

Spatial Bounding of Self-Affine Iterated Function System Attractor Sets

Jonathan Rice

Image Synthesis Group
Department of Computer Science
Trinity College
Dublin 2
Ireland

Abstract

An algorithm is presented which, given the parameters of an Iterated Function System (IFS) which uses affine maps, constructs a closed ball which completely contains the attractor set of the IFS. These bounding balls are almost always smaller than those computed by existing methods, and are sometimes much smaller. The algorithm is numerical in form, involving the optimisation of centre-point and radius relationships between the overall bounding ball and a set of smaller, contained balls which are derived by analysis of the contractive maps of the IFS. The algorithm is well-behaved, in that although it converges toward an optimal ball which it only achieves in the limit, the process may still be stopped after any finite number of steps, with a guarantee that the sub-optimal ball which is returned will still bound the attractor.

Keywords: Fractals, Iterated Function Systems, IFS, Minimisation Surfaces.

1 Introduction

Iterated Function Systems (IFSs) are a prominent technique for the generation and analysis of fractal structures [7, 21, 1, 4]. In IFS theory, fractals arise as the attractor sets of an iteration process on a certain function, the Hutchinson operator, which combines the effects of a given set of contractive maps. It is generally useful, in most applications of IFSs, to be able to place reliable *a priori* bounds on the spatial extent of such attractor sets, before commencing the iteration process. In a rendering context, for example, this could be used to determine if an attractor is visible in a given viewport, and to appropriately rescale the viewing coordinate system if it is not. However, the problem of how to compute such spatial bounds, before rendering, is not straightforward. Analysis of each contractive map in isolation, for example, yields no locational information except for the maps' fixed points, which together give only a general idea of the position of the whole attractor set. To solve the problem more exactly, what is needed is a method which analyses the characteristics of all maps of the Hutchinson operator simultaneously, since the attractor is the very result of their mutual interaction. In this paper, a new algorithm along these lines is proposed which can compute a reliable bounding ball around the attractor set, for the most commonly-used type of IFS – those based on affine transformations. In comparison with existing techniques for bounding attractor sets, the new algorithm almost always calculates smaller bounding structures, its efficiency being only marginally surpassed in certain special cases.

We begin by reviewing the basic theory of IFSs, in Section 2. In Section 3, we progress to consideration of the bounding problem and formalise our goal of constructing a bounding ball. In Section 4, previous approaches are studied, and their performance is assessed. The new algorithm is then presented in Section 5. In Section 6, comparison of bounding efficiency is made between the new algorithm and the previous approaches. Finally, in Section 7, conclusions are presented and possible extensions of the current work are indicated.

2 Iterated Function Systems

Let us briefly review the definition and basic properties of Iterated Function Systems. For more detailed exposition of the material presented here, and in particular for proofs of results, the reader may refer to virtually any text on fractal geometry [1, 4, 9, 16, 15, 17].

Let (X, d) be a complete metric space. Then an IFS F may be simply defined by a finite set of contractive mappings on that space. The usual notation [1, p.82] leaves the metric d understood:

$$F = \{X; w_1, w_2, \dots, w_m\},$$

for some $m \in \mathbb{N}$, where each w_i , for $i = 1, \dots, m$, is a contractive Lipschitz function on X :

$$d(w_i(x), w_i(y)) \leq s_i \cdot d(x, y), \quad \forall x, y \in X, \quad (1)$$

for some constant s_i , with $0 \leq s_i < 1$. Any suitable value s_i in the above equation is termed a *contractivity factor* of w_i (also called *contraction ratio*). Usually, however, we take s_i to be the infimum of these possible values, and refer to it simply as *the* contractivity factor of w_i .

Typically, in the literature, the maps w_i are affine functions on Euclidean spaces, and the distance metric d used is simply the Euclidean distance. In this paper, we shall concern ourselves exclusively with such *self-affine* IFSs. For IFSs of this sort, the component map contractivity factors are straightforward to calculate. Let $w_i(x) = A_i x + B_i$, for $x \in X$, $i = 1, \dots, m$, expressed in the usual manner in terms of a linear matrix A_i and a translation vector B_i . Then the contractivity factor s_i may be calculated as the square root of the maximum eigenvalue of the matrix product $A_i^T A_i$ [15, p.372]. Since $A_i^T A_i$ is a symmetric matrix, none of its eigenvalues can have an imaginary component. Thus, one of the simpler eigen-system solution algorithms, such as Jacobi Transformation [18, pp.360–367], may be applied. This calculation

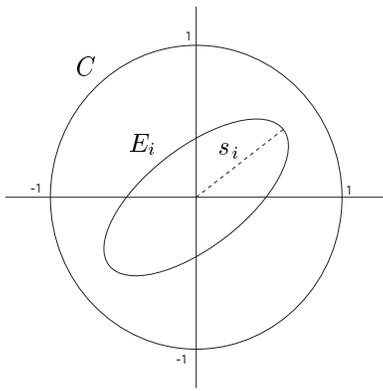


Figure 1: Characterisation of s_i as the magnitude of the semi-major axis of $E_i = \{A_i x \mid x \in C\}$.

is based on a characterisation of s_i as a measure of the direction of weakest contractivity of the map w_i , computed as the modulus of the semi-major axis of the ellipse E_i formed by transformation of the unit circle C by A_i (see Fig. 1).

Let $\mathcal{H}(X)$ be the space of non-empty compact subsets of X . We now define “set-wise” versions of the maps of F to operate on $\mathcal{H}(X)$. Let

$$\begin{aligned} w_i &: \mathcal{H}(X) \rightarrow \mathcal{H}(X), \\ w_i(A) &= \{w_i(x) \mid x \in A\}, \end{aligned}$$

for all $A \in \mathcal{H}(X)$, $i = 1, \dots, m$. We may now define the Hutchinson operator, H , as

$$\begin{aligned} H &: \mathcal{H}(X) \rightarrow \mathcal{H}(X) \\ H(A) &= \bigcup_{i=1}^m w_i(A) \quad \forall A \in \mathcal{H}(X). \end{aligned} \quad (2)$$

It may be proven that H is a contraction mapping on the metric space $(\mathcal{H}(X), h)$, where h is the usual Hausdorff metric on $\mathcal{H}(X)$ with respect to the underlying metric d . Because H is contractive and $\mathcal{H}(X)$ is complete, iteration of H converges to a unique fixed point in $\mathcal{H}(X)$, the *attractor* of the IFS, which we denote by \mathcal{A} :

$$\begin{aligned} H(\mathcal{A}) &= \mathcal{A}, \\ \lim_{n \rightarrow \infty} H^n A &= \mathcal{A}, \quad \text{for any } A \in \mathcal{H}(X). \end{aligned} \quad (3)$$

Since $\mathcal{A} \in \mathcal{H}(X)$, it is by definition a non-empty compact subset of X . Except for some specially-constructed cases, \mathcal{A} will be fractal in form – that is, its “fractal dimension”, as computed by one of a range of standard methods, is a non-integral number. Such sets are very complex in structure, with detail at all scales.

3 Spatial Bounding of IFS Attractor Sets

Before committing resources to rendering \mathcal{A} (by the Random Iteration Algorithm [2] or some other method [12, 14]), is it possible to make any global statements about

the general location and extent of this set? Yes, it is. An important first basic observation is that since the attractor is by definition a compact set, it must occupy a finite local region of space. In this section, we consider the problem of how to place a bound on the possible position and size of this region in some way, by constructing a structurally-simple *bounding set* which must fully enclose the attractor.

Probably the simplest type of structure to use as a bounding set is the *ball*, a generalisation to arbitrary dimensions of the familiar two-dimensional concept of a disc. Balls are convenient to use here since they allow certain problems arising from directionality in the IFS maps to be avoided¹. Let $B((X, d), x, r)$ denote a ball in the metric space (X, d) , centred at $x \in X$, with radius $r \in \mathbb{R}^+$:

$$B((X, d), x, r) = \{y \mid d(x, y) \leq r, y \in X\}.$$

Usually, the metric space may be assumed from the context and the above ball is thus referred to simply as $B(x, r)$.

We now have the language to state our goal precisely. Let $F = \{X; w_1, w_2, \dots, w_m\}$ be an IFS, as before, with attractor set $\mathcal{A} \in \mathcal{H}(X)$. Then we seek parameters $x_m \in X$ and $r_m \in \mathbb{R}^+$, such that

$$B(x_m, r_m) \supset \mathcal{A},$$

with r_m a minimum.

4 Previous Approaches

As part of a paper dealing with the rendering of IFS attractor sets, Hart and DeFanti [12] propose an iterative algorithm which generates, based on an arbitrary initial ball B_0 , a sequence of balls $\{B_i\}_1^\infty$ which converges toward a bounding ball B_∞ containing \mathcal{A} . Hart has since revised and updated this paper [11], but the operation of the bounding algorithm has remained unchanged. Hart and DeFanti define the algorithm only for \mathbb{R}^3 , but their approach is easily generalised to a Euclidean space X of arbitrary dimension. Let

$$B_i = B(x_i, r_i),$$

for $i = 0, 1, \dots, \infty$, with $x_i \in X$, and $r_i \in \mathbb{R}^+$. The initial values x_0 and r_0 are arbitrary (the authors suggest using the origin and unity, respectively). Then new centre and radius values are produced using the following formulae:

$$x_{i+1} = \frac{1}{m} \sum_{j=1}^m w_j(x_i), \quad (4)$$

$$r_{i+1} = \max_{j=1 \dots m} \{d(w_j(x_i), x_{i+1}) + s_j r_i\}, \quad (5)$$

where s_j is the contractivity factor of map w_j .

This algorithm performs well in practical terms for the application to which Hart and DeFanti put it – ray-traced

¹This simplification is at the expense of bounding efficiency, since the use of directional information should help to more closely bound the attractor. We shall touch on this topic again in Section 6.

rendering of fairly dense three-dimensional IFS attractor sets. The algorithm converges quickly toward a bounding sphere which is fairly well centred on their example attractor sets and whose radius is close to the minimum value necessary to bound the attractor for the given centre point. By recursively instantiating this sphere during ray-casting, the authors are able to adaptively refine their knowledge of the spatial layout of an attractor set to any required degree of precision. A ray is considered to intersect the attractor if it enters a bounding sphere whose diameter implies a suitably small associated pixel coverage. Their system manages to produce beautiful, highly-detailed images of some interesting fractal sets.

However, impressive images notwithstanding, the underlying bounding algorithm has two undesirable characteristics. Firstly, the centre point of even the limit ball, x_∞ , is, in general, not as well centred on \mathcal{A} as it could be. Ideally, x_∞ should be located at the *geometric centre*, c , of \mathcal{A} — the value of $x \in X$ such that the bounding radius expression $\max \{d(x, y) \mid y \in \mathcal{A}\}$ is minimised. It may be proven that x_∞ is in fact located at the barycentre of the invariant measure of the IFS². For general IFSs whose maps are not symmetrical in action, this has the effect of biasing x_∞ away from the geometric centre of \mathcal{A} (see Fig. 2), although it must remain within a tight convex hull drawn around the set, due to the convexity property of barycentric combination. The second drawback with the algorithm is that although the limit ball \mathbf{B}_∞ can be proven to bound \mathcal{A} , it is actually impossible to guarantee this condition for any of the finite balls \mathbf{B}_i in the sequence. The basic problem here is that the ball sequence always converges *directionally* towards \mathbf{B}_∞ from one side³ (see Fig. 3), and it can happen that the particular combination of centre and radius values necessary to bound \mathcal{A} only occur together in the limit. We do not have space here for formal proofs of all these results. They may be found in a separate report [20].

Canright [5] describes a bounding algorithm which is rather different in approach from that of Hart and DeFanti. Again, balls are used as bounding structures, but instead of trying to bound the whole attractor \mathcal{A} with a single ball, Canright attempts to construct a set of smaller balls \mathbf{B}_i , each of which bounds the image of \mathcal{A} under one of the IFS maps w_i , for $i = 1, 2, \dots, m$:

$$\mathbf{B}_i \supset w_i(\mathcal{A}),$$

where the centre of each ball \mathbf{B}_i is the fixed point of the associated map w_i . The subsets $w_i(\mathcal{A})$ are termed *attractortlets*, since they are contracted copies of the overall attractor. Since $\bigcup_{i=1}^m w_i(\mathcal{A}) = \mathcal{A}$, by the definition of the Hutchinson operator, the union of all of these balls, the

²We do not have space here for a full definition of the invariant measure. Loosely speaking, it is an importance weighting function defined on \mathcal{A} which also remains invariant under the action of the IFS maps. More details can be found in most texts on fractal geometry [1, 4, 16, 15, 17].

³There is a good reason for this behaviour. As we see in the Appendix, if all component IFS maps are affine, then the centre point sequence equation (Eq. 4) is itself a contractive affine function. Thus, the point sequence $\{x_i\}$ is a linear progression.

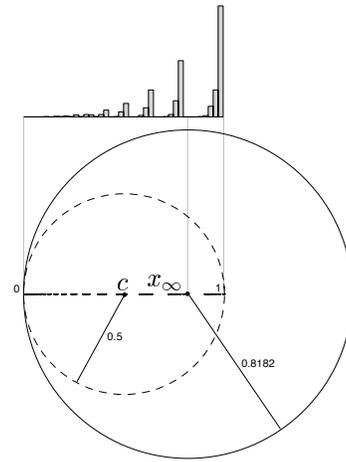


Figure 2: Inefficient bounding ball centre location by Hart and DeFanti's algorithm for the IFS $\{\mathbb{R}; w_1(x) = 0.8x, w_2(x) = 0.1x + 0.9\}$. The extent of \mathcal{A} is the unit interval. Optimal ball centre is at $c = 0.5$, allowing a radius of 0.5. Computed ball centre (the barycentre of the invariant measure indicated by the superimposed histogram) is at $x_\infty = 0.8182$, necessitating a larger radius of 0.8182 in order to bound \mathcal{A} . (Note that in this diagram, the extents of these 1D balls are illustrated in two dimensions by their minimally-enclosing circles.)

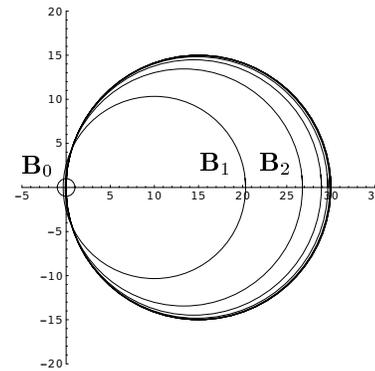


Figure 3: Hart and DeFanti ball sequence for the 1-D IFS $\{\mathbb{R}; w_1(x) = \frac{1}{3}x, w_2(x) = \frac{1}{3}x + 20\}$, starting with the initial ball $\mathbf{B}_0 = B(0, 1)$. The extent of \mathcal{A} is the interval $[0, 30]$. If i is finite, then $x_i + r_i < 30$, implying $\mathbf{B}_i \not\supset \mathcal{A}$.

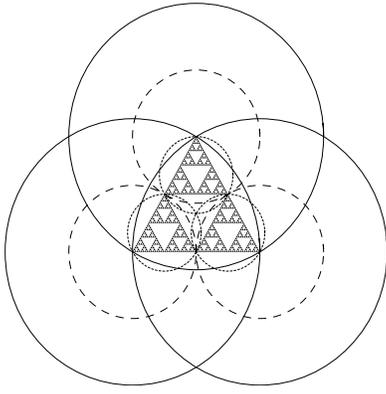


Figure 4: Attractorlet bounding ball efficiency: optimal balls (fine dashed line); smallest possible balls when map fixed points (at the outer corners of the triangle) are used as centers (medium dashed line); Canright's computed balls (solid line).

so-called *envelope*, \mathcal{E} , must contain \mathcal{A} :

$$\mathcal{E} = \bigcup_{i=1}^m B_i \supset \mathcal{A}.$$

Canright proves that if certain relationships between the radii of the balls are satisfied, then $\mathcal{E} \supset \mathcal{A}$. Let f_i and s_i be the fixed point and contractivity factor, respectively, of map w_i , for $i = 1, \dots, m$. Then the radius of the attractorlet bounding ball centred at f_i, r_i , is required to satisfy the following condition:

$$r_i \geq \frac{s_i(1 + s_j)}{1 - s_i s_j} d(f_i, f_j), \quad (6)$$

for all $j = 1, 2, \dots, m$, where $i \neq j$. Canright presents two algorithms for calculating a suitable set of radius values. The first method is iterative, but returns its answer in a finite number of steps. The second is not explicitly iterative (it does contain a sort). Both methods are computationally equivalent.

However, although \mathcal{E} may be a provably valid bounding structure, it is generally not very efficient, for two reasons. Firstly, the map fixed points tend not to be very well-centred on their associated attractorlets, forcing the resulting bounding ball radii to be larger than necessary. Secondly, in general, the algorithms do not even compute as efficient radii as they might for these already sub-optimal centre points. In fact, one can see from Eq. 6 that for a pair of IFS maps whose contractivity factors are both arbitrarily close to unity (not an unreasonable construction), the consequent computed radius value can increase without limit! Fig. 4 illustrates these issues, using the familiar Sierpinski Triangle as an example, while Fig. 5 shows a simple example of a 2-map IFS where the radius values are clearly unreasonably large. For more rigorous analysis of these inefficiencies in Canright's algorithm, the interested reader is again referred to [20].

5 The New Algorithm

In this section, a new algorithm for IFS bounding ball construction is presented which improves on the perfor-

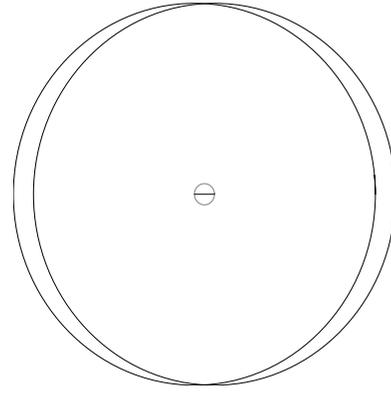


Figure 5: Inefficient bounding by Canright's algorithm for the weakly-contractive 1-D IFS $\{\mathbb{R}; w_1(x) = 0.9x, w_2(x) = 0.9x + 0.1\}$. In this case, the attractor \mathcal{A} is the interval $[0, 1]$, shown at the centre of the diagram as a small line. This set is optimally bounded by the ball $B(0.5, 0.5)$, shown in grey. The computed envelope \mathcal{E} consists instead of the union of the two much larger balls, $B(0, 9)$ and $B(1, 9)$, shown in black.

mance of the previous methods. Like Hart and Defanti's approach, the method is iterative, gradually approaching a bounding ball of minimum radius. Here, however, computed balls are always guaranteed to contain the attractor, even after a finite number of iterations. What is more, the new algorithm converges toward a bounding ball whose efficiency can be matched by Hart and DeFanti only for their best case (well-centred barycentre). The new algorithm also compares very favourably with Canright's algorithms, usually producing a bounding ball which is much more efficient than Canright's envelope structure.

Our treatment here is divided into three parts. First, we state the algorithm in general terms. We then show that a minimisation surface which is at the core of the algorithm will have a simple form, as long as the IFS maps obey certain conditions. Finally, we show that these conditions are met by self-affine IFSs, and that some optimisations can be made for this case.

5.1 General Statement of the Algorithm

We start our description of the algorithm by deriving some basic conditions for a ball which, if fulfilled, guarantee that the ball will contain the attractor of a given IFS.

Theorem 1 *Let $F = \{X; w_1, w_2, \dots, w_m\}$ be an IFS, with attractor set \mathcal{A} , as before. Let s_i be the contractivity factor of map w_i , as before. Suppose that, for some $x \in X$ and $r \in \mathbb{R}^+$,*

$$B(x, r) \supset B(w_i(x), s_i r), \quad (7)$$

for $i = 1, \dots, m$. Then,

$$B(x, r) \supset \mathcal{A}.$$

Proof: We construct an inductive proof, that if the condition of Eq. 7 holds, then $B(x, r)$ is a superset of its

image under repeated application of the Hutchinson operator H . That is, we prove

$$B(x, r) \supset H^j(B(x, r)), \quad \forall j = 1, 2, \dots, \infty. \quad (8)$$

Firstly, let us assume that

$$B(x, r) \supset H^j(B(x, r)), \quad \forall j = 1, 2, \dots, k, \quad (9)$$

for some $k \in \mathbb{N}$. As induction step, we attempt to prove that this equation holds for $k + 1$. For this, it is sufficient to prove the added term, that

$$B(x, r) \supset H^{k+1}(B(x, r)).$$

From Eq. 9, we have that $B(x, r) \supset H^k(B(x, r))$. Since all maps w_i , $i = 1, \dots, m$, are contractive, they are thus continuous [1, p.80]. Therefore, $w_i(B(x, r)) \supset w_i(H^k(B(x, r)))$. Thus, $\bigcup_{i=1}^m w_i(B(x, r)) \supset \bigcup_{i=1}^m w_i(H^k(B(x, r)))$, or, to rephrase, $H(B(x, r)) \supset H^{k+1}(B(x, r))$. But Eq. 9 also implies that $B(x, r) \supset H(B(x, r))$. Therefore,

$$B(x, r) \supset H^{k+1}(B(x, r)),$$

as required. Finally, we prove Eq. 9 to be true for $k = 1$:

$$B(x, r) \supset H(B(x, r)).$$

That is,

$$B(x, r) \supset \bigcup_{i=1}^m w_i(B(x, r)).$$

For each map w_i , $i = 1, \dots, m$, we may write down that

$$d(w_i(x), w_i(y)) \leq s_i r, \quad \forall y \in B(x, r).$$

Therefore,

$$B(w_i(x), s_i r) \supset w_i(B(x, r)) \quad \text{for } i = 1, \dots, m.$$

But, from our original conditions, we know that $B(x, r) \supset B(w_i(x), s_i r)$, for $i = 1, \dots, m$. Thus,

$$\begin{aligned} B(x, r) &\supset w_i(B(x, r)) && \text{for } i = 1, \dots, m, \\ \Rightarrow B(x, r) &\supset H(B(x, r)). \end{aligned}$$

So by induction, Eq. 8 is true. In particular, since $H^\infty(B(x, r)) = \mathcal{A}$, this means that $B(x, r) \supset \mathcal{A}$. \square

So, we have established Eq. 7 as a sufficient condition to guarantee that a ball contains the IFS attractor. We shall now examine how we may construct a ball with such properties. Let the condition of Eq. 7 hold, for a ball $B(x, r)$. Then this implies

$$d(x, w_i(x)) + s_i r \leq r, \quad \text{for } i = 1, \dots, m. \quad (10)$$

This relationship is illustrated in Fig. 6, which shows example balls for a simple three-map IFS in \mathbb{R}^2 . Solving for r , we get

$$r \geq \frac{d(x, w_i(x))}{1 - s_i}, \quad \text{for } i = 1, \dots, m.$$

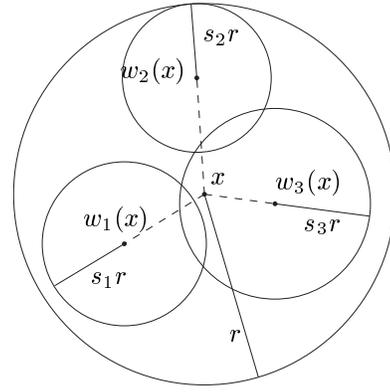


Figure 6: Sub-ball centre point and radius relationships.

Let us define

$$g_i(x) = \frac{d(x, w_i(x))}{1 - s_i}, \quad (11)$$

for $x \in X$, $i = 1, \dots, m$. Then, we can recast r as a function which returns the minimal admissible ball radius value for a given centre point x :

$$r(x) = \max_{i=1, \dots, m} \{g_i(x)\}. \quad (12)$$

In order to illustrate what these radius functions look like for a self-affine IFS, let us consider some plots, for an arbitrary example IFS F :

$$F = \{\mathbb{R}^2; w_1, w_2, w_3\},$$

where

$$\begin{aligned} w_1(x) &= \begin{bmatrix} -0.0804 & -0.3861 \\ -0.0901 & 0.3549 \end{bmatrix} x + \begin{bmatrix} 0.7526 \\ 0.2875 \end{bmatrix}, \\ w_2(x) &= \begin{bmatrix} -0.2114 & -0.1886 \\ -0.3753 & 0.1625 \end{bmatrix} x + \begin{bmatrix} 0.5991 \\ 0.9209 \end{bmatrix}, \\ w_3(x) &= \begin{bmatrix} -0.2118 & 0.0994 \\ -0.3108 & -0.0624 \end{bmatrix} x + \begin{bmatrix} 0.1811 \\ 0.4772 \end{bmatrix}. \end{aligned}$$

Fig. 7 shows contour plots for the associated radius functions g_1, g_2, g_3 , as well as the combined function r . They illustrate a general feature, that for affine IFS maps, the associated g functions, and hence, the combined radius function r , are “basin-like” in form.

Fig. 8 illustrates perhaps a little more clearly the form of r . One can see here how the graphs of the g functions intersect to form an upper surface which is the graph of r . This surface is a patchwork of different “zones of influence” of each of the g functions.

Now, applying Theorem 1, we reach the interesting result that for any chosen point $x \in X$, the ball $B(x, r(x))$ must contain the attractor of the IFS, \mathcal{A} . Let

$$r_m = \min_{x \in X} \{r(x)\}.$$

Assume that this minimum occurs at a point $x = x_m$. Then the minimal ball we seek is $B(x_m, r(x_m))$.

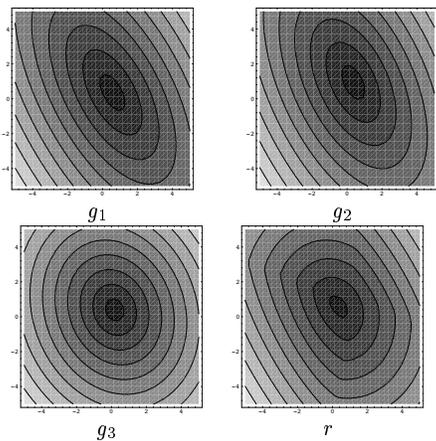


Figure 7: Contour plots of g and r functions.

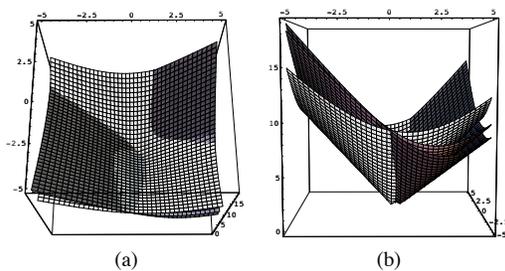


Figure 8: The radius function r is the maximum of the functions g_1, g_2, g_3 – that is, the top surface of the graph in (a) above. Note the different “zones of influence” due to g_1 (dark grey), g_2 (medium grey) and g_3 (white). In (b) we can see the same structure from a lower view angle.

The core of the new algorithm is the minimisation of r over X . In some simple cases, when dealing with low-dimensional spaces and IFSs with few maps, it may be possible to solve for the minimum analytically. In general, however, a numerical procedure must be used. In this context, the “basin-like” nature of r proves to be quite useful. When the underlying IFS maps are affine, it is possible to formally prove that r has no local maxima or minima and that the global minimum must occur at a unique point of X (i.e. there are no plateaus) [20]. These properties are of considerable importance in the selection of a minimisation algorithm. Coping with local minima is often the single-most difficult aspect of the minimisation process, and one that must usually be carefully tuned to the problem at hand. Here, we can ignore this issue. Additionally, although plateaus in a minimisation function can usually be classed as more of an irritation than a real impediment to solution, it is nice to know that this feature is also absent from r , as it enables us to characterise our solution for r_m as being unique. Taking these two factors into account, therefore, we can be optimistic that a relatively simple, “off-the-shelf” minimisation algorithm should be effective in our search for r_m .

Before going further, we note that if we assume that we are using the Euclidean distance metric, we may gain in efficiency by minimising the square of r (easily computed

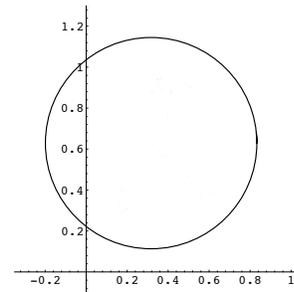


Figure 9: Bounding ball returned by new algorithm.

with the dot product), rather than r itself. This allows us to avoid repeated use of the computationally-costly square root function. Since r is constructed from maximisation and minimisation operations on strictly-positive quantities, the squaring operation does not change the location of the minimum point x_m . Let $\delta_i(x) = x - w_i(x)$, for $x \in X, i = 1, \dots, m$. Then we define

$$r^2(x) = \max_{i=1, \dots, m} \left\{ \frac{\delta_i(x) \cdot \delta_i(x)}{(1 - s_i)^2} \right\}.$$

Therefore, we can now minimise the simpler r^2 to find the location of the minimal ball centre x_m and then compute r_m as the square root of the value of r^2 at that point.

Now let us consider the minimisation algorithm itself. We will not engage in an exhaustive review of suitable algorithms, as the field is quite broad (see Press et al. [18], for example) and many different approaches should be feasible. It should be noted, however, that some algorithms may not cope well with the “composite” nature of the minimisation surface. For instance, direction set approaches which allow only fixed stepping directions (say, parallel to the coordinate axes) have been found to be prone to getting stuck in the valley-like seams between the regions of influence of the IFS maps. A version of Nelder and Mead’s flexible Downhill Simplex Method [18] has been tested on the problem and seems to work well, quickly returning a good approximation to the global minimum point.

In order to start the minimisation process somewhere near the attractor, we can use the fixed point of one of the maps of the IFS, or the average of these points, or even Hart and DeFanti’s measure barycentre x_∞ (see Appendix) as its starting point. The initial step size (characteristic scale) of the minimisation algorithm can be set to a certain fraction (say, $\frac{1}{10}$) of the value of r at the starting point. As a termination condition, we can require that the distance between successive step points should drop below a certain small fraction of the current value of r .

As an example, application of this algorithm to the IFS F whose radius function plots we saw in Figs. 7 and 8 yields the bounding circle $B((0.317, 0.629), 0.516)$, shown circumscribing the attractor in Fig. 9. One may verify that the centre of the circle is at the location of the minimum value of r in those plots.

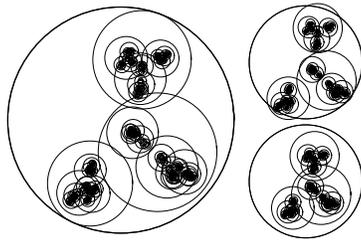


Figure 10: Just-touching bounding-ball hierarchy of the new algorithm (on left). If the radius is smaller, not all balls nest (top right). If larger, the fit is poorer (bottom right).

6 Bounding Ball Efficiency

In this section, we examine the efficiency of the balls computed by the new algorithm, and compare its performance with the algorithms of Hart and DeFanti and Canright.

Let us begin by investigating the optimality of the bounding balls produced by the new algorithm – is it possible to fit a smaller bounding ball around the attractor than the one which the algorithm computes? In general, the answer is “yes”. For example, referring to Fig. 9, we can see that it is possible to manually draw a smaller bounding circle around the attractor. However, although the new algorithm may not compute the absolutely smallest bounding ball around a particular IFS attractor set, it *does* calculate the smallest bounding ball that may be hierarchically instantiated. That is, the computed ball is the smallest that may be instantiated recursively in a bounding hierarchy of balls [12], where each parent ball is guaranteed to contain its child balls (see Fig. 10). The fact that the algorithm computes the *smallest* ball that may be so instantiated is a consequence of its original construction as the minimal ball $B(x, r)$ such that $B(x, r) \supset B(w_i(x), s_i r)$, for $i = 1, \dots, m$.

However, although the top-level bounding ball is not in itself optimal, the new algorithm still performs well compared to the approaches of Hart and DeFanti and Canright. As an empirical test, all three algorithms were applied to a sample set of two thousand randomly-generated two-dimensional IFSs. Each IFS had between three and ten maps, each map being constructed as the composition of random x and y shears and scales (with parameters between -1 and 1), a random rotation, and a random translation within the region $[0, 1] \times [0, 1]$. Because of the shears, such a map is not guaranteed to be contractive, although with the above choice of parameters, it is likely. If a given map turned out not to be contractive (as determined by the eigenvalue method of Section 2), it was discarded and another candidate map was generated in its place, and so on, until a valid map was found.

Results are plotted in Fig. 11 in the form of a cumulative frequency graph of the efficiency E in bounded area of the new algorithm’s computed balls, relative to the areas of both Hart and DeFanti’s balls and Canright’s envelope. E is simply the area of each algorithm’s bounding structure (ball or envelope, respectively, for Hart and DeFanti and for Canright), divided by the area of the new al-

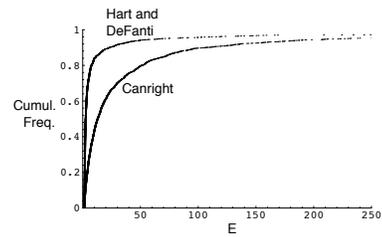


Figure 11: Cumulative frequency distribution of the area bounding efficiency E of the new algorithm, relative to both Canright’s and Hart and DeFanti’s algorithms.

gorithm’s ball. The area of Canright’s envelope was computed by reference to Green’s Theorem [22], summing area contributions linked to arcs of the boundary of the envelope.

The graph can be read as follows: take a certain value of E on the horizontal axis and follow it up to the curve of one of the algorithms. The associated value on the vertical axis is the proportion of the sample IFSs for which the new algorithm produced a bounding ball whose area was less than E times smaller than the old algorithm’s bounding structure (ball or envelope, respectively, for Hart and DeFanti’s and Canright’s algorithms). Not all data points are plotted on the graphs, as for both of the algorithms, there were some very large outlying values of E . For instance, in the worst test result for Hart and DeFanti’s algorithm, it computed a bounding ball which had almost 600,000 times the area of that returned by the new algorithm. The worst result for Canright’s algorithm was lower than this (perhaps surprisingly, given the concerns we expressed in Section 4 over its potentially large ball radius values), with a value of $E = 25,044$.

Generally speaking, we can see from the graphs that the new algorithm performs better than either of the previous ones. In an attempt to define an overall index of performance relative to the previous algorithms, we compute the median of both distributions⁴. This results in a value of 2.16 for Hart and DeFanti and 12.5 for Canright. These are the “typical” gains in efficiency in bounded area that one could hope to see when using the new algorithm for general two-dimensional IFSs. Note that in higher dimensions, the typical efficiency of the new algorithm (in terms of ratios of n -dimensional volumes) will be greater, since the volumes will increase in proportion to the n -th power of the radius.

Fig. 12 shows an enlargement of the graph around $E = 1$. This plot highlights some interesting features of the distributions of E . Firstly, for a small number of IFSs (1% of the sample set), Canright’s envelope structure actually has less area than the new algorithm’s ball (that is, $E < 1$). The best such value in the data set is $E = 0.658$. Such a situation can occur for certain “optimal” IFSs for Canright’s algorithms whose fixed points are well-centred on their attractorlets (see Fig. 13, for example).

A second interesting observation from the enlarged

⁴For highly-skewed distributions like these, this is generally a more descriptive statistic than the mean [6, p.30].

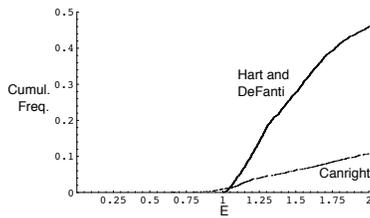


Figure 12: Enlargement (aspect ratio not maintained) of the plots of E (Fig. 11) around the value $E = 1$. Canright’s solution better the new algorithm ($E < 1$) in 1% of cases, whereas Hart and DeFanti’s can theoretically just equal it ($E = 1$).

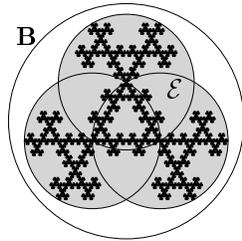


Figure 13: An optimal IFS for Canright’s algorithm. The envelope region \mathcal{E} (in grey) has less area than the new algorithm’s proposed bounding ball \mathbf{B} .

graph is that it appears that the lower bound of E for Hart and DeFanti’s algorithm is exactly unity. That is, it is possible that their algorithm can produce balls which are as small as those of the new algorithm, but no smaller. The reason for this rather precise result is that Hart and DeFanti’s limit ball can be formulated in terms of the new algorithm’s radius function r , evaluated at their limit center point x_∞ (see Appendix). Thus, their limit ball is a (usually sub-optimal) point in the solution space of the new algorithm.

7 Conclusions

The algorithm presented is an improvement over existing methods for calculating the bounds of an IFS attractor set. It computes a ball in finite time which reliably bounds the attractor and is smaller than the optimal balls of previous methods in almost all cases. Our empirical comparison of algorithms suggests that for two-dimensional IFSs, one can expect the new algorithm to return a bounding ball whose area is typically under one half that of Hart and DeFanti’s ball and under one twelfth that of Canright’s envelope structure. Perhaps more importantly, however, there does not appear to be any upper limit on the area ratios for either of these algorithms – their worst cases can be arbitrarily bad. What is more, for higher-dimensional spaces, such as \mathbb{R}^3 , we can expect the disparities in n -dimensional volume (Lebesgue Measure) to grow with r^n .

The algorithm is quite simple in form and should be relatively easy to implement. The most difficult part of this task will probably be the coding of the radius function

minimisation algorithm, but it should be possible to apply any of a number of off-the-shelf algorithms to this problem, as we have done with the Downhill Simplex method. This process is greatly aided by the fact that the minimisation surface we are dealing with has a provably simple “basin-like” form, with no local maxima or minima, and no plateaus. Finally, we note that although the algorithm does not in general compute the *minimal-sized* bounding ball around the attractor, it does compute the smallest-sized ball which may be hierarchically instantiated. This feature may be useful for certain applications, such as ray-tracing, and also when attempting to reason mathematically about the bounding hierarchy.

As regards applications of the algorithm, there are many in the field of IFS theory. Most uses of IFSs could benefit from better knowledge of the spatial bounds of attractor sets, even if only for the purpose of appropriately sizing and centering images of these sets for display on a computer screen. The algorithm was originally developed to serve as the core of a ray-tracing rendering system for IFSs, much along the lines of Hart and DeFanti’s work [12], and has proved to be useful there.

There are several interesting avenues of research which arise from this work. Firstly, the new algorithm could be generalised so that the overall bounding ball \mathbf{B} is required to contain only its ellipsoidal images under the Hutchinson operator (i.e. $\mathbf{B} \supset H(\mathbf{B})$), rather than the set of balls used here. Typically, this should result in a smaller top-level ball, since the bounding hierarchy will now consist of tighter nested ellipsoids. Such an extension is highly desirable since it would greatly increase the efficiency of our ray-tracing application. Secondly, the algorithm could be extended to handle more complex forms of IFS, such as Recurrent IFSs [3], Language-Restricted IFSs [19] or Mutually-Recursive Function Systems [8], by solving the cyclic radius relationships in such a structure’s transformation graph. Both of these topics are subjects of current research by the author. Finally, it would be interesting to investigate the application of the algorithm to some more exotic IFS variants, perhaps using non-Euclidean distance metrics [1] or nonlinear IFS maps [10, 13].

A Analytic Solution for Hart and DeFanti’s Limit Ball

Hart and DeFanti’s limit bounding ball $\mathbf{B}_\infty = B(x_\infty, r_\infty)$, which their algorithm approximates by an iterative process, can in fact be directly solved for analytically. Let $w_i(x) = A_i x + B_i$, for $i = 1, \dots, m$, be the component maps of a self-affine IFS, as before. We define $\mathbf{A} = \frac{1}{m} \sum_{i=1}^m A_i$, and $\mathbf{B} = \frac{1}{m} \sum_{i=1}^m B_i$. Then x_∞ is simply the fixed point of the affine function $f(x) = \mathbf{A}x + \mathbf{B}$:

$$x_\infty = (I - \mathbf{A})^{-1}\mathbf{B}.$$

The reader may have noted a certain similarity between Hart and DeFanti’s radius production equation (Eq. 5) and the new algorithm’s original radius bounding inequality (Eq. 10). In fact, the two equations become exactly equivalent as the limit of Eq. 5 is reached ($i \rightarrow \infty$). Thus,

$$r_\infty = r(x_\infty),$$

where r is the radius function of the new algorithm (Eq. 12). This means that (x_∞, r_∞) is a point on the new algorithm's radius minimisation surface. Since all such points correspond to provably-valid bounding balls around the attractor, this result provides us with proof that Hart and DeFanti's limit ball is indeed valid. It also gives us a preferable, direct method for computing this ball: solve for x_∞ as above, then evaluate r at this point.

References

- [1] M. F. Barnsley. *Fractals Everywhere*. Academic Press, London, 1988.
- [2] M. F. Barnsley and S. Demko. Iterated function systems and the global construction of fractals. In *Proc. Royal Soc. London*, volume A399, pages 243–275, 1985.
- [3] M. F. Barnsley, J. H. Elton, and D. P. Hardin. Recurrent iterated function systems. *J. Constructive Approximation*, 5(1):3–31, 1989.
- [4] Michael F. Barnsley and Lyman P. Hurd. *Fractal Image Compression*. A.K. Peters Ltd., Wellesley, Mass., 1993.
- [5] D. Canright. Estimating the spatial extent of attractors of iterated function systems. *Computers and Graphics*, 18(2):231–238, 1994.
- [6] C. Chatfield. *Statistics for Technology*. Chapman and Hall, London, 1983.
- [7] S. Demko, L. Hodges, and B. Naylor. Construction of fractal objects with iterated function systems. In B. A. Barsky, editor, *Computer Graphics (SIGGRAPH '85 Proceedings)*, volume 19, pages 271–278, July 1985.
- [8] S. Dube. Using fractal geometry for solving divide-and-conquer recurrences. *J. Austral. Math. Soc. Ser. B*, 37:145–171, 1995.
- [9] Kenneth Falconer. *Fractal Geometry – Mathematical Foundations and Applications*. John Wiley and Sons, Chichester, 1990.
- [10] E. Gröller. Modeling and rendering of nonlinear iterated function systems. *Computers and Graphics*, 18(5):739–748, 1994.
- [11] J. C. Hart. Linear fractals in 3d computer graphics. In *SIGGRAPH '95 Course 33 Notes: Procedural Modeling, Texturing, and Animation Techniques*, pages 5.49–5.69, Electronic Visualization Laboratory, Univ. Illinois at Chicago, August 1995. ACM SIGGRAPH.
- [12] J. C. Hart and T. A. DeFanti. Efficient anti-aliased rendering of 3D linear fractals. In Thomas W. Sederberg, editor, *Computer Graphics (SIGGRAPH '91 Proceedings)*, volume 25, pages 91–100, July 1991.
- [13] Siu Chi Hsu and Irene H. H. Lee. Drawing and animation using skeletal strokes. In Andrew Glassner, editor, *Proceedings of SIGGRAPH '94 (Orlando, Florida, July 24–29, 1994)*, Computer Graphics Proceedings, Annual Conference Series, pages 109–118. ACM SIGGRAPH, ACM Press, July 1994. ISBN 0-89791-667-0.
- [14] D. M. Monro and F. Dudbridge. Rendering algorithms for deterministic fractals. *IEEE Computer Graphics and Applications*, 15(1):32–41, January 1995.
- [15] H.-O. Peitgen, H. Jürgens, and Dietmar Saupe. *Fractals for the Classroom – Part One: Introduction to Fractals and Chaos*. Springer-Verlag, Berlin, New York, 1992.
- [16] H.-O. Peitgen and D. Saupe, editors. *The Science of Fractal Images*, London, 1988. Springer-Verlag.
- [17] Heinz-Otto Peitgen, Hartmut Jürgens, and Dietmar Saupe. *Chaos and Fractals: new Frontiers of Science*. Springer-Verlag, New York, 1992.
- [18] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. *Numerical Recipes in C: The Art of Scientific Computing*. Cambridge University Press, 2nd edition, 1992.
- [19] Przemyslaw Prusinkiewicz and Mark Hammel. Escape-time visualization method for language-restricted iterated function systems. In *Proceedings of Graphics Interface '92*, pages 213–223, 1992.
- [20] Jonathan Rice. Spatial bounding of self-affine iterated function system attractor sets. Technical Report TCD-CS-96-1, Dept. Computer Science, Trinity College, Dublin, Ireland, January 1996.
- [21] J. Stark and P. Bressloff. Iterated function systems. In M. Farge, J. C. R. Hunt, and J. C. Vassilicos, editors, *Wavelets, Fractals, and Fourier Transforms*. Clarendon Press, Oxford, 1993. Proceedings of the conference “Wavelets, Fractals, and Fourier Transforms: New Developments and New Applications” held at Newnham College, Cambridge, December 1990.
- [22] G. B. Thomas and R. L. Finney. *Calculus and Analytic Geometry*. Addison-Wesley, Reading, Mass., 8th edition, 1992.