

Manipulation of Non-Linear IFS attractors using Genetic Programming

Frédéric Raynal, Evelyne Lutton, Pierre Collet, Marc Schoenauer

► **To cite this version:**

Frédéric Raynal, Evelyne Lutton, Pierre Collet, Marc Schoenauer. Manipulation of Non-Linear IFS attractors using Genetic Programming. Congress on Evolutionary Computation, Jul 1999, Washington DC, USA. inria-00000878

HAL Id: inria-00000878

<https://hal.inria.fr/inria-00000878>

Submitted on 29 Nov 2005

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Manipulation of Non-Linear IFS Attractors Using Genetic Programming

Frédéric Raynal, Evelyne Lutton, Pierre Collet,

INRIA - Rocquencourt, B.P. 105, 78153 LE CHESNAY Cedex, France

Frederic.Raynal@inria.fr, Evelyne.Lutton@inria.fr, Collet@cheverny.u-bourgogne.fr,

<http://www-rocq.inria.fr/fractales/>

Marc Schoenauer

EAAX-CMAPx, Ecole Polytechnique, 91128 Palaiseau Cedex, France

Marc.Schoenauer@polytechnique.fr

<http://www.eeaax.polytechnique.fr>

Abstract-

Non-linear Iterated Function Systems (IFSs) are very powerful mathematical objects related to fractal theory, that can be used in order to generate (or model) very irregular shapes. We investigate in this paper how Genetic Programming techniques can be efficiently exploited in order to generate randomly or interactively artistic “fractal” 2D shapes. Two applications are presented for different types of non-linear IFSs:

- interactive generation of Mixed IFSs attractors using a classical GP scheme,
- random generation of Polar IFSs attractors based on an “individual” approach of GP.

1 Introduction

IFSs (Iterated Functions System) theory is an important topic in fractals, and provides powerful tools to investigate fractal sets. The action of systems of contractive maps to produce fractal sets has been considered by many authors (see for example [11, 2, 3, 7, 10]), and most fractal image compression techniques are based on IFSs [4, 12, 8].

Non-affine IFSs —which we call “Mixed IFSs,” in order to emphasize¹ the fact that they are not anymore restricted to a composition of affine functions— provide an interesting variety of shapes, whose practical interest might be large. They can be represented as trees in a GP, see section 3. The main problem to be addressed with such Mixed IFSs is the contractivity constraint: The “interesting” IFSs, i.e. that admit an attractor, are composed of contractive functions. We will see in section 4 how it has been solved in the framework of an application of interactive image generation, in the “Karl Sims manner,” based on a GP algorithm interfaced with a JAVA applet.

We propose in section 5 to consider a subclass of non-linear IFSs which, by construction, admit a unique fixed point. A polar representation of these IFSs is very convenient:

¹In the literature related to IFSs, the great majority of papers consider affine IFS, so that usually when “IFS” are mentioned, they are often implicitly supposed to be affine.

- each function is represented in polar coordinates with respect to a central point. A simple constraint on the ρ coordinate insures the convergence towards a central point, which happens to be the fixed point of the function, see section 5 — we use the term “Polar IFS” to designate an IFS built with these functions,
- polar IFSs can still be represented as trees just as “Mixed IFSs,”
- the handling of contractance constraints is simpler than with mixed IFSs thanks to the constraint on ρ ,
- the set of Polar IFSs contains a much higher percentage of “contractive” IFSs than a set of Mixed IFSs. Hence, polar IFSs provide a more efficient (less sparse) search space to the optimisation algorithm than Mixed IFSs, see [6].

We finally present in section 7 a more “economic” approach of the GP technique: it is based on the co-evolving capacities of populations in evolutionary algorithms. The solution to the problem is then represented by the whole population (or a subset of the whole population) and not any more by a single individual (just as “classifier systems” approaches, see [9]). Although more complex to implement (mainly with respect to the fitness computations), this approach allows, in some particular cases, to build more efficient algorithms.

2 IFS theory

An IFS (Iterated Function System) $U = \{F, (w_n)_{n=1, \dots, N}\}$ is a collection of N functions defined on a complete metric space (F, d) .

Let W be the operator defined on the space of subsets of F :

$$\forall K \subset F, W(K) = \bigcup_{n \in \{1, \dots, N\}} w_n(K)$$

Then, if the w_n functions are contractive (the IFS is then often called a *hyperbolic* or *contractive* IFS), there exists a unique set A such that:

$$W(A) = A$$

A is called the **attractor** of the IFS.

Recall:

A mapping $w : F \rightarrow F$, from a metric space (F, d) into itself, is called **contractive** if there exists a positive real number $s < 1$ such that:

$$d(w(x), w(y)) \leq s \cdot d(x, y) \quad \forall x, y \in F$$

The uniqueness of a hyperbolic attractor is a result of the Contractive Mapping Fixed Point Theorem for W , which is contractive according to the HAUSDORFF distance:

- HAUSDORFF *distance*:

$$d_H(A, B) = \max \left(\max_{x \in A} \left(\min_{y \in B} d(x, y) \right), \max_{y \in B} \left(\min_{x \in A} d(x, y) \right) \right)$$

- *Contractive Mapping Fixed Point Theorem*:

if (F, d) is a complete metric space, and $W : F \rightarrow F$ is a contractive transformation, then W has a unique fixed point.

From a computational viewpoint, an attractor can be generated according to two techniques:

- **Stochastic method (toss-coin):**

Let x_0 be the fixed point of one of the w_i functions. We build the point sequence x_n as follows: $x_{n+1} = w_i(x_n)$, i being randomly chosen in $\{1..N\}$. Then $\bigcup_n x_n$ is an approximation of the real attractor of \mathcal{U} . The larger n , the more precise the approximation.

- **Deterministic method:**

From any initial set S_0 , we build the sets sequence $\{S_n\}$:

$$S_{n+1} = W(S_n) = \bigcup_i w_i(S_n)$$

When n tends to ∞ , S_n is an approximation of the real attractor of \mathcal{U} .

3 Mixed IFS

In the case of affine IFS, each contractive map w_i of \mathcal{U} is represented as:

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

When the w_i are not anymore restricted to affine functions, we call the corresponding IFS **Mixed IFS**. The first point we have to address is the one of finding an adequate representation of these mixed IFSs. A natural one is to represent them as trees.

The w_i functions are built from a set of basic functions, a set of variables (x and y), and a set of constants. In examples which follow, the constants belong to $[0, 1]$, and the basic functions set is:

$$\begin{aligned} +, -, \times, div(x, y) &= \frac{x}{0.0001+|y|}, \\ \cos, \sin, \\ root(x) &= \sqrt{|x|}, \log a(x) = \log(1 + |x|) \end{aligned}$$

The trees of the w_i functions are then gathered to build the main tree which represents the IFS \mathcal{U} . This is a very simple structure which allows to code IFSs with different numbers and different types of functions.

The set of possible IFSs depends on the choice of the basic functions set and constants set. A difficult problem for mixed IFSs is to verify that the w_i are contractive, in order to select *hyperbolic* IFSs. On the contrary to affine IFSs, this verification is not straightforward, and is in fact computationally intractable. We thus propose to use some heuristics that reject strongly non-contractive functions. The simplest way to do that (see [14] for details) is to verify the contractivity on some sample points, for example during the estimation process of the fixed points x_i^* of each w_i .

As we have chosen to generate IFSs whose attractors are in the $\square = [0, 1] \times [0, 1]$ domain, we build for each w_i two sequences of points $x_{n+1} = w_i(x_n)$, starting from two opposite corners of \square , that converge on x_i^* if it exists. If w_i is contractive such sequences rapidly converge in \square , and in the same time a contractive factor s_i can be estimated.

4 Genetic Programming for the interactive generation of Mixed IFSs attractors

A Genetic Programming algorithm may be used in an interactive way in order to investigate the space of mixed IFSs attractors. We present below an implementation which interactively generates images of attractors. The GP algorithm is used as an exploration tool in an image space: the implicit optimisation function is the “user’s satisfaction.” This interactive approach is not new: Karl Sims [15] has extensively shown the power of the method in the framework of computer graphics (see also [1]).

This application, based on the software PROGON we developed, has been developed in JAVA in order to provide a versatile exploration tool.

The fitness function of the GP algorithm is made of two parts:

- an “internal” fitness, that only depends on the characteristics of the individual which represents a Mixed IFSs: it takes into account contractivity constraints (as in [14]) and a density parameter (set by the user), in order to generate attractors that tend to occupy a prescribed number of pixels in the image.
- an “external” fitness, which is set by the user through the JAVA interface during the GP run: only the 6 best

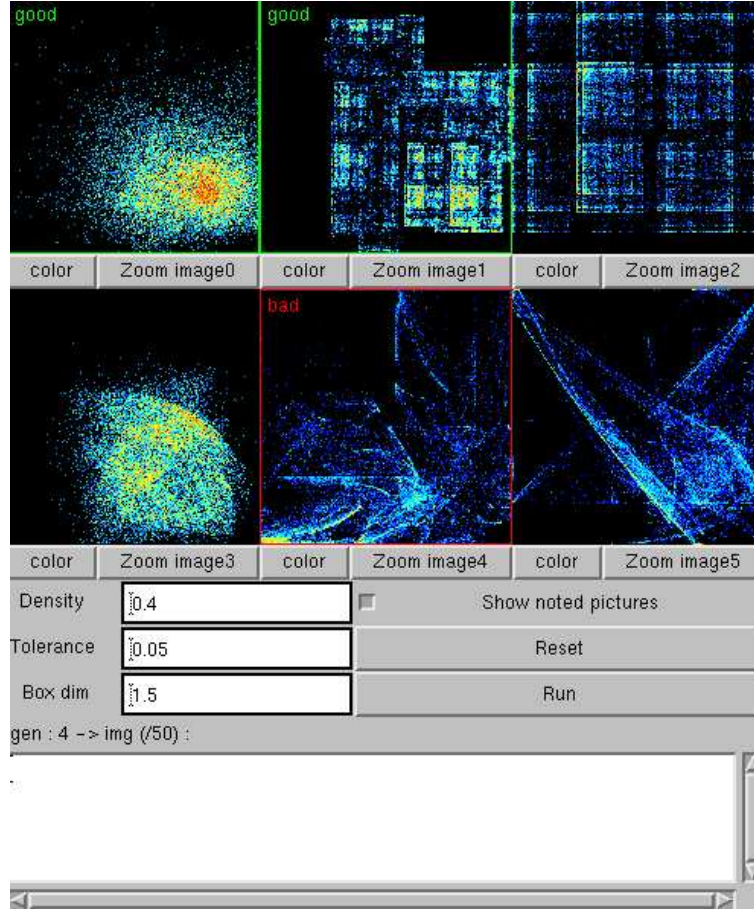


Figure 1: JAVA interface of the algorithm. The GP uses a population of 50 individuals.

individuals of the previous generation are presented to the user. Marks affected can take three values: nice, indifferent and ugly. Figure 1 shows the window, and a user's vote.

The global fitness, that the GP algorithm tries to maximise, is simply the sum of internal and external fitness. Figure 2 presents images obtained with this applet. Of course, the functions of these IFSs are saved and can be used in order to generate more precise or larger attractor images.

5 Polar IFS

Let us use a subset of non-linear functions, w_i , contracting with respect to a point P_i :

$$\forall M \in E = [0, 1]^2 \quad \|\overline{P_i w_i(M)}\| < \|\overline{P_i M}\| \quad (1)$$

which can be transcribed in polar coordinates centered on P_i as (th represents the hyperbolic tangent):

$$\overline{P_i w_i(M)} = \begin{pmatrix} \rho \frac{th(k * F(\rho, \theta)) + 1}{2} \\ G(\rho, \theta) \end{pmatrix} \quad (2)$$

$F(\rho, \theta)$ and $G(\rho, \theta)$ are random non-linear functions which can be represented with a tree (as for mixed-IFS functions).

The form $\rho^{\frac{th(k * F(\rho, \theta)) + 1}{2}}$ insures that the relation (1) is verified, because the factor $\frac{th(k * F(\rho, \theta)) + 1}{2}$ is always < 1 . The form of this factor has been chosen in order to make a rather smooth bijective mapping of \mathbb{R} onto $(0, 1)$. k is fixed to 10^{-7} for the same reasons.

The central points P_i of these w_i functions are fixed points:

$$\forall M \in E \quad \lim_{n \rightarrow \infty} w_i^n(M) = P_i$$

However, these functions are not systematically contractant. Thus the restriction to functions verifying (2) does not unfortunately prevent from checking whether the functions are contractant or not. This restriction is however very interesting, because functions constructed this way are quite often contractant and have a known unique fixed point. What is more, this set of functions is general enough to produce a wide variety of shapes through their attractor, see figure 3 to 6.

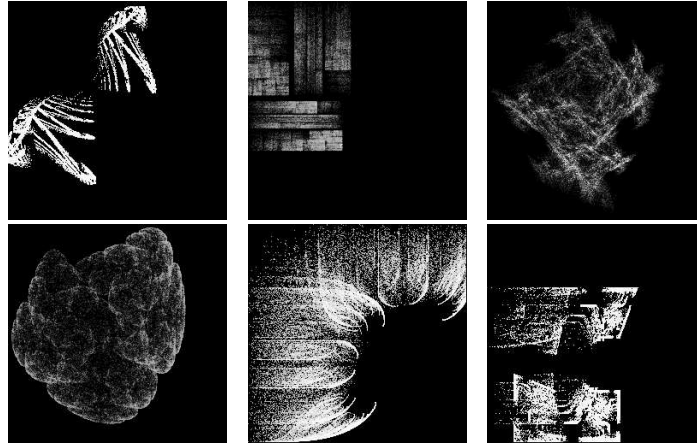


Figure 2: Some Mixed IFS attractors generated using interactive GP

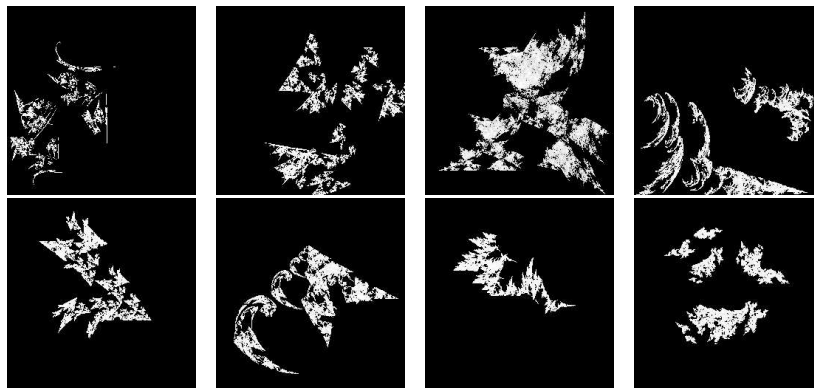


Figure 3: Examples of Polar IFSs attractors

6 ‘Individual’ approach in evolutionary programming

The standard approach, which uses evolutionary methods as stochastic optimisers (where a set of individuals in the search space evolves, via specific or classical genetic operators, so that the best individual of the population converges towards the desired optimum) may sometimes seem wasteful : only the best individual of the final population is kept, while the others are discarded. The behaviour of GA however leads to think that an important part of the final population bears significant information on the structure of the search space. This constatation has led to, and justified such techniques as *sharing*, *niching* or *Michigan type* GA’s (see for example *classifier systems* [9] or [13]), that get more out of evolutionary algorithms than guiding the best individual towards the global optimum.

If the solution to the problem is represented by an important set of individuals, or by the whole population, the implementation of the algorithm is more delicate:

- All optimisation problems may not be formulated into a union of split sub-problems.

- One must be able to correctly evaluate the contribution of each of the individuals to the global solution (one can quite often use a local evaluation function for an individual along with a global evaluation function for the whole population). Finally, it seems indispensable to use a “sharing” method with a well chosen distance function to place each individual with reference to the others. The evolution of the system can be seen as seeking a position of balance.

In order to manipulate Polar IFSs using an *individual GP* approach, for instance, each individual of the population represents a w_i function. The IFS is then represented by the whole population, or by a part of the population. Each w_i is mainly evaluated as a function of the position of its fixed point (which is always defined and known, thanks to the use of polar IFS) and as a function of its attractor with respect to the problem to be solved. A distance is defined on the search space (*sharing* method) to get the individuals to be as far as possible one of each other (linked to the Euclidean distance between the w_i fixed points). The w_i population then evolves so that the attractor of the best individuals of the population has the desired property.

7 Random generation of Polar IFS attractors with a prescribed surface

If each individual of the population encodes one single w_i function, the IFS is represented by the whole population, or by a part of the population. We present below how this structure can be evolved to produce an IFS having an attractor of arbitrary surface in the image.

The advantage of using Polar IFSs is twofold:

1. each function converges towards a fixed point, and the functions are rather frequently contractive, see [6] for experiments,
2. access to the fixed point of the function is direct.

This second point allows to consider differently genetic operators on fixed points and on tree structures. Genetic operators are classical GP *mutation* and *crossover* for the tree structured part of the w_i . Fixed points are mutated according to a random shot in a disk of radius σ centered on it. Fixed points crossover is a classical barycentric crossover whose parameter is randomly chosen in $[-1, 2]$ for each offspring (i.e. an offspring $P' = \gamma P + (1 - \gamma)Q$ for the parents P and Q).

The main problem of the “individual” approach is the need to carefully design the fitness function, and to use a sharing scheme, in order to disperse the individuals of the population.

The fitness function can be made up of two main contributions: a local fitness that measures the intrinsic characteristic of the individual, and a global fitness that stems from the global performance of the population, redistributed on each individual, proportionally to its “contribution.”

- **Local fitness:** a combination of three terms according to:

1. *The fixed point position* with respect to the image (represented as \square , the $[0, 1]^2$ square).

A very simple property of attractors is that the fixed point of each w_i belongs to the attractor of $\{w_i\}$. Thus, if we wish the attractor of the $\{w_i\}$ to be inside the image, the fixed points need to be within \square .

A first term is proportional to the inverse distance between x_i^* the fixed point of w_i and \square .

$$F_1(w_i) = \frac{1}{1 + D(x_i^*, \square)}$$

F_1 is maximum and equals 1 when $x_i^* \in \square$, and tends to zero when x_i^* is far away from \square .

2. *The $w_i(\square)$ position with respect to \square :* We compute the image $w_i(\square)$, in order to test if it is included in \square .

Let $\#[X]$ be the number of pixels (according to the considered image resolution) of the set X ,

then:

$$F_2(w_i) = \frac{\#[w_i(\square) \cap \square]}{\#[w_i(\square)]}$$

$F_2(w_i)$ is maximum (and = 1) if $w_i(\square) \subset \square$.

3. *The size of $w_i(\square)$ in \square :*

$$F_3(w_i) = \frac{\#[w_i(\square) \cap \square]}{\#[\square]}$$

We have chosen to favour w_i 's that generate large images.

The local fitness for individual w_i is:

$$F_{loc}(w_i) = F_1(w_i) + F_2(w_i) + \alpha F_3(w_i) \quad (3)$$

α tunes the relative importance of term F_3 in comparison to F_1 and F_2 (F_1 and F_2 tend easily to one, while F_3 is more difficult to make grow).

- **Contractivity constraints:**

The contractance test can be included in the computation of the image of $w_i(\square)$. In the same time, the mean contraction factor s_i can be estimated. If the function is not contractive, F_2 and F_3 are not computed and are directly fixed to zero, as well as F_{loc} in order to discard this individual.

- **Global fitness:**

The N (to be determined with respect to the local fitness²) best individuals of the evolved population represent a solution to our problem. A toss-coin algorithm can thus be used in order to compute the attractor \mathcal{U} of these individuals, and a global fitness can be defined for a prescribed image occupancy $S \in [0, 1]$ as:

$$F_{glob} = \frac{2}{1 + 100 \left(\frac{\#[\mathcal{U}]}{\#[\square]} - S \right)^2}$$

F_{glob} is a measurement of the distance between $\frac{\#[\mathcal{U}]}{\#[\square]}$ and S . The function has been chosen so that $F_{glob} \in [1, 2]$ when $\#[\mathcal{U}] \in [S - 10\%, S + 10\%]$.

This global fitness can be distributed on the N w_i which have been selected from the current population (the global fitness of the individuals that have not been selected is simply F_{loc}), proportionally to their contribution to \mathcal{U} i.e.: to $F_2(w_i)$, or grossly to s_i :

$$F(w_i) = F_{loc}(w_i) \times N \frac{s_i}{\sum s_i} F_{glob} \quad (4)$$

F_{glob} is used as a multiplicative factor, thus improving (if ≥ 1) or degrading the individuals' fitness with respect to their global performance.

²In fact, we select all the contractive individuals of the population with $F_3(w_i) > 0.1$.

Table 1: Parameters setting for the random generation of attractors, using GA-Lib

SIGMA	0.2
Local fitness tuning	$\alpha = 20$
Mutation probabilities	
constant \rightarrow constant	0.15 according to a Gaussian law of variance SIGMA
variable \rightarrow constant	0.02 randomly chosen in $[-1, 1]$
constant \rightarrow variable	0.06
variable \rightarrow variable	0.08
function \rightarrow function (same arity)	0.08
fixed points:	0.03 according to a uniform law in the circle of radius SIGMA
Crossover probability	
PCROSS	0.95 for trees and fixed points
Sharing	
σ (Goldberg)	$2 * \text{SIGMA}$
Population replacement scheme	
replacement percentage	50% with overlapping populations

F_{glob} can also be used as a stopping criterion for the GP: stop the algorithm when the target surface is approximated with a fixed threshold.

A GP with sharing is used, the distance being simply the Euclidean distance between fixed points of the w_i functions.

Results obtained with the parameter setting of table 1 are presented in figures 4 and 5.

8 Conclusion

We have presented in this paper how GP can be used to generate “nice” fractal 2D shapes. Important issues highlighted were:

- the potential power of “individual” approaches (see also [5] and [13] for other applications) in Artificial Evolution for applications where the problem can be split into inter-dependent subproblems such as the generation of IFS attractors,
- the Polar IFS model, which provides a less sparse search space: this is particularly interesting in applications such as the one we presented here or for the resolution of the 2D inverse problem for IFS (given a target shape, find the most similar IFS attractor). Inverse problem for 2D Polar IFS has been studied in [5], and can be applied for fractal image compression or data representation [17].

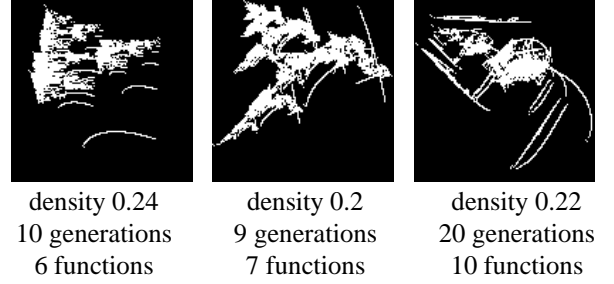


Figure 4: Attractors in a 128x128 image, generated for $S = 0.2$, with a population size of 20 individuals. The algorithm is stopped as soon as the density is $> S$.

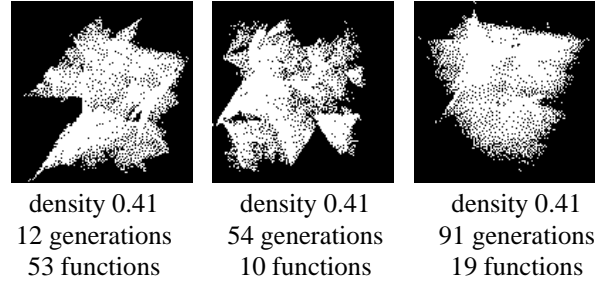


Figure 5: Attractors in a 128x128 image, generated for $S = 0.4$, with a population size of 30 individuals, the algorithm is stopped as soon as the density is $> S$.

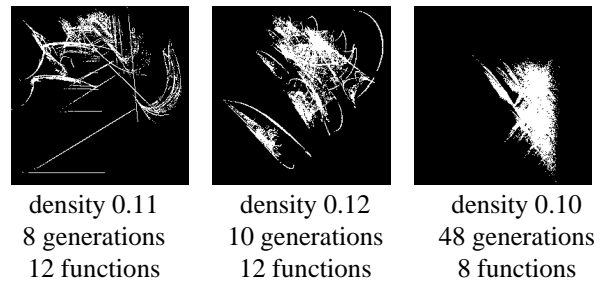


Figure 6: Attractors in a 256x256 image, generated for $S = 0.1$, with a population size of 30 individuals, the algorithm is stopped as soon as the density is $> S$.

Future work on this topic will mainly concern the implementation of a general interactive “individual” GP for the generation of Affine, Mixed and Polar IFSs attractors.

Bibliography

- [1] Peter J. Angeline, “Evolving Fractal Movies”, *Genetic Programming 1996: Proceedings of the First Annual Conference*, John R. Koza and David E. Goldberg and David B. Fogel and Rick L. Riolo (Eds), pp 503–511, 1996.
- [2] M. Barnsley and S. Demko, “Iterated Function System and the Global Construction of Fractals,” *Proceedings of the Royal Society, A* 399:243–245, 1985.
- [3] M. Barnsley, S. Demko, J. Elton, and J. Geronimo, “Invariant Measures for Markov Processes Arising from Iterated Function Systems with Place-Dependent Probabilities,” *Georgia Tech.* preprint.
- [4] M. F. Barnsley, “Fractals Everywhere,” *Academic Press*, New-York, 1988.
- [5] Pierre Collet, Evelyne Lutton, Frédéric Raynal, and Marc Schoenauer, “Individual GP: an Alternative Viewpoint for the Resolution of Complex Problems,” submitted to *GECCO99*.
- [6] Pierre Collet, Evelyne Lutton, Frédéric Raynal, and Marc Schoenauer, “Polar IFS and “Individual” Genetic Programming,” *Technical report*, INRIA Research Report, 1999.
- [7] J. H. Elton, “An Ergodic Theorem for Iterated Maps,” in *Georgia Tech. preprint*, 1986.
- [8] Y. Fisher, “Fractal Image Compression,” in *Siggraph 92 course notes*, 1992.
- [9] David A. Goldberg, “Genetic Algorithms in Search, Optimization, and Machine Learning,” *Addison-Wesley Publishing Company, inc.*, Reading, MA, January 1989.
- [10] D. P. Hardin, “Hyperbolic Iterated Function Systems and Applications,” *PhD thesis*, Georgia Institute of Technology, 1985.
- [11] J. Hutchinson, “Fractals and Self-Similarity,” *Indiana University Journal of Mathematics*, 30:713–747, 1981.
- [12] A.E. Jacquin, “Fractal Image Coding: a Review,” *Proc. of the IEEE*, 81(10), 1993.
- [13] Jean Louchet, “Obstacle Detection and Segmentation-Free Stereo-Vision: an Evolutionary Approach,” *submitted to EVOIASP 99 Workshop*.
- [14] E. Lutton, J. Lévy Véhel, G. Cretin, P. Glevarec, and C. Roll, “Mixed IFS: resolution of the inverse problem using genetic programming,” *Complex Systems*, 9:375–398, 1995. (see also Inria Research Report No 2631.)
- [15] K. Sims, “Interactive evolution of dynamical systems,” in *First European Conference on Artificial Life*, pages 171–178, 1991. Paris, December.
- [16] Karl Sims, “Artificial Evolution for Computer Graphics,” *Computer Graphics*, 25(4):319–328, July 1991.
- [17] Jacques Lévy Véhel, Khalid Daoudi, and Evelyne Lutton, “Fractal modeling of speech signals,” *Fractals*, 2(3):379–382, September 1994.