Escape-time Visualization Method for Language-restricted Iterated Function Systems

Przemyslaw Prusinkiewicz and Mark Hammel
Department of Computer Science
University of Calgary
Calgary, Alberta, Canada T2N 1N4
e-mail: pwp|hammel@cpsc.ucalgary.ca

From *Proceeding of Graphics Interface '92*, pages 213–223, May 1992
Held in Vancouver, British Columbia, 11–15 May 1992

Escape-time Visualization Method for Language-restricted Iterated Function Systems

Przemyslaw Prusinkiewicz
Mark S. Hammel
Department of Computer Science
University of Calgary
Calgary, Alberta, Canada T2N 1N4

Abstract

The escape-time method was introduced to generate images of Julia and Mandelbrot sets, then applied to visualize attractors of iterated function systems. This paper extends it further to language-restricted iterated function systems (LRIFS's). They generalize the original definition of IFS's by providing means for restricting the sequences of applicable transformations. The resulting attractors include sets that cannot be generated using ordinary IFS's. The concepts of this paper are expressed using the terminology of formal languages and finite automata.

Keywords: fractal, iterated function system, escape-time method, graphics algorithm, formal language, finite automaton.

1. Introduction

Although mathematicians have explored the properties of fractals since the turn of century, they could not visualize the objects of their study without the aid of computers. Computer graphics made it possible to recognize the beauty of fractals, and turned them into an art form [13]. Peitgen and Richter [14] perfected and popularized images of Julia and Mandelbrot sets. Many of them were created using the escape-time method. In its original setting, it consisted of testing how fast points z outside the attractor diverged to infinity while iterating function $z \to z^2 + c$ in the complex plane. The resulting values were interpreted as colors in a two-dimensional image, or height values in a "fractal landscape" [15, Section 2.7].

The escape-time visualization method was extended from Julia sets to iterated function systems [12] by Barnsley [2] and Prusinkiewicz and Sandness [18]. This paper extends it further to language-restricted iterated function systems, introduced in [16]. They generalize the original definition of IFS's by providing means for restricting the sequences of applicable transformations to a particular set. The resulting attractors form a larger class than those generated using ordinary IFS's. The definition of an LRIFS leaves open the mechanism for sequencing transformations, thus LRIFS's incorporate the earlier generalizations committed to a particular mechanism, such as hierarchical IFS's [4], sofic systems [1], recurrent IFS's [3], Markov IFS's [21], mixed IFS's [5], controlled IFS's [17], and mutually recursive function systems [7, 8]. Several other authors considered similar generalizations without giving them a name. Visualization of the attractors of generalized IFS's has been addressed by Hart [9], referring to his earlier results with DeFanti [10].

This paper is organized as follows. Sections 2 and 3 summarize the background material related to formal languages and iterated function systems. Section 4 presents the escape-time method for IFS's in a way suitable for further extensions. Section 5 defines the language-restricted iterated function systems. The escape-time method is extended to LRIFS's in Section 6. A special case of regular languages is considered and illustrated using examples in Section 7. Section 8 summarizes the results.

2. Formal languages

An *alphabet* V is as a finite nonempty set of *symbols* or *letters*. A *string* or *word* over alphabet V is a finite sequence of zero or more letters of V, whereby the same letter may occur several times. The total number of letters in a word w is called its *length*, and denoted $\text{length}(w)$. The word of zero length is called the *empty word* and denoted ϵ. The *concatenation* of words $x = a_1 a_2 \ldots a_m$

and $y = b_1 b_2 \ldots b_n$ is the word formed by extending the sequence of symbols x with the sequence y, thus $xy = a_1 a_2 \ldots a_m b_1 b_2 \ldots b_n$. If $xy = w$ then the word x is called the *prefix* of w, denoted $x \prec w$. The remaining word y is called the *suffix*. We assume that the relation $\prec$ is reflexive, that is, $w \prec w$. The n-fold concatenation of a word w with itself is called its n-th *power*, and denoted w^n. By definition, $w^0 = \epsilon$ for any w. If $w = a_1 a_2 \ldots a_n$, then the word $w^R = a_n \ldots a_2 a_1$ is called the *mirror image* of w. It can be shown that $(xy)^R = y^R x^R$ for any words x and y.

The set of all words over V is denoted by $V^\star$, and the set of nonempty words by V^+. A *formal language* over an alphabet V is a set L of words over V, hence $L \subset V^\star$. The concatenation and mirror image of words are extended to languages as follows:

$$\begin{aligned} L_1 L_2 &= \{xy : x \in L_1 \,\&\, y \in L_2\}, \\ L^R &= \{w^R : w \in L\}. \end{aligned}$$

A language L is *prefix extensible* if there exists a word $v \in V^+$ such that $vL \subset L$. In other words, $vw \in L$ for every word $w \in L$. The *right derivative* of a language $L \subset V^\star$ with respect to a word $v \in V^\star$ is the language:

$$L//v = \{w \in V^\star : vw \in L\}.$$

The set of all prefixes of a language L is called the *prefix closure* of L:

$$\mathcal{P}(L) = \{x \in V^\star : (\exists w \in L)\, x \prec w\}.$$

3. Iterated function systems

Let $\langle X, d \rangle$ be a complete metric space with support X and distance function d (in this paper, we will only consider the plane with the Euclidean distance). A function $F : X \to X$ is called a *contraction* in X if there is a constant $r < 1$ such that

$$d(F(P), F(Q)) \leq r d(P, Q)$$

for all $P, Q \in X$. The parameter r is called the *Lipschitz constant* of F.

An *iterated function system* (IFS) in X is a quadruplet $\mathcal{I} = \langle X, \mathcal{F}, V, h, \rangle$, where:

- X is the underlying metric space,
- $\mathcal{F}$ is a set of contractions in X,
- V is an alphabet of contraction labels,
- $h : V \to \mathcal{F}$ is a *labeling function*, taking the letters of alphabet V to the contractions from $\mathcal{F}$.

In the literature, an IFS is usually defined as the pair $\langle X, \mathcal{F} \rangle$. We extend this definition by specifying the alphabet V and the labeling function h to facilitate the introduction of language-restricted IFS's in Section 5.

The function h is extended to words and languages over V using the equations:

$$\begin{aligned} h(a_1 a_2 \ldots a_n) &= h(a_1) \circ h(a_2) \circ \ldots \circ h(a_n), \\ h(L) &= \bigcup_{w \in L} h(w), \end{aligned}$$

where the symbol $\circ$ denotes function composition,

$$x \circ f_1 \circ f_2 \circ \cdots \circ f_n = f_n(\cdots(f_2(f_1(x)))\cdots).$$

The *attractor* of an IFS $\mathcal{I}$ is the smallest nonempty set $\mathcal{A} \subset X$, closed with respect to all transformations of $\mathcal{F}$, and closed in the set-theoretic sense. Hutchinson showed that the attractor of an arbitrary IFS always exists and is unique [12]. Consequently, it can be found by selecting a point $P \in \mathcal{A}$, and applying to it all possible sequences of transformations from $\mathcal{F}$:

$$\mathcal{A} = \mathrm{cl}(P \circ h(V^\star)),$$

where the symbol cl represents the set-theoretic closure of the argument set. There are several methods for finding the initial point $P \in \mathcal{A}$. For example, the fixed point of any transformation $F \in \mathcal{F}$ is known to belong to $\mathcal{A}$ [12].

A legible notation for specifying transformations is needed while defining particular IFS's. In this paper we express transformations by composing operations of translation, rotation, and scaling in an underlying Cartesian coordinate system. The following symbols are used:

- $t(a, b)$ is a translation by vector (a, b).
- $a(\alpha)$ is a rotation by (oriented) angle α with respect to the origin of the coordinate system. The angles are expressed in degrees.
- $s(r_x, r_y)$ is a scaling with respect to the origin of the coordinate system: $x' = r_x x$ and $y' = r_y y$. If $r_x = r_y = r$, we write $s(r)$ instead of $s(r, r)$.

For example, Figure 1 shows the attractor of an IFS $\mathcal{I} = \langle X, \mathcal{F}, V, h \rangle$, where the set $\mathcal{F}$ consists of two transformations:

$$\begin{aligned} F_1 &= s\left(\frac{\sqrt{2}}{2}\right) \circ r(45), \\ F_2 &= s\left(\frac{\sqrt{2}}{2}\right) \circ r(135) \circ t(0, 1). \end{aligned}$$

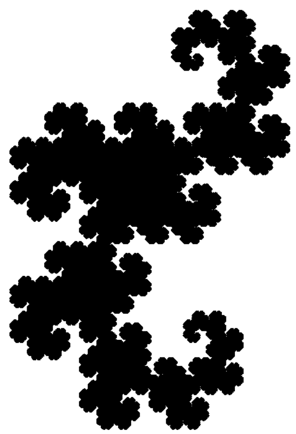


Figure 1: The dragon curve

4. The escape-time method

Consider an IFS $\mathcal{I} = \langle X, \mathcal{F}, V, h \rangle$, where all functions $F \in \mathcal{F}$ are invertible. Let $\tilde{h}(a)$ denote the inverse of the contraction $F = h(a) \in \mathcal{F}$, or $\tilde{h}(a) = (h(a))^{-1}$. The function $\tilde{h}$ is extended to words and languages over V in a way similar to h:

$$\begin{aligned} \tilde{h}(a_1 a_2 \ldots a_n) &= \tilde{h}(a_1) \circ \tilde{h}(a_2) \circ \ldots \circ \tilde{h}(a_n), \\ \tilde{h}(L) &= \bigcup_{w \in L} \tilde{h}(w). \end{aligned}$$

A *trajectory* of a point Q with respect to a word $w \in V^\star$ is the set:

$$\mathrm{Tr}(Q, w) = \{Q \circ \tilde{h}(x) : x \prec w\}.$$

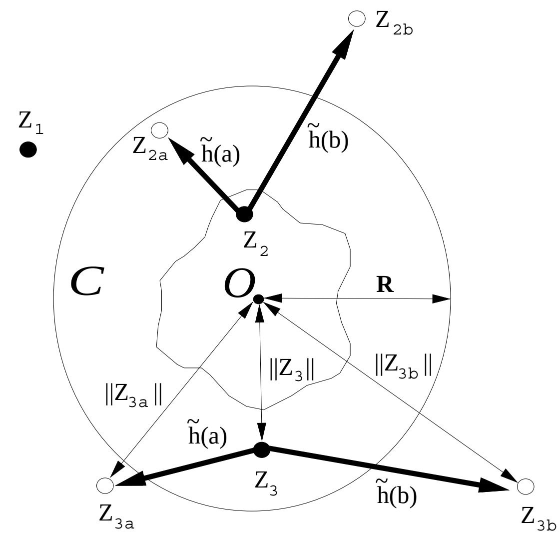
The length of w is referred to as the length of the trajectory. The escape-time method for visualizing the attractor of $\mathcal{I}$ is based on the following Theorem, proven in [18]:

Theorem 1. (a) If a starting point Q belongs to the attractor $\mathcal{A}$ of an IFS $\mathcal{I}$, there exists an infinitely long trajectory entirely included in $\mathcal{A}$. (b) If the point Q does not belong to $\mathcal{A}$, all trajectories diverge to infinity.

To estimate the speed with which the divergence occurs, we enclose the attractor in a circle. Since attractors of IFS's are bounded, it is always possible to find a circle $\mathcal{C}$ of a finite radius R, completely enclosing $\mathcal{A}$. The escape time of a point $Q \notin \mathcal{A}$ is then defined as the length of the longest trajectory included in $\mathcal{C}$:

$$E_1(Q) = \max_{w \in V^\star} \{\mathrm{length}(w) : \mathrm{Tr}(Q, w) \subset \mathcal{C}\}.$$

According to this definition, the function $E_1(Q)$ is integer-valued. In order to represent the escape time



- $\mathrm{res}(Z_1, a) = \mathrm{res}(Z_1, b) = 0$, since $Z_1 \notin \mathcal{C}$,
- $\mathrm{res}(Z_2, a) = 0$, since $Z_2 \circ \tilde{h}(a) = Z_{2a} \in \mathcal{C}$,
- $\mathrm{res}(Z_3, a) > \mathrm{res}(Z_3, b)$, since $\|Z_{3a}\| < \|Z_{3b}\|$.

Figure 2: Illustration of the residual terms $\mathrm{res}(Z, a)$.

with a higher precision, Hepting *et al.* [11] introduced a residual term that reflects the distance between the last point in the escape trajectory $Tr(Q, w)$ and the border of circle $\mathcal{C}$:

$$E_2(Q) = \max_{wa \in V^\star} \{\mathrm{length}(w) + \mathrm{res}(Q \circ \tilde{h}(w), a) : \mathrm{Tr}(Q, w) \subset \mathcal{C}\}.$$

Let $Z = Q \circ \tilde{h}(w)$. The function $\mathrm{res} : X \times V \to [0, 1)$ is defined as follows:

$$\mathrm{res}(Z, a) = \begin{cases} \dfrac{\log R - \log \|Z\|}{\log \|Z \circ \tilde{h}(a)\| - \log \|Z\|} & \text{if } Z \in \mathcal{C} \text{ and } Z \circ \tilde{h}(a) \notin C, \\ 0 \quad \text{otherwise.} \end{cases} \tag{1}$$

The norm symbol $\|Z\|$ denotes the distance between point Z and the center O of circle $\mathcal{C}$, thus $\|Z\| = d(Z, O)$. The function $\mathrm{res}(Z, a)$ has the following properties (Figure 2):

- it takes a nonzero value if point Z lies inside the circle $\mathcal{C}$ and its image $Z \circ \tilde{h}(a)$ lies outside this circle;
- it tends to 0 if the point Z approaches the boundary of circle $\mathcal{C}$, and to 1 if the image $Z \circ \tilde{h}(a)$ approaches this boundary.

Observe that, when the length of the longest trajectory included in $\mathcal{C}$ is incremented as a result of moving the starting point Q towards the attractor, the largest residual term changes its value from 1 to 0. Consequently, $E_2(Q)$ is a continuous function of the position of point Q in the domain $X \backslash \mathcal{A}$. For a formal proof of this property see [11]. A justification of the choice of Formula 1 is given in the Appendix.

The escape-time functions $E_1(Q)$ and $E_2(Q)$ are not defined inside the attractor $\mathcal{A}$, as one can find there infinite sequences of points remaining in $\mathcal{A}$ and therefore remaining in the circle $\mathcal{C}$. In order to make the definition of the escape time computationally effective, we evaluate the escape trajectories up to a predefined maximum length m. The escape-time functions, limited in this way, can be computed in the entire space X using the following formulae:

$$\underline{E}_1(Q,m) = \begin{cases} 0 \quad \text{if } Q \notin \mathcal{C} \text{ or } m = 0, \\ 1 + \max_{a \in V}\{\underline{E}_1(Q \circ \tilde{h}(a), m-1)\} \quad \text{otherwise.} \end{cases}$$

$$\underline{E}_2(Q,m) = \begin{cases} 0 \quad \text{if } Q \notin \mathcal{C} \text{ or } m = 0, \\ \max_{a \in V}\{res(Q,a)\} \quad \begin{array}{l} \text{if } Q \in \mathcal{C},\ m > 0, \text{and} \\ Q \circ \tilde{h}(a) \notin \mathcal{C} \text{ for all } a \in V, \end{array} \\ 1 + \max_{a \in V}\{\underline{E}_2(Q \circ \tilde{h}(a), m-1)\} \quad \text{otherwise.} \end{cases}$$

It is intuitively clear that $\underline{E}_1(Q,m) = E_1$ for all points Q with the escape time $E_1(Q)$ less than m, since the recursive formula evaluates step-by-step the same trajectories as its non-recursive counterpart. Similarly, $\underline{E}_2(Q,m) = E_2(Q)$ for all points Q such that $E_2(Q) < m$. Rigorous proofs of these equalities can be carried out by induction on m.

Figure 3 visualizes the dragon curve from Figure 1 using the continuous escape-time function $\underline{E}_2(Q,m)$. The inverse functions are:

$$\begin{aligned} F_1^{-1} &= r(-45) \circ s(\sqrt{2}), \\ F_2^{-1} &= t(0,-1) \circ r(-135) \circ s(\sqrt{2}). \end{aligned}$$

It is assumed that the circle $\mathcal{C}$ has radius R equal to 5, and the limit m is equal to 20. The values of function $\underline{E}_2(Q,m)$ are interpreted as a height field.

5. Language-restricted IFS's

A *language-restricted* iterated function system (LRIFS) is a quintuplet $\mathcal{I}_L = \langle X, \mathcal{F}, V, h, L\rangle$, where $X, \mathcal{F}, V$, and

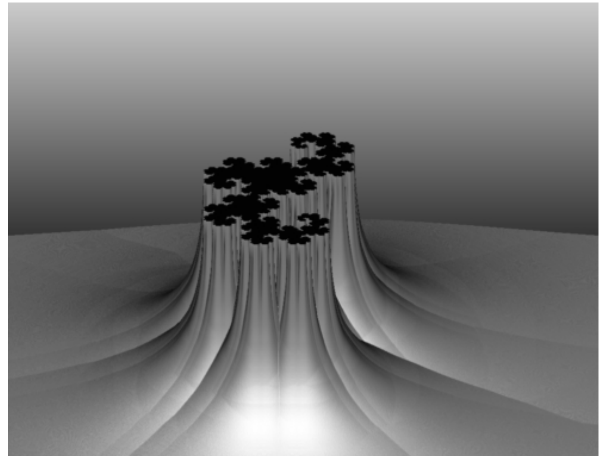


Figure 3: The dragon curve visualized using the escape-time method

h form an "ordinary" IFS, and $L \subset V^\star$ is a language over the alphabet V.

Consider a starting point P that belongs to the attractor $\mathcal{A}$ of the IFS $\mathcal{I}$, and let $\mathcal{A}_L(P)$ denote the closure of the image of P with respect to the transformations $h(L)$. The following inclusion holds:

$$\mathcal{A}_L(P) = \mathrm{cl}(P \circ h(L)) \subset \mathrm{cl}(P \circ h(V^*)) = \mathcal{A}.$$

Thus, the set $\mathcal{A}_L(P)$ generated by the LRIFS $\mathcal{I}_L$ with the starting point $P \in \mathcal{A}$ is a subset of the attractor $\mathcal{A}$.

For example, consider an LRIFS $\mathcal{F}_L = \langle X, \mathcal{F}, V, h, L\rangle$, where:

- the space X is the plane,
- the IFS $\mathcal{F}$ consists of four transformations:

$$\begin{aligned} F_1 &= s(0.5), \\ F_2 &= s(0.5) \circ t(0, 0.5), \\ F_3 &= s(0.5) \circ r(45) \circ t(0,1), \\ F_4 &= s(0.5) \circ r(-45) \circ t(0,1), \end{aligned}$$

- the alphabet V consists of four letters a, b, c, d,
- the homomorphism h is defined by:

$$h(a) = F_1, \quad h(b) = F_2, \quad h(c) = F_3, \quad h(d) = F_4,$$

- the language L consists of words in which no letter c or d is followed by an a or b. [1]

[1]Thus, L is defined by the regular expression:

$$L = (a \cup b)^*(c \cup d)^*.$$

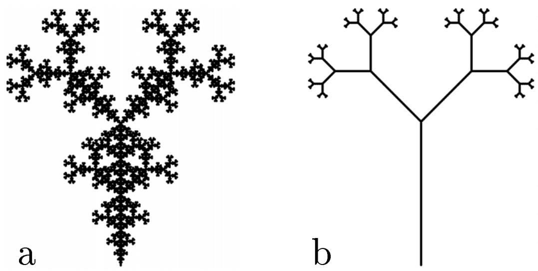


Figure 4: Attractor $\mathcal{A}$ and its subset $\mathcal{A}_L(P)$

Figure 4 compares the attractor $\mathcal{A}$ of the IFS $\mathcal{I}$ with the set $\mathcal{A}_L(P)$ generated by the LRIFS $\mathcal{I}_L$ using the starting point $P = (0,0)$. Clearly, the branching structure of Figure (b) is a subset of the original attractor (a).

In general, the set $\mathcal{A}_L(P)$ depends on the choice of the starting point P. Nevertheless, if the language L is prefix extensible (i.e., $vL \subset L$), the smallest set $\mathcal{A}_L$ does exist and can be found as $\mathrm{cl}(P_0 \circ h(L))$, where P_0 is the invariant point of the transformation $h(v)$. This results from the following inclusions, satisfied for any $P \in X$:

$$\begin{aligned}\mathrm{cl}(P_0 \circ h(L)) &= \mathrm{cl}((\lim_{n\to\infty} P \circ h(v^n)) \circ h(L)) \\ &= \lim_{n\to\infty} \mathrm{cl}(P \circ h(v^n L)) \subset \mathrm{cl}(P \circ h(L)).\end{aligned}$$

The limits are calculated in the space of all closed nonempty bounded subsets of the space X with the Hausdorff metric [16]. By analogy with the "ordinary" IFS's, we call $\mathcal{A}_L$ the attractor of the LRIFS $\mathcal{I}_L$.

6. The escape-time method for LRIFS's

While extending the escape-time method to LRIFS's, we consider mirror images of words and use the following lemma.

Lemma. Consider IFS $\mathcal{I} = \langle X, \mathcal{F}, V, h\rangle$, and let all functions $F = h(a) \in \mathcal{F}$ be invertible. Then for any word $w \in V^\star$, the equality $\tilde{h}(w) = (h(w^R))^{-1}$ holds.

Proof. The set $\mathcal{F}$ forms a group of transformations with the operations of function composition and inversion, thus $(F_i \circ F_j)^{-1} = F_j^{-1} \circ F_i^{-1}$ for any $F_i, F_j \in \mathcal{F}$. Consequently, the following equalities are true for any word $w = a_1 a_2 \dots a_n \in V^\star$:

$$\begin{aligned}\tilde{h}(w) &= \tilde{h}(a_1 a_2 \dots a_n) \\ &= \tilde{h}(a_1) \circ \tilde{h}(a_2) \circ \dots \circ \tilde{h}(a_n) \\ &= (h(a_1))^{-1} \circ (h(a_2))^{-1} \circ \dots \circ (h(a_n))^{-1} \\ &= (h(a_n) \circ \dots \circ h(a_2) \circ h(a_1))^{-1} \\ &= (h(a_n \dots a_2 a_1))^{-1} = (h(w^R))^{-1}. \ \square\end{aligned}$$

The escape-time method for LRIFS's is based on the following extension of Theorem 1 from Section 4:

Theorem 2. Consider an LRIFS $\mathcal{I}_L = \langle X, \mathcal{F}, V, h, L\rangle$, and assume that the language L is prefix extensible, $vL \subset L$. Denote by $\mathcal{A}$ the attractor of the IFS $\langle X, \mathcal{F}, V, h\rangle$, and by $\mathcal{A}_L$ the attractor of $\mathcal{I}_L$. (a) If a starting point $Q \in X$ belongs to the attractor $\mathcal{A}_L$, then for any $n \geq 0$ there exists a word w in the prefix closure $\mathcal{P}(L^R)$ such that $\mathrm{length}(w) \geq n$ and $\mathrm{Tr}(Q, w) \subset \mathcal{A}$. (b) If the point Q does not belong to $\mathcal{A}_L$, all trajectories $\mathrm{Tr}(Q, w)$ with $w \in \mathcal{P}(L^R)$ diverge to infinity as $\mathrm{length}(w) \to \infty$.

Proof. (a) Let P_0 denote the invariant point of the transformation $h(v)$. According to the definition of the attractor $\mathcal{A}_L$, there exists a word $y \in L$ such that $P_0 \circ h(y) = Q$.[2] Since $P_0 = P_0 \circ h(v)$, the equality $P_0 \circ h(v^i y) = Q$ holds for any $i \geq 0$. Let i satisfy the inequality $\mathrm{length}(v^i y) \geq n$, and $w = (v^i y)^R$. The word w belongs to L^R and henceforth to $\mathcal{P}(L^R)$, has length greater than or equal to n, and maps point Q to the point $P_0 \in \mathcal{A}_L$:

$$Q \circ \tilde{h}(w) = Q \circ (h(v^i y))^{-1} = P_0 \in \mathcal{A}_L.$$

In order to show that the entire trajectory $\mathrm{Tr}(Q, w)$ is included in the attractor $\mathcal{A}$, let us consider an arbitrary partition of the word w into a prefix x_1 and a suffix x_2; thus $x_1 x_2 = w$. From the equality

$$Q \circ \tilde{h}(w) = Q \circ \tilde{h}(x_1) \circ \tilde{h}(x_2) = P_0$$

it follows that

$$\begin{aligned}Q \circ \tilde{h}(x_1) &= P_0 \circ (\tilde{h}(x_2))^{-1} \\ &= P_0 \circ h(x_2^R) \in P_0 \circ h(V^\star) \subset \mathcal{A}.\end{aligned}$$

Since this argument holds for any $x \prec w$, we obtain:

$$\mathrm{Tr}(Q, w) = \{Q \circ \tilde{h}(x) : x \prec w\} \subset \mathcal{A}.$$

(b) Let $\mathcal{C}$ be an arbitrary circle enclosing the attractor $\mathcal{A}$, and R denote the radius of $\mathcal{C}$. We have to prove that if $Q \notin \mathcal{A}_L$, there exists a number $n \geq 0$ such that for any word $w \in \mathcal{P}(L^R)$ of length greater then or equal to n, the escape trajectory $\mathrm{Tr}(Q, w)$ is not entirely included in $\mathcal{C}$. Let D denote the distance between point Q and the attractor $\mathcal{A}_L$, and r_{max} be the largest Lipschitz constant found among the transformations $F \in \mathcal{F}$. Since $D > 0$ and $r_{max} < 1$, there exists a number $n \geq 0$ such that $2Rr_{max}^n < D$. Consider an arbitrary word $w \in \mathcal{P}(L^R)$ with $\mathrm{length}(w) \geq n$, and let $wy \in L^R$, or

[2] Strictly speaking, there exists a word $y \in L$ such that $P_0 \circ h(y)$ is arbitrarily close to Q.

$y^R w^R \in L$. Then there exist points $P_0, P \in \mathcal{A}_L$ such that $P_0 \circ h(y^R w^R) = P$. We decompose the last equality by introducing an intermediate point P':

$$P_0 \circ h(y^R) = P' \quad \text{and} \quad P' \circ h(w^R) = P.$$

It follows that

$$P \circ \tilde{h}(w) = P' = P_0 \circ h(y^R) \in P_0 \circ h(V^\star) \subset \mathcal{A}.$$

The distance between points P and Q is at least D, and the Lipschitz constant of the composite transformation $\tilde{h}(w) = (h(w^R))^{-1}$ is at least r_{max}^{-n}, thus

$$\begin{aligned} d(Q \circ \tilde{h}(w), P \circ \tilde{h}(w)) &\geq d(Q, P) r_{max}^{-n} \\ &\geq D r_{max}^{-n} > 2R. \end{aligned}$$

Since $P \circ \tilde{h}(w) = P' \in \mathcal{A} \subset \mathcal{C}$, and the distance of $Q \circ \tilde{h}(w)$ from P' is greater than the diameter of $\mathcal{C}$, the point $Q \circ \tilde{h}(w)$ must lie outside of $\mathcal{C}$, or

$$\mathrm{Tr}(Q, w) \not\subset \mathcal{C}. \quad \Box$$

Theorem 2 reveals an analogy between the escape trajectories of an LRIFS and an ordinary IFS. In both cases we find infinitely long trajectories confined to $\mathcal{A}$ if the starting point Q belongs to the attractor — respectively $\mathcal{A}_L$ or $\mathcal{A}$. For a point Q outside an attractor, all trajectories diverge to infinity as their length increases. However, in the case of an ordinary IFS we consider escape trajectories with respect to all possible words $w \in V^\star$, while in the case of an LRIFS the words w are confined to the prefix closure $K = \mathcal{P}(L^R)$.

As a result of these observations, we can extend the escape-time formulae from Section 4 to LRIFS's as follows:

$$E_{L1}(Q) = \max_{w \in K} \{\mathrm{length}(w) : \mathrm{Tr}(Q, w) \subset \mathcal{C}\},$$

$$E_{L2}(Q) = \max_{wa \in K} \{\mathrm{length}(w) + \mathrm{res}(Q \circ \tilde{h}(w), a) : \mathrm{Tr}(Q, w) \subset \mathcal{C}\}.$$

In the recursive counterparts of these functions, the key issue is the selection of mappings $\tilde{h}(a)$ that can be applied in each step. We use the derivatives of the language K to find the appropriate letters a at each level of recursion. As previously, m limits the recursion depth.

$$\underline{E}_{L1}(Q, K, m) = \begin{cases} 0 & \text{if } Q \notin \mathcal{C} \text{ or } m = 0, \\ 1 + \max_{a \in K} \{\underline{E}_{L1}(Q \circ \tilde{h}(a), K//a, m-1)\} & \text{otherwise.} \end{cases}$$

$$\underline{E}_{L2}(Q, K, m) = \begin{cases} 0 & \text{if } Q \notin \mathcal{C} \text{ or } m = 0, \\ \max_{a \in K} \{res(Q, a)\} & \text{if } Q \in \mathcal{C},\ m > 0, \text{and } Q \circ \tilde{h}(a) \notin \mathcal{C} \text{ for all } a \in K, \\ 1 + \max_{a \in K} \{\underline{E}_{L2}(Q \circ \tilde{h}(a), K//a, m-1)\} & \text{otherwise.} \end{cases}$$

These formulae can be used for any language $K = \mathcal{P}(L^R)$, provided that L has the prefix property, as assumed in Theorem 2. The required operations on languages are particularly simple if L is regular. It can be then specified using a finite-state automaton, which reduces operations on infinite languages to the operations on their finite representations. Details are given in the following section.

7. The application of finite automata

We start by recalling the necessary notions of the theory of finite automata. For the original presentation see [19].

A *nondeterministic finite-state (Rabin-Scott) automaton* is a quintuplet:

$$\mathcal{M} = < V, S, s_0, T, I >,$$

where:

- V is an alphabet,
- S is a finite set of states,
- $s_0 \in S$ is a distinguished element of S, called the initial state,
- $T \subset S$ is a distinguished subset of S, called the set of final states,
- $I \subset V \times S \times S$ is a state transition relation.

We often write $(a, s_i) \to s_k$ instead of $(a, s_i, s_k) \in I$.

Finite state automata are commonly represented as directed graphs, with the nodes corresponding to states, and arcs representing transitions. The initial state is pointed to by a short arrow. The final states are distinguished by double circles.

A word $w = a_1 a_2 \ldots a_n \in V^\star$ is *accepted* by the automaton $\mathcal{M}$ if there exists a sequence of states $s_0, s_1, s_2, \ldots, s_{n-1} \in S$ and $s_n \in T$ such that:

$$(a_1, s_0) \to s_1, \quad (a_2, s_1) \to s_2, \quad \cdots, \quad (a_n, s_{n-1}) \to s_n.$$

Thus, w is accepted by $\mathcal{M}$ if there exists a directed path in the graph of $\mathcal{M}$ starting in the initial state s_0, ending

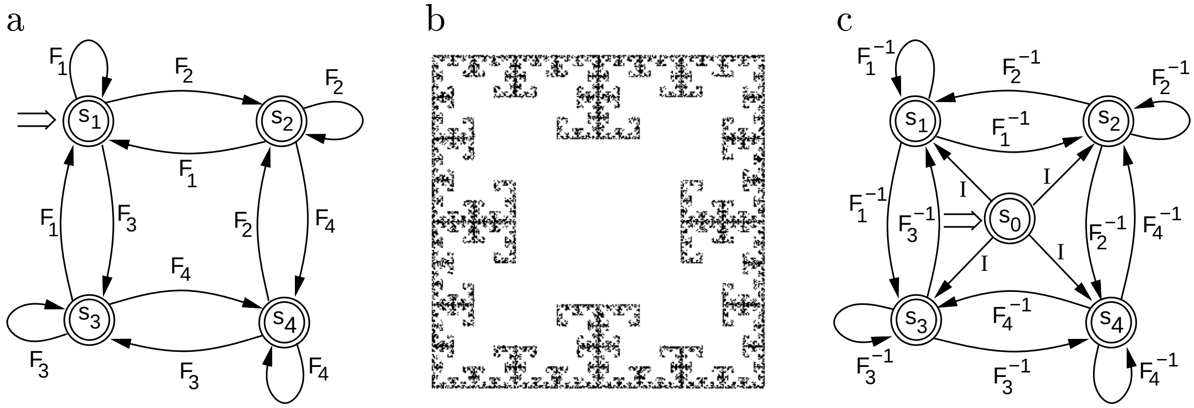


Figure 5: (a) The automaton $\mathcal{M}_1$ defining the language L_1, (b) the attractor of the LRIFS $\mathcal{I}_1$, and (c) the automaton $\mathcal{M}_1^{\mathcal{RP}}$ defining the language $K_1 = \mathcal{P}(L_1^R)$

in some final state s_n, and labeled with the consecutive letters of w. The set of all words accepted by an automaton $\mathcal{M}$ is called the *language accepted by* $\mathcal{M}$, and denoted by $L(\mathcal{M})$.

It is known that the mirror image of the language $L(\mathcal{M})$ is accepted by the automaton

$$\mathcal{M}^{\mathcal{R}} = \langle V, S \cup \{s_0'\}, s_0', \{s_0\}, I^R \rangle,$$

where $I^R =$

$$\{(\epsilon, s_0', s_k) : s_k \in T\} \cup \{(a, s_j, s_i) : (a, s_i, s_j) \in I\}.$$

Thus, the automaton $\mathcal{M}^{\mathcal{R}}$ is obtained from $\mathcal{M}$ by:

- creating a new initial state $s_0' \notin S$,
- creating transitions labeled ϵ from s_0' to all final states of $\mathcal{M}$,
- reversing the directions of all other transitions,
- making s_0 the unique final state of $\mathcal{M}^{\mathcal{R}}$.

Given an automaton $\mathcal{M}$ defining a language L, the prefix closure $\mathcal{P}(L^R)$ is accepted by the automaton $\mathcal{M}^{\mathcal{RP}}$ obtained from $\mathcal{M}^{\mathcal{R}}$ by making all its states final.

Consider an LRIFS $\mathcal{I} = \langle X, \mathcal{F}, V, h, L \rangle$, where L is accepted by a given finite automaton $\mathcal{M}$. Using the method given above, we can construct the automaton $\mathcal{M}^{\mathcal{RP}} = \langle V, S, s_0, S, I \rangle$ that accepts the language $K = \mathcal{P}(L^R)$. A word w belongs to K if and only if there exists a path in $\mathcal{M}^{\mathcal{RP}}$ starting in s_0 and labeled with the consecutive letters of w. Thus, the recursive computation of the derivatives of K, needed to evaluate functions E_{L1} and E_{L2}, can be replaced by the recursive construction of paths in $\mathcal{M}^{\mathcal{RP}}$, starting is s_0. This leads to the following recursive definitions:

$$\underline{E}_{M1}(Q, s_i, m) = \begin{cases} 0 \quad \text{if } Q \notin \mathcal{C} \text{ or } m = 0, \\ 1 + \max\limits_{(a, s_i, s_j) \in I} \{\underline{E}_{M1}(Q \circ \tilde{h}(a), s_j, m-1)\} \\ \hfill \text{otherwise.} \end{cases}$$

$$\underline{E}_{M2}(Q, s_i, m) = \begin{cases} 0 \quad \text{if } Q \notin \mathcal{C} \text{ or } m = 0, \\ \max\limits_{(a, s_i, s_j) \in I} \{res(Q, a)\} \quad \begin{array}{l} \text{if } Q \in \mathcal{C},\ m > 0, \text{and} \\ Q \circ \tilde{h}(a) \notin \mathcal{C} \\ \text{for all } (a, s_i, s_j) \in I, \end{array} \\ 1 + \max\limits_{(a, s_i, s_j) \in I} \{\underline{E}_{M2}(Q \circ \tilde{h}(a), s_j, m-1)\} \\ \hfill \text{otherwise.} \end{cases}$$

The evaluation of functions $\underline{E}_{M1}$ and $\underline{E}_{M2}$ starts with $s_i = s_0$. The equivalence of the formulae for $\underline{E}_{L1}$ and $\underline{E}_{M1}$, as well as $\underline{E}_{L2}$ and $\underline{E}_{M2}$, can be proved by induction on the maximum path length in $\mathcal{M}^{\mathcal{RP}}$.

Example 1. The following LRIFS $\mathcal{I}_1 = \langle X, \mathcal{F}_1, V_1, h_1, L_1 \rangle$ was described by Berstel and Abdallah [6]. It is assumed that:

- X is the plane,
- $\mathcal{F}_1$ consists of four transformations:

$$\begin{aligned} F_1 &= s(0.5) \circ t(0.0, 0.5), \\ F_2 &= s(0.5) \circ t(0.5, 0.5), \\ F_3 &= s(0.5), \\ F_4 &= s(0.5) \circ t(0.5, 0.0), \end{aligned}$$

- $V_1 = \{F_1, F_2, F_3, F_4\}$,
- $h_1(F_i) = F_i$ for $i = 1, 2, 3, 4$.

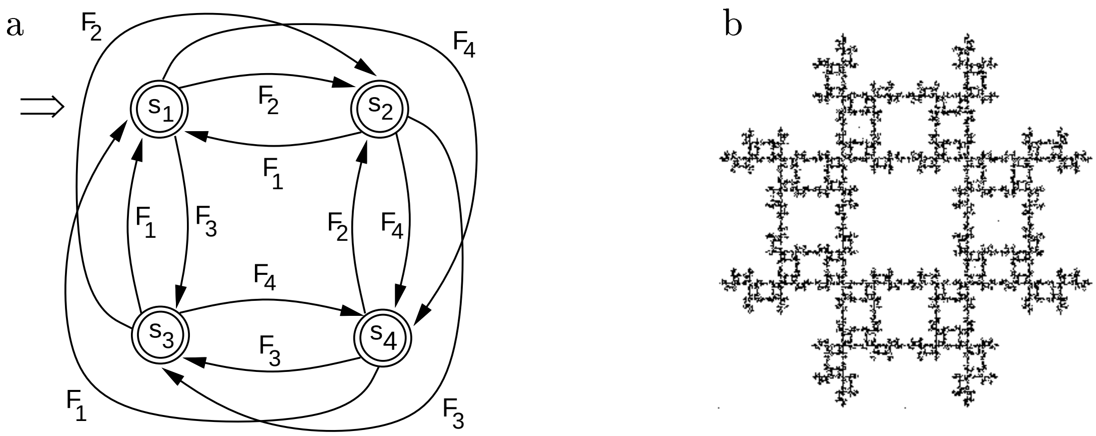


Figure 6: (a) The automaton $\mathcal{M}_2$ defining the language L_2, and (b) the attractor of the LRIFS $\mathcal{I}_2$

The language L_1 is defined using the finite automaton $\mathcal{M}_1$ shown in Figure 5a, and the corresponding attractor is given in Figure 5b. The automaton defining the language $K_1 = \mathcal{P}(L_1^R)$ is shown in Figure 5c, with the transitions labeled using the inverse transformations of $\mathcal{F}_1$. The label I indicates the identity transformation, associated with the ϵ-transitions of $\mathcal{M}_1^{\mathcal{RP}}$. Plate 1 (left) visualizes the escape time function computed with a recursion depth limit $m = 20$, using a bounding circle $\mathcal{C}$ with radius $R = 5$. The escape time values are interpreted as indices to a color map, arbitrarily divided into several ramps. Plate 1 (right) presents the same function as a height field.

Example 2. The LRIFS $\mathcal{I}_2$ considered in this example was described by Vrscay [20]. It uses the same set of transformations $\mathcal{F}$ and the labeling function h as $\mathcal{I}_1$, but the language L_2 is different. The automaton $\mathcal{M}_2$ defining L_2 and the resulting attractor are shown in Figure 6. The escape time function is presented in Plate 2.

Example 3. The LRIFS $\mathcal{I}_3$, taken from [16], describes a leaf-like structure with the alternating and opposite branches. The set of transformations is specified below:

$$
\begin{aligned}
F_1 &= s(0.5) \circ t(-0.002, 0)\\
F_2 &= s(0.5) \circ t(0.002, 0)\\
F_3 &= s(0.5) \circ t(-0.002, 0.13)\\
F_4 &= s(0.5) \circ t(0.002, 0.13)\\
F_5 &= s(0.42) \circ r(45)\\
F_6 &= s(0.2) \circ r(90) \circ t(-0.05, 0.05)\\
F_7 &= s(0.2) \circ t(-0.05, 0.05)\\
F_8 &= t(0.3, -0.3) \circ s(0.74) \circ t(-0.3, 0.3)\\
F_9 &= s(0.37) \circ r(-45) \circ t(0, 0.14)\\
F_{10} &= s(0.172) \circ r(-90) \circ t(0.05, 0.19)\\
F_{11} &= s(0.172) \circ t(0.05, 0.19)\\
F_{12} &= t(-0.265, -0.405) \circ s(0.74) \circ t(0.265, 0.405)\\
F_{13} &= t(0, -1) \circ s(0.74) \circ t(0, 1)
\end{aligned}
$$

The automaton $\mathcal{M}_3$ defining L_3 and the corresponding attractor are shown in Figure 7. The escape time function is visualized in Plate 3.

8. Conclusions

This paper presents methods for computing the escape-time functions of language-restricted iterated function systems. The LRIFS's generalize the ordinary IFS's by imposing restrictions on the applicable sequences of transformations. The escape-time functions can be computed for any set of sequences constituting a prefix-extensible formal language L. The computation of the escape time involves finding the mirror image L^R, determining the prefix language $K = P(L^R)$, and calculating its derivatives. These operations can be performed in a simple way if L is regular, using a specification of L by a finite automaton. All examples considered in this paper refer to this case. It is an open problem whether non-regular languages can yield other attractors and visualizations.

One could raise a question, whether this paper applies computer graphics to visualize an important mathematical concept, or whether it merely employs mathematics to create images for the sake of their visual appeal. Our motivation falls in both areas — we wanted to extend the mathematical concept of escape-time functions to LRIFS's, realizing that it is primarily used for image synthesis. In addition, we found that the well-established theory of automata and formal languages had unexpected applications in computer graphics.

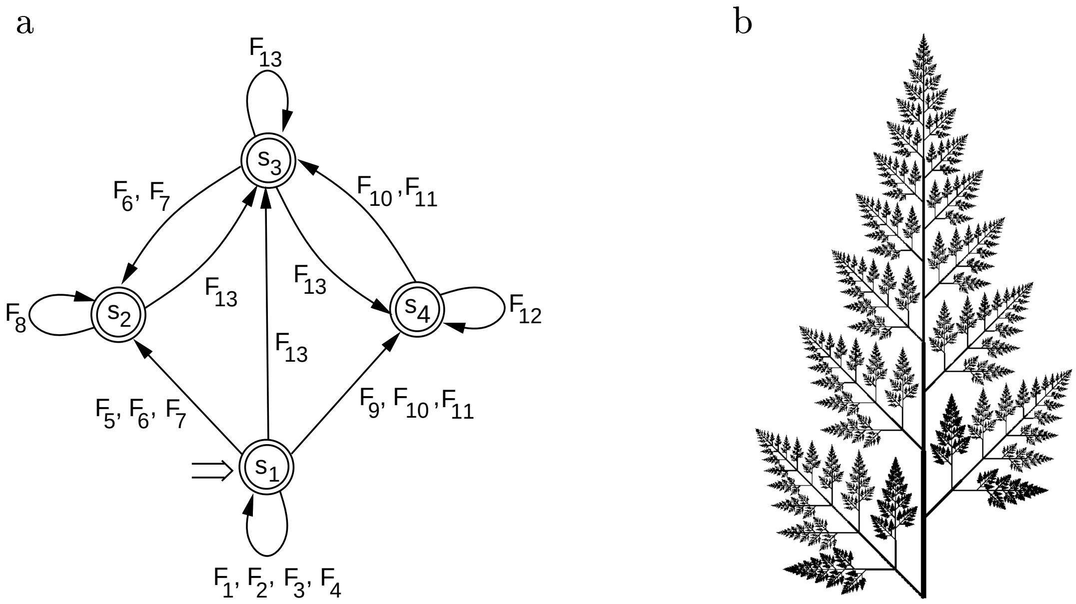


Figure 7: (a) The automaton $\mathcal{M}_3$ defining the language L_3, and (b) the attractor of the LRIFS $\mathcal{I}_3$

Appendix: Justification of the logarithmic formula for $\mathrm{res}(Z, a)$.

The continuity of the escape function can also be maintained by residual terms other than that given by Formula 1, for example:

$$\mathrm{res}'(Z,a) = \begin{cases} \dfrac{R - \|Z\|}{\|Z \circ \tilde{h}(a)\| - \|Z\|} & \text{if } Z \in \mathcal{C} \text{ and } Z \circ \tilde{h}(a) \notin C, \\ 0 & \text{otherwise.} \end{cases}$$

In order to explain the advantages of Formula 1, let us consider an IFS consisting of a single complex function $F(z) = z/c$. By definition, $F(z)$ is a contraction, thus $|c| > 1$. Given a circle $\mathcal{C}$ with radius R and center O in the origin of the coordinate system, the integer-valued escape-time function $E_1(z)$ is equal to:

$$E_1(z) = \max\{n \in \mathcal{N} : \|zc^n\| \le R\}.$$

The symbol $\mathcal{N}$ represents the set of natural numbers (including zero), and the module of a complex number is identified with its norm, $|zc^n| = \|zc^n\|$. A continuous (and infinitely differentiable) extension of function $E_1(z)$ is:

$$E_2(z) = \max\{u \in \mathcal{R}^+ : \|zc^u\| \le R\},$$

where $\mathcal{R}^+$ is the set of nonnegative real numbers. Obviously, the value $E_2(z)$ satisfies the equation:

$$\|zc^{E_2(z)}\| = R.$$

Consider point $Z = zc^{E_1(z)}$. By representing $E_2(z)$ as a sum $E_1(z) + \mathrm{res}(Z)$, we obtain[3]:

$$\|zc^{E_1(z)+\mathrm{res}(Z)}\| = \|Zc^{\mathrm{res}(Z)}\| = R.$$

Note that $c = Zc/Z = F^{-1}(Z)/Z$, and take logarithms of both sides of the previous equation:

$$\log\|Z\| + \mathrm{res}(Z)(\log\|F^{-1}(Z)\| - \log\|Z\|) = \log R.$$

Consequently,

$$\mathrm{res}(Z) = \frac{\log R - \log\|Z\|}{\log\|F^{-1}(Z)\| - \log\|Z\|}.$$

Although the above reasoning applies to a particular IFS, it justifies the use of Function 1 also in other cases. In general, the distance between the origin of circle $\mathcal{C}$ and consecutive points in an escape trajectory tends to grow exponentially for large distance values. Consequently, Formula 1 minimizes first-order discontinuities in the escape-time function, yielding visually pleasing graphical representations.

Acknowledgements

We would like to thank Dr. Dietmar Saupe for the initial discussions of the concepts presented in this paper, the detailed comments of the manuscript, and for providing us with the program for rendering height fields.

[3]There is no need for specifying the second argument to the function res, as the IFS under consideration consists of a single transformation.

This research was sponsored by an operating grant and a graduate scholarship from the Natural Sciences and Engineering Research Council of Canada. The images were generated using facilities of the University of Calgary and the University of Regina.

References

[1] Ch. Bandt. Self-similar sets III. Constructions with sofic systems. *Monatsh. Math*, 108:89–102, 1989.

[2] M. F. Barnsley. *Fractals Everywhere*. Academic Press, San Diego, 1988.

[3] M. F. Barnsley, J. H. Elton, and D. P. Hardin. Recurrent iterated function systems. *Constructive Approximation*, 5:3–31, 1989.

[4] M. F. Barnsley, A. Jacquin, F. Malassenet, L. Reuter, and A. D. Sloan. Harnessing chaos for image synthesis. Proceedings of SIGGRAPH '88 (Atlanta, Georgia, August 1-5, 1988). In *Computer Graphics*, 22, 4 (July 1988), pages 131–140, ACM SIGGRAPH, New York, 1988.

[5] M. A. Berger. Images generated by orbits of 2-D Markov chains. *Chance*, 2(2):18–28, 1989.

[6] J. Berstel and A. Nait Abdallah. Tétrarbres engendreés par des automates finis. Technical Report 89-7, Laboratoire Informatique Théorique et Programmation, Université P. et M. Curie, 1989.

[7] K. Culik II and S. Dube. Affine automata and related techniques for generation of complex images. Manuscript, University of South Carolina in Columbia.

[8] K. Culik II and S. Dube. Balancing order and chaos in image generation. Manuscript, University of South Carolina in Columbia.

[9] J.C. Hart. The object instancing paradigm for linear fractal modeling. In *Graphics Interface '92*, pages 224–231. CIPS, 1992.

[10] J.C. Hart and T.A. DeFanti. Efficient antialiased rendering of 3-d linear fractals. Proceedings of SIGGRAPH '91 (Las Vegas, Nevada, July 28 – August 2, 1991). In *Computer Graphics*, 25, 4 (July 1991), pages 91–100, ACM SIGGRAPH, New York, 1991.

[11] D. Hepting, P. Prusinkiewicz, and D. Saupe. Rendering methods for iterated function systems. In *Fractals in the Fundamental and Applied Sciences*, pages 183–224. Elsevier, 1991.

[12] J. E. Hutchinson. Fractals and self-similarity. *Indiana University Journal of Mathematics*, 30(5):713–747, 1981.

[13] B. B. Mandelbrot. *The fractal geometry of nature*. W. H. Freeman, San Francisco, 1982.

[14] H.-O. Peitgen and P. H. Richter, editors. *The Beauty of Fractals*. Springer-Verlag, Heidelberg, 1986.

[15] H.-O. Peitgen and D. Saupe, editors. *The Science of Fractal Images*. Springer-Verlag, New York, 1988.

[16] P. Prusinkiewicz and M. Hammel. Automata, languages, and iterated function systems. In J. C. Hart and F. K. Musgrave, editors, *Fractal Modeling in 3D Computer Graphics and Imagery*, pages 115–143. ACM SIGGRAPH, 1991. Course Notes C14.

[17] P. Prusinkiewicz and A. Lindenmayer. *The algorithmic beauty of plants*. Springer-Verlag, New York, 1990. With J. S. Hanan, F. D. Fracchia, D. R. Fowler, M. J. M. de Boer, and L. Mercer.

[18] P. Prusinkiewicz and G. Sandness. Koch curves as attractors and repellers. *IEEE Computer Graphics and Applications*, 8(6):26–40, November 1988.

[19] M. O. Rabin and D. Scott. Finite automata and their decision problems. *IBM J. Res. Develop.*, 3:114–125, 1959.

[20] E. R. Vrscay. Iterated function systems: Theory, applications and the inverse problem. In *Proceedings of the NATO Advanced Study Institute on Fractal Geometry held in Montreal, July 1989*. Kluwer Academic Pulishers, 1990.

[21] T. E. Womack. Linear and Markov iterated function systems in fractal geometry. Master's thesis, Virginia Polytechnic Institute and State University, Blacksburg, Virginia, 1989.

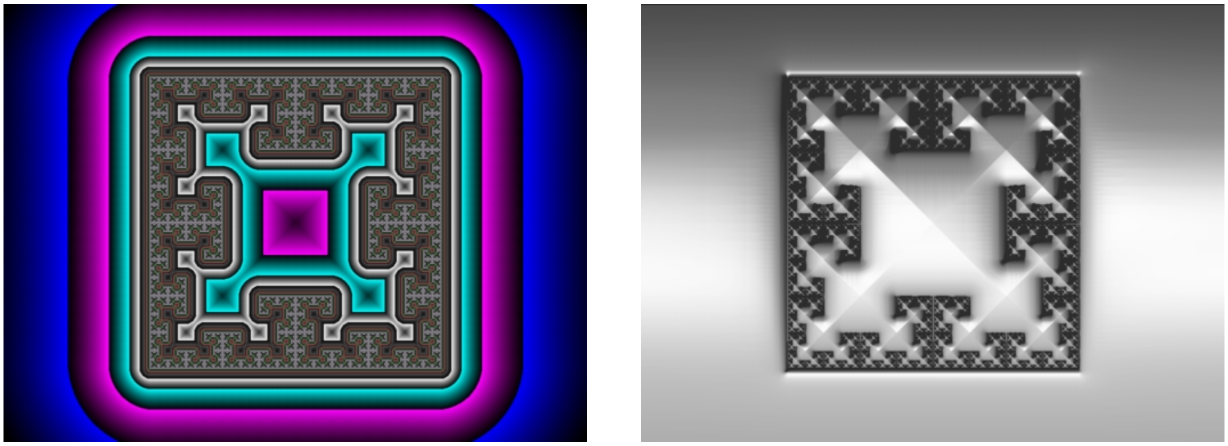


Plate 1: The escape time function for the attractor of the LRIFS $\mathcal{I}_1$, shown using a color map and as a height field

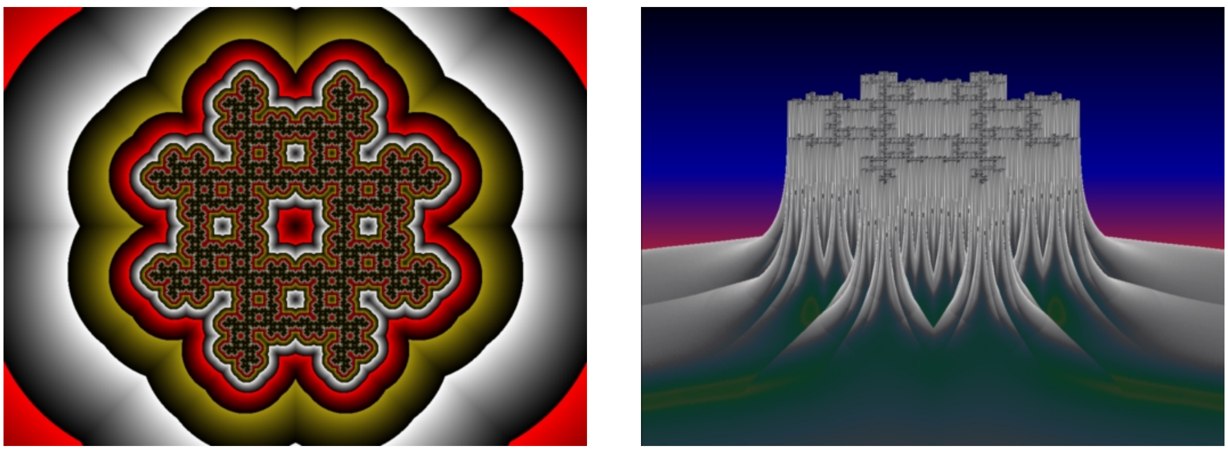


Plate 2: The escape time function for the attractor of the LRIFS $\mathcal{I}_2$, shown using a color map and as a height field

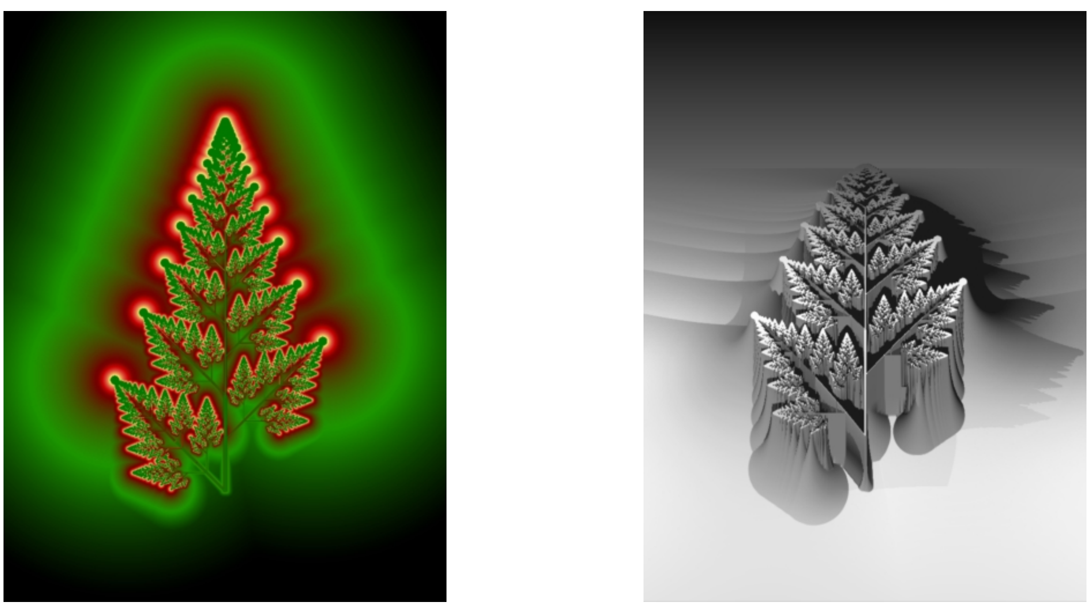


Plate 3: The escape time function for the attractor of the LRIFS $\mathcal{I}_3$, shown using a color map and as a height field