Bounding the Attractor of an IFS

Abbas Edalat, David W.N. Sharp and R. Lyndon While[1]

Imperial College Research Report DoC 96/5

email: {ae, dwns}@doc.ic.ac.uk, lyndon@cs.uwa.edu.au

April 16, 1999

Keywords: Algorithms, analysis of algorithms, iterated function systems, fractals.

Abstract

Fractal images defined by an iterated function system (IFS) are specified by a finite number of contractive affine transformations. In order to plot the attractor of an IFS on the screen of a digital computer, it is necessary to determine a bounding area for the attractor. Given a point on the plane, we obtain a formula for the radius of a circle centred on that point that contains the attractor of the IFS. We then describe an algorithm to find the point on the plane such that the bounding circle centred on that point has minimum radius.

1 Introduction

Iterated function system (IFS) fractal images, as popularised by Barnsley[1, 2], are constructed from sheared, reduced, rotated and displaced copies of themselves. For example, the *curly* image in Figure 1 is constructed from two transformed copies of itself: the blackened rightmost curl and the less black remainder. The blackened rightmost curl is a reduced, rotated and displaced copy of the whole image, produced by applying transformation T_1. The remainder of the image is produced by applying transformation T_2, which shrinks the whole image and rotates it anti-clockwise. The *curly* image is the attractor of the IFS specified by the transformations T_1 and T_2.

Various algorithms exist for plotting fractal images from their affine transformations[1, 4, 7, 8, 9, 10]. However, in order to plot a fractal image on the screen of a digital computer, all of these algorithms require advance knowledge of a *bounding area* inside which the image is known to lie. In this paper we derive a formula which expresses the dimensions of a bounding circle for the attractor of an IFS in terms of the parameters of its transformations. Following the approach of [5], in Section 3 we derive

[1]On sabbatical at Imperial College: normally at the Dept. of Computer Science, The University of Western Australia.

an upper bound for the radius of a bounding circle centred at an arbitrary (given) point on the plane. In Section 4 we describe an algorithm for determining the centre at which the radius of the bounding circle is minimum. Section 5 describes some results and Section 6 concludes the paper.

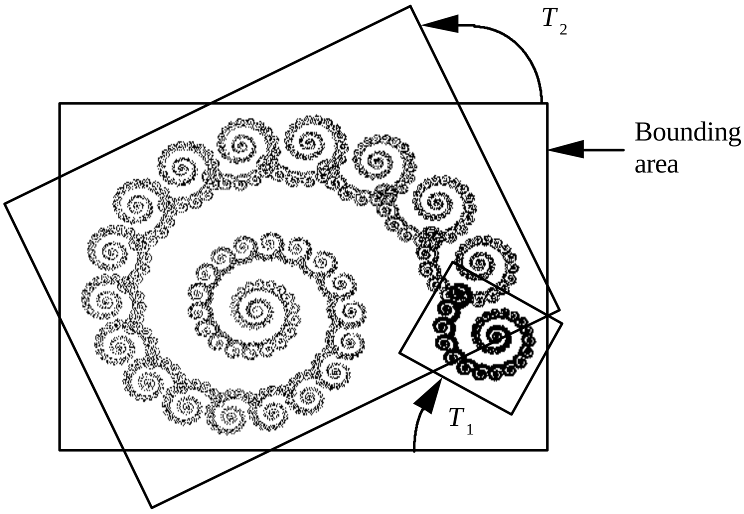

Figure 1 The fractal image *curly* and its transformations.

2 Definitions

An IFS is specified by n contractive, affine transformations T_i, $1 \le i \le n$. Each transformation T_i has the form $\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} a_i & b_i \\ c_i & d_i \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e_i \\ f_i \end{pmatrix}$. Each T_i has a fixpoint $\begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$ that is mapped to itself under the transformation, i.e. $\begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}$. Solving the equation defining $\begin{pmatrix} x_0 \\ y_0 \end{pmatrix}$ gives

$$\begin{pmatrix} x_0 \\ y_0 \end{pmatrix} = \begin{pmatrix} (b_i f_i + e_i(1 - d_i))/d \\ (c_i e_i + f_i(1 - a_i))/d \end{pmatrix}$$

where $d = (1 - a_i)(1 - d_i) - b_i c_i$.

The *attractor* of an IFS can be obtained as follows[6]. Let D be any disk with $T_i(D) \subseteq D$, $1 \le i \le n$. The attractor is given by

$$\cap_{m \ge 0} T^m(D)$$

where $T(A) = \cup_{i=1}^{n} T_i(A)$.

The effect of T_i on two points (x_1, y_1) and (x_2, y_2) is to map them to two points (x_1', y_1') and (x_2', y_2') that are closer together than the original points. The *contractivity factor* of T_i is the least number s_i such that $\sqrt{(x_1'-x_2')^2+(y_1'-y_2')^2} \le s_i\sqrt{(x_1-x_2)^2+(y_1-y_2)^2}$. It is well known that[see, for example, 3]

$$s_i = \sqrt{\alpha + \beta + \sqrt{(\alpha-\beta)^2+\gamma^2}}$$

where $\alpha = (a_i^2+c_i^2)/2$, $\beta = (b_i^2+d_i^2)/2$, $\gamma = a_ib_i + c_id_i$.

3 A bounding circle for the attractor of an IFS

A *bounding circle* for an IFS is a circle that contains the attractor of the IFS. Given a point $u=(x,y)$, we shall obtain a formula for the radius of a bounding circle centred on u. We start by considering one transformation of the IFS and then consider the IFS as a whole.

3.1 A bounding circle for one transformation

Consider the effect of a transformation T_i on an arbitrary point p that lies on or inside a circle B_i centred on u. T_i maps u to the point u_i and p to the point p_i, as shown in Figure 2. We denote the radius of B_i by $R_{B_i}(u)$.

1. B_i is a bounding circle for T_i if p_i lies on or inside B_i, i.e. if

$$up_i \le R_{B_i}(u)$$

2. By the triangular inequality, $up_i \le uu_i + u_ip_i$, and we have

$$up_i \le R_{B_i}(u) \text{ if } uu_i + u_ip_i \le R_{B_i}(u)$$

3. As T_i is contractive, $u_ip_i \le s_iup$, and we have

$$uu_i + u_ip_i \le R_{B_i}(u) \text{ if } uu_i + s_iup \le R_{B_i}(u)$$

4. As p lies on or inside B_i, $up \le R_{B_i}(u)$, and we have

$$uu_i + s_iup \le R_{B_i}(u) \text{ if } uu_i + s_iR_{B_i}(u) \le R_{B_i}(u)$$

Solving for $R_{B_i}(u)$ gives

$$R_{B_i}(u) \ge \frac{uu_i}{1-s_i}$$

Hence, B_i is a bounding circle for T_i if $R_{B_i}(u) \ge \dfrac{uu_i}{1-s_i}$.

Note that uu_i is easily calculated from the coordinates of u. Clearly, $R_{B_i}(u)$ is minimum when the two sides of the inequality have the same value. Figure 2 illustrates the construction.

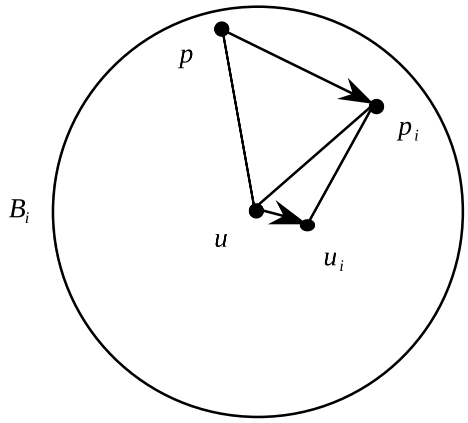

Figure 2 The circle B_i is a bounding circle for T_i if $R_{B_i}(u) \geq \frac{uu_i}{1-s_i}$.

3.2 A bounding circle for an IFS with n transformations

Applying the formula for $R_{B_i}(u)$ to each of the n transformations individually gives n concentric circles centred on u. Clearly, the largest of the concentric circles is a bounding circle B for the IFS. Its radius is given by

$$R_B(u) = \max_{i=1}^{n} \frac{uu_i}{1-s_i}$$

4 Choosing an optimal centre for the bounding circle

The quality of the upper bound given by the formula for $R_B(u)$ depends acutely on the choice of u. We shall describe an algorithm for deriving the optimal choice of u for a given IFS. We start by considering an IFS with two transformations and then extend the algorithm to the general case.

4.1 An optimal centre for an IFS with two transformations

We wish to find $u = (x, y)$ where $R_B(u)$ is minimum. Let

$$z_i(x, y) = R_{B_i}^2(x, y) = \frac{(a_i x + b_i y + e_i - x)^2 + (c_i x + d_i y + f_i - y)^2}{(1-s_i)^2}$$

The value of $z_i(x, y)$ for a transformation T_i forms an elliptic cone as (x,y) is varied, with the vertex of the cone at the fixpoint of T_i. The cones for the two transformations defining *curly* are shown in Figure 3.

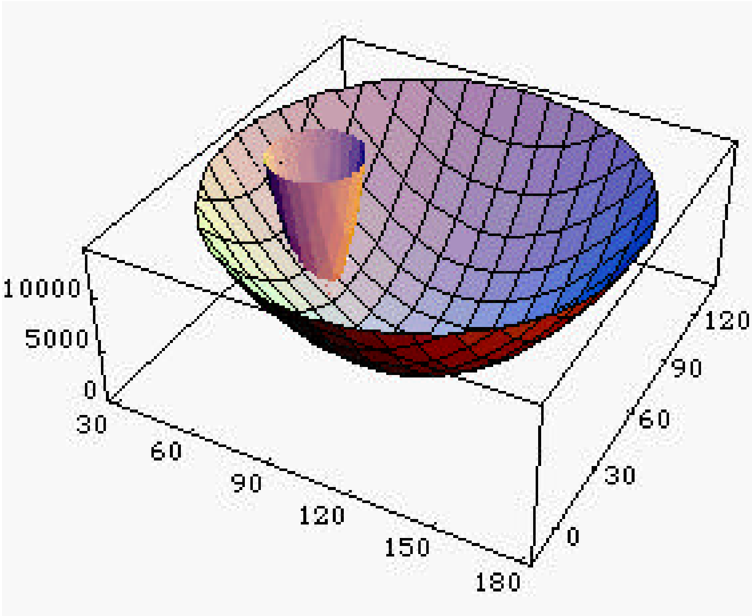

Figure 3 The elliptic cones for the two transformations defining *curly* for $z_i(x, y) \leq 14500$. The intersection of the two cones is clear.

The two cones meet in a conic section. The point $(x,y,z_i(x,y))$ in the intersection that gives the minimum value of $R_B(x,y)$ is the optimal centre for the bounding circle. Using Lagrange's multipliers, this is obtained by minimising the function

$$g(x, y, l) = z_1(x,y) + l\,(z_1(x,y) - z_2(x,y))$$

with respect to x, y and l and solving numerically for x and y.

4.2 An optimal centre for an IFS with $\boxed{n}$ transformations: Algorithm $\boxed{\text{E}}$

In the case of three transformations, the optimal centre lies either at the intersection of the three cones (if they meet), or at the minimum of one of the three pairwise intersections. For each candidate centre (x,y), we calculate $R_B(x,y)$, and the candidate with the smallest value is the optimal centre.

This algorithm generalises to an IFS with more than three transformations in the obvious manner. For n transformations, the optimal centre lies either at the intersection of any k cones, $3 \leq k \leq n$ (if they meet), or at the minimum of one of the ${}^{n}C_{2}$ pairwise intersections. For each candidate centre (x,y), we calculate $R_B(x,y)$, and the candidate with the smallest value is the optimal centre. We call this Algorithm E.

Note that if a set of k cones, $3 \le k < n$, does not meet, then no superset of that set meets. Algorithm E is $O({}^{n}C_2 + {}^{n}C_3)$, i.e. $O(n^3)$ in the best case and $O(\sum_{i=2}^{n} {}^{n}C_i)$, i.e. $O(2^n)$ in the worst case, for n transformations. The worst case is encountered rarely.

5 Results

We exercised Algorithm E on a range of fractals to get an idea of the tightness of the bounding circles. We used Algorithm E to generate a bounding circle for a fractal, then plotted the fractal and determined the tightest possible bounding circle using a search technique.

We found that, for typical fractals, Algorithm E generates bounding circles that are at most 70% larger than the tightest possible. Some examples are illustrated in Figures 4(a)—(d). The outer circle in each figure is derived using Algorithm E and the inner circle is the tightest possible bounding circle.

Numerical results are given in Table 1 and the parameters for the fractals are given in Appendix A.

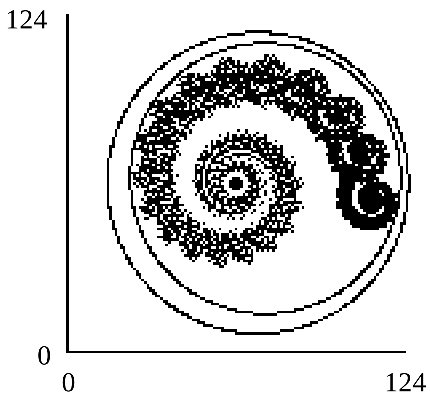

Figure 4(a) Bounding circles for *curly*.

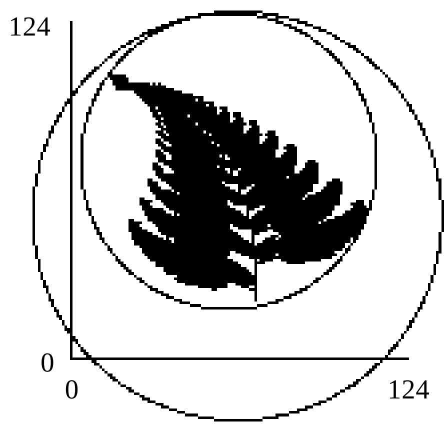

Figure 4(b) Bounding circles for *fern*.

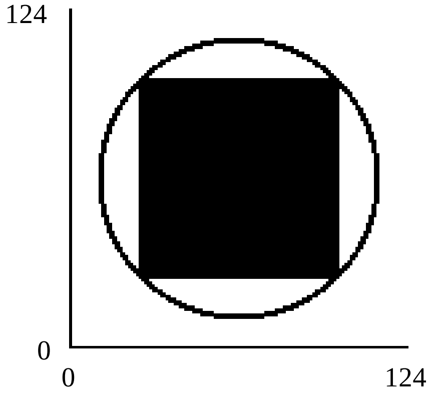

Figure 4(c) Bounding circles for *square*.

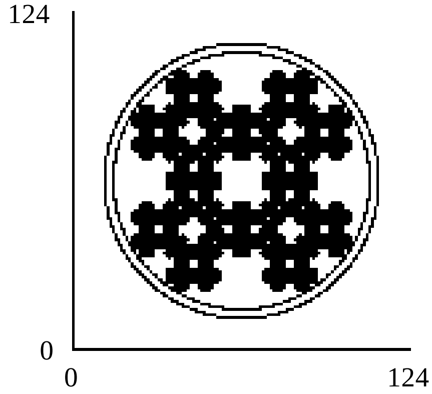

Figure 4(d) Bounding circles for *square* rotated $p / 4$.

	Algorithm E		Search		Diff.
	Radius	Centre	Radius (±1)	Centre	
Curly	56.4	(70.2, 61.7)	51	(73, 60)	11%
Fern	76.5	(61.2, 71.5)	55	(58, 51)	39%
Square	51.4	(62.0, 62.0)	52	(62, 62)	0%
Square + p/4	51.4	(62.0, 62.0)	48	(62, 62)	7%

Table 1 Bounding circles for the fractals illustrated.

The *square* fractal is instructive. The basic *square* is specified by four transformations, each one mapping the square to one of its quadrants. None of the transformations rotate the square and Algorithm E is able to derive the tightest possible bounding circle exactly (within measuring error). If we introduce a rotation of $p / 4$ to each transformation, we get the attractor shown in Figure 4(d), which clearly occupies a smaller area than the original. However, Algorithm E derives the same bounding circle: in fact it derives the same bounding circle whatever rotation is introduced.

6 Conclusions

Plotting a fractal image from its transformations on the screen of a digital computer requires advance knowledge of a bounding area for the image. We have derived a formula that expresses the radius of a bounding circle for a fractal in terms of the parameters of the transformations specifying the fractal and described an algorithm for finding the point at which a bounding circle should be centred to minimise its radius. Experiments indicate that this algorithm gives reasonable results for typical fractals. This will improve the effectiveness of algorithms that plot IFS fractal images.

Acknowledgements

We would like to thank Peter Potts for his help with Mathematica.

References

[1] BARNSLEY, M.F.: "Fractals Everywhere", Boston: Academic Press, 1988.

[2] BARNSLEY, M.F. and SLOAN, A.D.: "A Better Way to Compress Images", Byte Magazine, January 1988.

[3] EDALAT, A.: "Power Domain Algorithms for Fractal Image Decompression", Dept. of Computing Internal Report DoC 93/44, Imperial College, 1993.

[4] EDALAT, A.: "Power Domains and Iterated Function Systems", Information and Computation, Vol. 124, pp. 182-97, 1996.

[5] EDALAT, A., SHARP, D.W.N. and WHILE, R.L.: "An Upper Bound on the Area Occupied by a Fractal", Proc. ICASSP '95, Detroit, Michigan, Vol. 4, pp. 2443-6, May 1995.

[6] FALCONER, K.: "Fractal Geometry: Mathematical Foundations and Applications", Wiley, Chichester, 1990.

[7] HEPTING, D., PRUSINKIEWICZ, P. and SAUPE, D.: "Rendering Methods for Iterated Function Systems", Proc. IFIP Fractals, 1990.

[8] MONRO, D.M., DUDBRIDGE, F. and WILSON, A.: "Deterministic Rendering of Self-affine Fractals", IEE Colloquium on the Application of Fractal Techniques in Image Processing, IEE Publications, December 1990.

[9] SHARP, D.W.N. and CRIPPS, M.D.: "Parallel Algorithms That Solve Problems by Communication", Proc. 3rd IEEE Symposium on Parallel and Distributed Processing, Dallas, Texas, pp. 87-94, December 1991.

[10] SHARP, D.W.N. and WHILE, R.L.: "Pattern Recognition Using Fractals", Proc. Int. Conf. on Parallel Processing, Illinois, Vol. III, pp. 82-9, August 1993.

Appendix A

A.1 *Curly*

Curly is specified by two transformations with the following parameters.

	a	**b**	**c**	**d**	**e**	**f**
T_1	-0.18	0.18	-0.18	-0.18	113.46	90.52
T_2	0.89	0.33	-0.33	0.89	-13.64	27.28

A.2 *Fern*

Fern is specified by four transformations with the following parameters.

	a	**b**	**c**	**d**	**e**	**f**

T_1	0.85	0.09	-0.02	0.85	0.68	3.54
T_2	0.00	0.00	0.00	0.16	67.64	86.34
T_3	0.20	-0.60	0.10	0.22	115.66	60.37
T_4	-0.15	0.64	0.11	0.24	11.50	66.89

A.3 *Square*

Square is specified by four transformations. With a rotation of q , they have the following parameters.

	a	**b**	**c**	**d**	**e**	**f**
T_1	$\frac{\cos q}{2}$	$-\frac{\sin q}{2}$	$\frac{\sin q}{2}$	$\frac{\cos q}{2}$	$31(\sqrt{2}-\cos q+\sin q)$	$31(\sqrt{2}-\cos q-\sin q)$
T_2	$\frac{\cos q}{2}$	$-\frac{\sin q}{2}$	$\frac{\sin q}{2}$	$\frac{\cos q}{2}$	$31(\sqrt{2}-\cos q+\sin q)$	$31(4-\sqrt{2}-\cos q-\sin q)$
T_3	$\frac{\cos q}{2}$	$-\frac{\sin q}{2}$	$\frac{\sin q}{2}$	$\frac{\cos q}{2}$	$31(4-\sqrt{2}-\cos q+\sin q)$	$31(\sqrt{2}-\cos q-\sin q)$
T_4	$\frac{\cos q}{2}$	$-\frac{\sin q}{2}$	$\frac{\sin q}{2}$	$\frac{\cos q}{2}$	$31(4-\sqrt{2}-\cos q+\sin q)$	$31(4-\sqrt{2}-\cos q-\sin q)$