3 (Minimum bounding box). Let $S \subset R^2$ be a subset of R^2, the minimum bounding box $M(S)$ of S is a box $Box(x_{min}, x_{max};\ y_{min}, y_{max}) \equiv \{(x, y) \in R^2 \mid x_{min} \leqslant x \leqslant x_{max}$ *and* $y_{min} \leqslant y \leqslant y_{max}\}$ that contains the supreme values of coordinates of points in S. That is, for any point $(x_p, y_p) \in S$, $x_{min} \leqslant x_p \leqslant x_{max}$ and $y_{min} \leqslant y_p \leqslant y_{max}$, and there exist four supreme points p_1, p_2, p_3 and $p_4 \in S$ such that $x_{p_1} = x_{min}$, $x_{p_2} = x_{max}$, $y_{p_3} = y_{min}$ and $y_{p_4} = y_{max}$.

Let the minimum bounding box of $W^{i-1}(B_0)$ be $M(W^{i-1}(B_0)) = Box(x_{min}, x_{max};\ y_{min}, y_{max})$, then $W^{i-1}(B_0) \subset M(W^{i-1}(B_0))$. On the other hand, $W^i(B_0) \subset W^{i-1}(B_0)$ implies $W^i(B_0) \subset M(W^{i-1}(B_0))$ by Theorem 3. Therefore the supreme values of $M(W^{i-1}(B_0))$ are bounded by the rectangle of $(x_{min}, x_{max};\ y_{min}, y_{max})$. This means that $M(W^i(B_0)) \subset M(W^{i-1}(B_0))$ which implies that we will get smaller boxes as we continuously apply the IFS operator W on an initial box B_0, if the condition $W(B_0) \subset B_0$ holds. Let A be the attractor of the IFS W, we know $W^\infty(B_0) = A$ by Theorem 1. Therefore, the sequence $\{M(W^i(B_0))\}_{i=0}^{\infty}$ converges to the minimum bounding box $M(A)$ of the attractor A.

2.2. Existence of an initial box

Assume a box B_0 is found, and the condition $W(B_0) \subset B_0$ holds. Then the box B_0 is a loose bounding box of the attractor of the IFS W. This allows us to generate tighter bounding boxes when we iteratively apply W on B_0. This hypothesis however is based on the existence of the initial box B_0. Does a box B_0 such that the condition $W(B_0) \subset B_0$ holds for any IFS W exist? The answer is negative. For instance, we cannot find such a box for the IFS code of the fractal "dragon" in Table 1. However, we can ensure the hypothesis holds for some IFSs under a certain restriction described below.

Theorem 4. *Let* $W = \{R^2; w_1, w_2, \ldots, w_n\}$ *be an iterated function system of affine mappings in* R^2. *The affine mappings are*

$$w_i\left(\begin{bmatrix} x \\ y \end{bmatrix}\right) = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} e_i \\ f_i \end{bmatrix}.$$

If we assume that

$$|a_i| + |b_i| < 1, \quad |c_i| + |d_i| < 1 \quad \textit{for } i = 1, \ldots, n, \tag{13}$$

then there exists a box B_0 *such that* $W(B_0) \subset B_0$.

Proof. We prove this by finding a box that obeys the condition $W(B_0) \subset B_0$. We check the absolute values of the coefficients of the mappings $\{w_i\}$. Let

$$S = \max_{1 \leqslant i \leqslant n} \{\max\{|a_i| + |b_i|, |c_i| + |d_i|\}\}, \tag{14a}$$

$$E = \max_{1 \leqslant i \leqslant n} |e_i|, \tag{14b}$$

$$F = \max_{1 \leqslant i \leqslant n} |f_i| \tag{14c}$$

and

$$K = \frac{E + F}{1 - S}. \tag{15}$$

Then, the box $B \equiv Box(x_{min}, x_{max};\ y_{min}, y_{max}) = (-K, K; -K, K)$ will be such a box that the condition $W(B_0) \subset B_0$

Table 1
IFS code for dragon

w_i	a_i	b_i	c_i	d_i	e_i	f_i
1	0.824074	0.281482	−0.212346	0.864198	−1.882290	−0.110607
2	0.088272	0.520988	−0.463889	−0.377778	0.785360	8.095795

holds. The verification of this success is given in the appendix. □

Eq. (13) holds for the affine maps of the IFS codes for the fractals "coral" and "fern" in Tables 2 and 3. Hence, we can find an initial box B_0 with Eq. (15), then refine the box into a very tight bounding box as $M(W^m(B_0))$, where m is a large number. However, Eq. (13) does not hold for the fractal "dragon". We overcome this problem by Theorem 5 listed below.

Definition 4. Let W_1, W_2 be two iterated function systems of affine mappings in R^2, and let A_1, A_2 be the attractors associated with W_1, W_2, respectively. We say that W_1, W_2 are equivalent if A_1, A_2 are identical.

Theorem 5. *Let* $W = \{R^2; w_1, w_2, \ldots, w_n\}$ *be an iterated function system of affine mappings in* R^2. *Let* $f_i = w_1 w_i$, *for* $i = 1, 2, \ldots, n$. *Then the IFS* $W' = \{R^2; f_1, f_2, \ldots, f_n, w_2, \ldots, w_n\}$ *is equivalent to* W.

Proof. Let A and A' be the attractors of IFSs W, and W', respectively. Let z be the fixed point of w_1. Then $z \in A$ by Lemma 1. Also by Theorem 1, there exists an integer k, such that $z = w_1^{2k}(z) = f_1^k(z)$. Thus $z \in A'$, too. Thus by Theorem 2, the attractors are $A = \{z\} \cup \{W(z)\} \cup \{W^2(z)\} \cup \cdots \cup \{W^\infty(z)\}$ and $A' = \{z\} \cup \{W'(z)\} \cup \{W'^2(z)\} \cup \cdots \cup \{W'^\infty(z)\}$.

(a) For any point $p \in A$, there exists an integer c such that $p \in W^c(z)$. Let $p = t_1 t_2 \ldots t_c(z)$ where $t_i \in \{w_1, w_2, \ldots, w_n\}$. We scan the sequence $t_1 t_2 \ldots t_c$, and transfer the pattern $t_j t_{j+1}$ into f_s if $t_j = w_1$ and $t_{j+1} = w_s$. If $t_c = w_1$ and $t_{c-1} \neq w_1$, then we can transfer t_c into f_1, as $w_1(z) = w_1^2(z) = z$. Consequently, we can rewrite $t_1 t_2 \ldots t_c(z)$ as $u_1 u_2 \ldots u_d(z)$ where $u_i \in \{f_1, f_2, \ldots, f_n, w_2, \ldots, w_n\}$. Hence $p \in W'^d(z)$ and $p \in A'$.

(b) For any point $p \in A'$, there exists an integer d such that $p \in W'^d(z)$. Let $p = u_1 u_2 \ldots u_d(z)$ where $u_i \in \{f_1, f_2, \ldots, f_n, w_2, \ldots, w_n\}$. It is apparent by definition that there exists an integer c such that $p = t_1 t_2 \ldots t_c(z)$, where $t_i \in \{w_1, w_2, \ldots, w_n\}$. Thus $p \in A$.

By (a) and (b), the attractors A and A' are identical. □

Given an IFS $W = \{R^2; w_1, w_2, \ldots, w_n\}$, we can decompose the affine mapping w_i into $(w_i w_1, w_i w_2, \ldots, w_i w_n)$, if Eq. (13) does not hold for w_i. Thus forming a new IFS, W', that is equivalent to the original IFS. Moreover, we can iteratively apply this decomposition procedure until Eq. (13) holds for all of affine mappings. Because all the affine mappings in W are contractive, the decomposition procedure always succeeds. For example, we can decompose the affine mapping w_1 of the IFS for the "dragon"in Table 1 into $(w_1 w_1 w_1, w_1 w_1 w_2, w_1 w_2)$, and the new IFS $W' = \{R^2;\ w_1 w_1 w_1,\ w_1 w_1 w_2,\ w_1 w_2,\ w_2\}$ is equivalent to the original IFS $W = \{R^2; w_1, w_2\}$. Table 4

Table 2
IFS code for coral

w_i	a_i	b_i	c_i	d_i	e_i	f_i
1	0.307692	−0.531469	−0.461538	−0.293706	5.401953	8.655175
2	0.307692	−0.076923	0.153846	−0.447552	−1.295248	4.152990
3	0	0.545455	0.692308	−0.195804	−4.893637	7.269794

Table 3
IFS code for fern

w_i	a_i	b_i	c_i	d_i	e_i	f_i
1	0	0	0	0.16	0	0
2	0.85	0.04	−0.04	0.85	0	1.6
3	0.2	−0.26	0.23	0.22	0	1.6
4	−0.15	0.28	0.26	0.24	0	0.44

Table 4
Equivalent IFS code for dragon

w_i	a_i	b_i	c_i	d_i	e_i	f_i
1-a	0.409460	0.585012	−0.441325	0.492851	−4.682884	0.792306
1-b	−0.165779	0.143135	−0.350368	−0.446332	0.869093	5.474304
1-c	−0.057834	0.322995	−0.419636	−0.437105	1.043725	6.718995
2	0.088272	0.520988	−0.463889	−0.377778	0.785360	8.095795

lists the new IFS code W' for the fractal "dragon". Since Eq. (13) holds for all of affine mappings in W', we can compute a tight bounding box of the fractal "dragon" using Theorem 3.

3. The new algorithm

As we have seen in the previous section, we are able to find a tight bounding box of the attractor A of an affine IFS W. Given a box B_0 such that $W(B_0) \in B_0$, we know that the sequence $\{M(W^i(B_0))\}_{i=0}^n$ converges to the minimum bounding box $M(A)$ of the attractor A. Thus $W^i(B_0)$ with large i values will be close to $M(A)$. Let the corners of B_0 be b_1, b_2, b_3, b_4. Then the supreme values of the x- and y-coordinates of points in $W^i(B_0)$ will fall in $W^i(\{b_1, b_2, b_3, b_4\})$. Thus, we compute $W^i(\{b_1, b_2, b_3, b_4\})$ to find the bounding box $M(W^i(B_0))$. However, the computation of $W^i(\{b_1, b_2, b_3, b_4\})$ is expensive. It requires exponential time. For example, the computation for a tight bounding box $M(W^{100}(\{b1, b2, b3, b4\}))$ of the fractal "fern" in Table 3 needs to compute 4×4^{100} points (corners) which is not feasible by exhaustive computations. Fortunately, the computations for most of the points are redundant, and can be reduced using the following lemma.

Lemma 2. *Let* $W = \{R^2; w_1, w_2, \ldots, w_n\}$ *be an iterated function system of affine mappings in* R^2. *Let* B_0 *be an arbitrary box and* $W(B_0) \subseteq B_0$. *Then*

$$w_j(W^i(B_0)) \subseteq w_j(W^{i-1}(B_0)), \quad \text{for } i \geqslant 1, \ j = 1, \ldots, n. \tag{16}$$

Proof. This lemma is a natural extension of Theorem 3, because $W^i(B_0) \subseteq W^{i-1}(B_0)$ implies that $w_j(W^i(B_0)) \subseteq w_j(W^{i-1}(B_0))$. □

Fig. 5 shows a tree representation of computing $W^2(B_0)$. Let $xmax_1$ be the maximum x-coordinate of $W(B_0)$ that comes from one of the x-coordinates of $w_2(B_0)$. Then the x-coordinates of the points $w_1(B_0)$ are all less than $xmax_1$. Moreover, by Lemma 2, the x-coordinates of points in $w_1 W(B_0) = \{w_1 w_1(B_0), w_1 w_2(B_0), \ldots, w_1 w_n(B_0)\}$ are always less than $xmax_1$, too. As a result, the computation of $w_1 W(B_0)$ may be unnecessary for the supreme x-value of $W^2(B_0)$. Therefore, we compute $w_2 W(B_0)$ first. Let $xmax_2$ be the maximum x-coordinate of $w_2 W(B_0)$. If all of the x-coordinates of the points $w_1(B_0)$ are still less than $xmax_2$, then it confirms the uselessness of computing $w_1(W(B_0))$. On the contrary, we perform $w_1(W(B_0))$ to check whether $xmax_2$ is still the supreme x-value of $W^2(B_0)$. In this way, we can build a dynamic programming algorithm as the following pseudocode.

```
BboxIFS(W, Epsilon, MaxLayer)
{
/*Each mapping w[i] in the IFS W is composed of six coefs:
  a[i], b[i], c[i], d[i], e[i], f[i] as equation-2 */
/*Calculate the initial BBox */
  S = Max(Max(|a[i]| + |b[i]|), Max(|c[i]| + |d[i]|)
  E = Max(|e[i]|)
  F = Max(|f[i]|)
  K = (E + F)/(1-S)
  Box = (K, -K, K, -K)
  IdentMap = [1, 0, 0, 1, 0, 0]
/*Refine the BBox*/
  while(1)
  {
  XMax = XMaxAtLayerN(IdentMap, 1)
  XMin = XMinAtLayerN(IdentMap, 1)
  XMax = YMaxAtLayerN(IdentMap, 1)
  XMax = YMinAtLayerN(IdentMap, 1)
  NewBox = (Xmax, XMin, YMax, YMin)
  if (Box-NewBox < Epsilon)
    break
  else
    Box = NewBox
  }
  return NewBox
}
XMaxAtLayerN(CurMap, Layer)
{
/* We need another set of mappings t[i] for the
   combination of w*w*...*w*/
  if (CurMap has not been visited)
  {
    for i = 1 to n
    {
      t[i] = CurMap*w[i]
      apply the mapping t[i] to the four corners of Box,
      denoted as t[i](Box)
      compute the max x-coordinates of t[i](Box),
      denoted as SubXMax[Layer][i]
    }
  }
  else
  {
    (XMax, i) = Max(SubXMax[Layer][1], ...,
      SubXMax[Layer][n])
    t[i] = CurMap*w[i]
    Update t[i](Box) and SubXMax[Layer][i]
  }
  (XMax, i) = Max(SubXMax[Layer][1], ...,
    SubXMax[Layer][n])
  if (Layer == MaxLayer) return XMax
  SubXMax[Layer][i] = XMaxAtLayerN(t[i], Layer + 1)
  (XMax, j) = Max(SubXMax[Layer][1], ...,
    SubXMax[Layer][n])
  return XMax
}
```

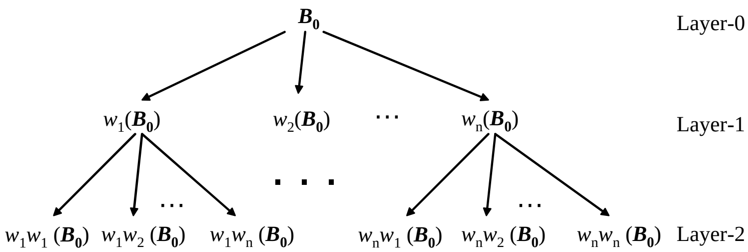

Fig. 5. Transformed sub-boxes of an initial bounding box for two layers.

Table 5
Computed areas of the proposed algorithm and Hart's algorithm

Fractal	Proposed bounding box algorithm		Hart's bounding ball algorithm	
	$(X_{min}, X_{max}, Y_{min}, Y_{max})$	Area value	Origin (X, Y), Radius	Area value
Dragon	$(-6.20, 6.15, -0.21, 10.11)$	127.45	(2.17, 4.31), 19.41	1183.59
Coral	$(-4.95, 5.00, -0.10, 10.03)$	100.79	(0.46, 5.01), 11.43	410.43
Fern	$(-2.18, 2.66, 0, 10.00)$	48.40	(0.03, 1.44), 9.28	270.55

Before we preform the function BboxIFS( ), we have to verify if Eq. (13) holds for each mapping in the IFS W. If there exists some mapping for which Eq. (13) does not hold, we have to compute an equivalent IFS instead. For convenience, we compute the supreme values of x and y values separately in BboxIFS(), and we only represent the routine XMaxAtLayerN() here. The three other routines XMinAtLayerN(), YMaxAtLayerN(), and YMinAtLayerN() are similar. The routine XMaxAtLayerN() represents the structure of dynamic programming for recursively computing the supreme values. It's noticeable that the computed values SubXMax[Layer][i] have to be saved for the use in the next iteration. Only the mappings that provide current supreme values are updated recursively.

4. Experimental results

We have computed bounding boxes for the fractals "dragon", "coral", and "fern" with the parameters Epsilon = 0.0001, and Maxlayer = 100. For comparison, we have also computed the bounding balls for these fractals using Hart and DeFanti's algorithm. Table 5 lists the corners of the computed bounding boxes, the origins and the radii of the computed bounding balls. Also the area values of these two kinds of bounding extent are listed. The results show that the areas of the computed bounding boxes are much smaller than those of the bounding balls by using Hart and DeFanti's algorithm.

5. Conclusion

We have proposed an algorithm to compute the tight bounding boxes of the attractors of iterated function systems. Firstly we compute a loose initial bounding box by a simple equation. If the initial box is not available, we have to decompose the iterated function system. Then we refine the bounding box until a predefined condition is reached. The computed bounding boxes are so tight that we can minimize the required space of drawing fractals. Using the bounding box algorithm, we have two choices. We only need to compute once the very tight bounding boxes with strict conditions (small epsilon and deep layers), the computed bounding box can then be permanently associated with its IFS code. Or we can compute a bounding box with loose conditions (larger epsilon and fewer layers) in real-time applications.

Acknowledgements

The authors would like to express the deepest gratitude to the anonymous reviewers for their valuable comments, which were very useful in improving the presentation of this article.

Appendix. The proof of the success of the initial box B_0

In Theorem 4, we give an upper bounding box $B = (-K, K; -K, K)$, and we claim $W(B_0) \subset B_0$ holds. We have to show that for any point $p = (x, y) \in R^2$, $-K \leqslant x, y \leqslant K$, then the x- and y-coordinates of $W(p)$ lie in the interval $[-K, K]$. That is, for any point $(x', y') = q = w_i(p)$, $i = 1 \ldots n$, $-K \leqslant x', y' \leqslant K$.

(a) We illustrate $x' \leqslant K$ by using the notations as defined in Theorem 4.

$$\because\ x' = a_i x + b_i y + e_i \quad \text{and} \quad |x| \leqslant K,$$

$$\therefore\ x' \leqslant |a_i||x| + |b_i||y| + |e_i|,$$

$$\begin{aligned}\therefore\ x' &\leqslant |a_i|K + |b_i|K + E \\ &= (|a_i| + |b_i|)K + E \\ &= (|a_i| + |b_i|)\left(\frac{E+F}{1-S}\right) + E \\ &= \frac{(|a_i| + |b_i|)(E+F) + E(1-S)}{1-S} \\ &= \frac{(|a_i| + |b_i| - S)E + E + (|a_i| + |b_i|)F}{1-S} \\ &= \frac{E + (|a_i| + |b_i|)F}{1-S} + \frac{(|a_i| + |b_i| - S)E}{1-S} \\ &\leqslant \frac{E + (|a_i| + |b_i|)F}{1-S}\end{aligned}$$

since $|a_i| + |b_i| \leqslant S < 1\ \forall 1 \leqslant i \leqslant n$, so

$$\frac{(|a_i| + |b_i| - S)E}{1-S} \leqslant 0$$

$$\Rightarrow x' \leqslant \frac{E+F}{1-S} = K.$$

(b) Similarly, we illustrate $-K \leqslant x'$ as follows.

$$\because\ x' = a_i x + b_i y + e_i \quad \text{and} \quad |x| \leqslant K,$$

$$\therefore\ x' \geqslant -|a_i||x| - |b_i||y| - |e_i|,$$

$$\begin{aligned}\therefore\ x' &\geqslant -|a_i|K - |b_i|K - E \\ &= -(|a_i| + |b_i|)K - E \\ &= -(|a_i| + |b_i|)\left(\frac{E+F}{1-S}\right) - E \\ &= -\frac{(|a_i| + |b_i|)(E+F) + E(1-S)}{1-S} \\ &= -\frac{(|a_i| + |b_i| - S)E + E + (|a_i| + |b_i|)F}{1-S} \\ &= -\frac{E + (|a_i| + |b_i|)F}{1-S} - \frac{(|a_i| + |b_i| - S)E}{1-S} \\ &\geqslant -\frac{E + (|a_i| + |b_i|)F}{1-S}\end{aligned}$$

since $|a_i| + |b_i| \leqslant S < 1\ \forall 1 \leqslant i \leqslant n$, so

$$-\frac{(|a_i| + |b_i| - S)E}{1-S} \geqslant 0$$

$$\Rightarrow x' \geqslant -\frac{E+F}{1-S} = -K.$$

From (a) and (b), $-K \leqslant x' \leqslant K$ is proved. Similarly, we can show that $-K \leqslant y' \leqslant K$. Thus we have proved the transformed point $q = w_i(p)$ is in the box $B = Box(-K, K; -K, K)$, hence the condition $W(B_0) \subset B_0$ holds.

References

[1] Barnsley MF. Fractals everywhere, 2nd ed. New York: Academic Press, 1993.

[2] Bell S. Fractals: a fast, accurate and illuminating algorithm. Image and Vision Computing 1995;13(4):253–7.

[3] Chu HT, Chen CC. A fast algorithm for generating fractals. Proceedings ICPR'2000. Barcelona, Spain, 2000. p. 306–9.

[4] Monro DM, Dudbridge F. Rendering algorithms for deterministic fractals. IEEE Computer Graphics and Applications 1995;15(1):32–41.

[5] Hart JC, DeFanti TA. Efficient anti-aliased rendering of 3D linear fractals. Computer Graphics 1991;25(4):91–100.

[6] Canright D. Estimating the spatial extent of attractors of iterated function systems. Computers and Graphics 1994;18(2):231–8.

[7] Edalat A, Sharp DWN, While RL. An upper bound on the area occupied by a fractal. Proceedings ICASSP'95. 1995. p. 2443–6.

[8] Rice J. Spatial bounding of self-affine iterated function system attractor sets. Proceedings of the Conference of Graphics Interface GI'96. 1996. p. 107–15.