ArtiE-Fract : Interactive Evolution of Fractals

Jonathan Chapuis, Evelyne Lutton

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

Jonathan.Chapuis@inria.fr, Evelyne.Lutton@inria.fr

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

Abstract

Non-linear Iterated Functions 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 an interactive evolutionary algorithm can be efficiently exploited in order to generate randomly or interactively artistic “fractal” 2D shapes. This algorithm has been build up in an easy-to-use interface ArtiE-Fract with advanced interactive tools.

1 Introduction

Fractal images have been considered as interesting artistic objects as they combine complexity and some "hierarchical" structure. The complete mathematical structure that supports these pictures provides indirect access to their characteristics and, therefore, allows shape manipulation and exploration. ArtiE-Fract is a user friendly software for the creation of fractal images based on an interactive evolutionary algorithm.

Evolutionary algorithms (EA) are nowadays known as powerful stochastic optimisation techniques but can also be used in order to generate artistic pictures. The appropriate tool is interactive EA, i.e. an EA where the function to be optimised is partly set by the user, in order to optimise something related to the "user 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]). We extend this approach to the exploration of a fractal images space and improve its flexibility with help of advanced interactive tools related to the specific fractal model we use.

ArtiE-Fract evolves a population of fractal images, and displays it via an interface. More precisely, these fractal images are encoded as sets of contractive non-linear 2D functions (affine and non-affine), defined either in Cartesian or polar coordinates. Each set of these contractive functions represents an IFS (Iterated Functions System), to which a particular 2D image, its attractor, is associated.

In ArtiE-Fract the interaction is twofold:

– a classical interaction (as in [16]): the user guide the EA by giving notations to each image of the population via the main window that displays the whole population.

– a direct interaction: images can be manipulated via a specialized window and modified individuals can be added or replaced in the current population (this is a sort of interactive "local" deterministic optimisation). A large set of geometric, colorimetric, structural modification are available. Moreover, due to the IFS model, some control points can be defined on the attractor images (fixed points) that help to distort the shape in a convenient, but non trivial, manner.

The ArtiE-Fract interface has been carefully designed in order to give access to a wide variety of parameters. This, together with the two particularities of giving access to unusual fractal images (non-linear IFS), and allowing the user to interfere at any time with the evolutionary process, make of this software a flexible and user-friendly artistic image generation tool.

This paper is organized as follows: IFS theory and attractor's construction are described in section 2 and functions classes (affine, mixed, polar) are detailed in section 3. In section 4, ArtiE-Fract interactive capabilities are developed.

2 IFS theory

Iterated Functions Systems 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 [10, 3, 4, 8]), and most fractal image compression techniques are based on IFSs [2, 11].

An **Iterated Functions System** (IFS) $\mho = \{F, (w_n)_{n=1,..,N}\}$ 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:

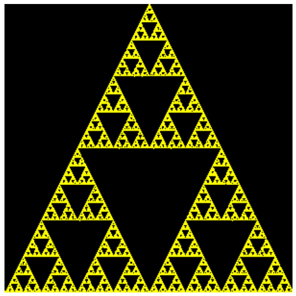
$$\forall\ K \ \subset \ F, \ \ W(K) = \bigcup_{n \in \{1,..,N\}} w_n(K)$$

Then, if the w_n functions are contractive (the corresponding IFS is then called *contractive* IFS), there exists a unique set A such that: $W(A) = A$. A is called the **attractor**[1] of the IFS.

The uniqueness of a contractive attractor is a result of the Contractive Mapping Fixed Point Theorem for W, which is contractive according to some distance (the Hausdorff distance, see [14]).

Figure 1 displays the Sierpinski triangle. It is the attractor of an IFS made of three affine (see 3.1) functions, all having a scaling factor of 1/2. This attractor has a fractal dimension of 1.66.

[1]An IFS attractor A can be considered as a "fractal" set because the relation $W(A) = A$ reads $\bigcup w_i(A) = A$, meaning that A is exactly the union of reduced / transformed copies of itself (self similarity principle).



$$w_1 = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$
$$w_2 = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} + \begin{bmatrix} \frac{1}{2} \\ 0 \end{bmatrix}$$
$$w_3 = \begin{bmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{bmatrix} + \begin{bmatrix} \frac{1}{4} \\ \frac{1}{2} \end{bmatrix}$$

Figure 1: The Sierpinski triangle and its three functions

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

- **Deterministic method:** a straightforward implementation that simulates the convergence of a sequence of sets $\{S_n\}$: $S_{n+1} = W(S_n) = \bigcup_i w_i(S_n)$ from any initial set S_0 (see figure 2).

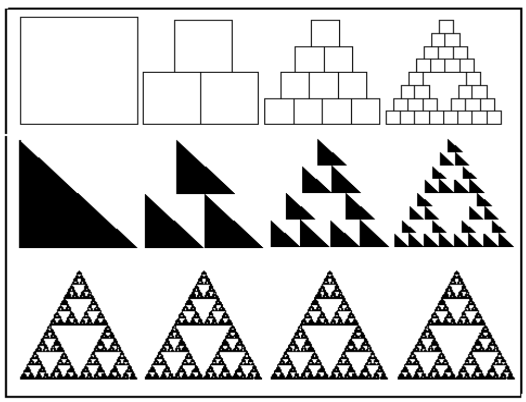


Figure 2: Deterministic construction of the Sierpinski triangle: from any initial image (left), the sequence S_n converges to the Sierpinski triangle. Note that the Sierpinski triangle is invariant with respect to W (last row).

- **Stochastic method (toss-coin):** it has been shown that the following point sequence: $x_{n+1} = w_i(x_n)$, i being randomly chosen in $\{1..N\}$, starting from any of the w_i fixed points, provides an approximation of the real attractor of $\mho$.

The stochastic method is usually preferred due to its computational efficiency and is used in ArtiE-Fract.

The colors of the attractor are set according to the number of time each pixel of the attractor is hitted by the toss-coin sequence. It is related to the invariant measure associated with the IFS (see [4]).

3 Contractive functions classes

Usual attractor images and compression techniques are based on affine IFS, however our recent works on the topic tend to prove that non affine IFS (mixed or polar), i.e., IFS that are not anymore restricted to be made of affine functions, are interesting in many applications (see [17, 12]) and moreover from an artistic viewpoint.

3.1 Affine functions

In the case of affine IFS, each contractive affine map w_i of $\mho$ is represented as:

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

Affine functions are combinations of simple geometric transformation: scaling, symmetries, rotations and translations. Their contractance factor is directly calculated as the maximum of the module of the eigen values. Affine IFS are thus easy to handle, which explains their success.

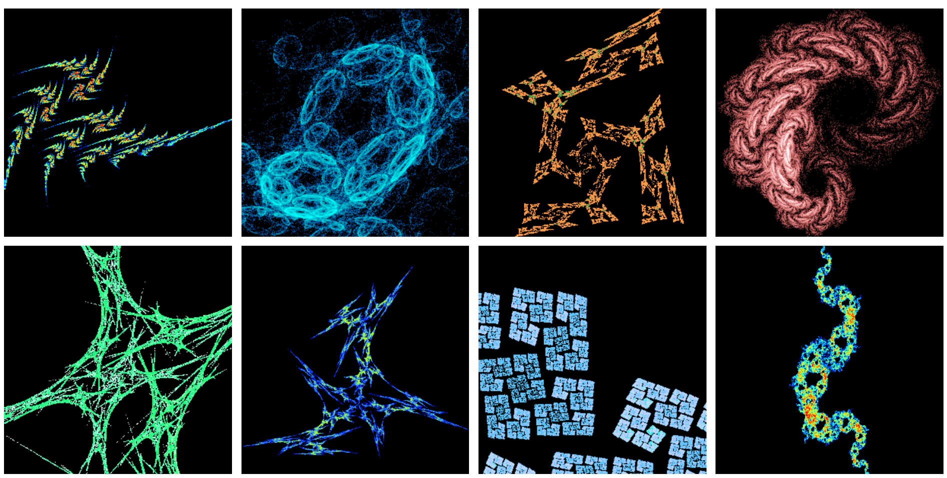


Figure 3: Examples of affine IFS attractors

3.2 Mixed functions

We use the term **mixed** IFS [13] in order to emphasize the fact that the w_i functions are not anymore restricted to be affine[2]. In this case, the first point to be addressed is the one of finding an adequate representation. A natural one is based on trees (see [14]); the w_i functions are built from a set of basic operators ($+$,$-$,$\times$,$/$, pow, log, exp, sin, cos, $\ldots$), a set of variables (x and y), and a set of constants. In the following examples, the constants belong to $[-1, 1]$.

[2] In 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.

Another difficult problem for mixed functions is the contractance check for each w_i in order to select contractive IFSs. On the contrary to affine functions, this verification is not straightforward, and is in fact computationally intractable. We thus rely on some heuristics that reject strongly non-contractive functions. The simplest way to do that (see [12] for details) is to verify the contractivity on sample points.

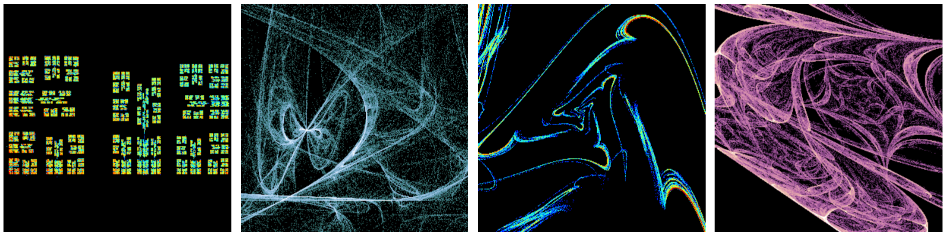


Figure 4: Examples of mixed IFS attractors

3.3 Polar Functions

In order to have a better control on the contractance, a subclass of mixed functions is introduced: **polar** functions. The w_i are encoded in polar coordinates centered on a point P_i as (th represents the hyperbolic tangent):

$$w_i(\rho,\theta) = \begin{pmatrix} \dfrac{th(kF(\rho,\theta))+1}{2}\rho \\ G(\rho,\theta) \end{pmatrix}$$

$F(\rho,\theta)$ and $G(\rho,\theta)$ are non-linear functions which can be represented with a tree (as for mixed functions). The factor $\frac{th(kF(\rho,\theta))+1}{2}$ is always < 1 and therefore ensures the convergence of these functions toward the central point P_i (see [14]).

Contractance tests are still necessary (convergence toward a point does not ensure contractance), but the search space of polar contractive functions is less sparse that the one of contractive mixed functions (see [7]).

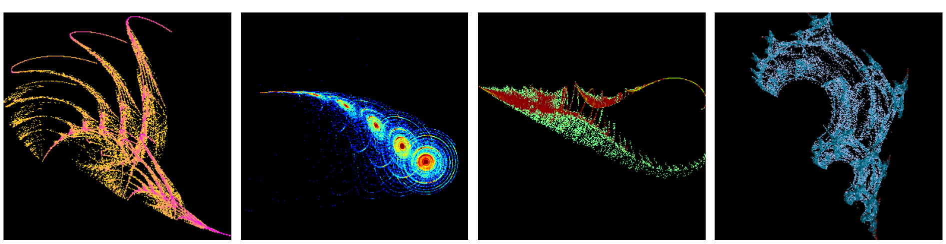


Figure 5: Examples of polar IFS attractors

4 ArtiE-Fract

ArtiE-Fract is based on an interactive EA designed to help the user to explore the space of fractal shapes encoded with the previous IFS models.

4.1 Interactive evolutionary algorithm

EAs and can be considered as a computer implementation of a Darwinian evolution model. Their main characteristic is that they manipulate populations of individuals (that represent solutions, points of a search space, programs, rules, images, signals, etc ...), and involve a set of operations (selection, mutation, crossover) applied randomly to each individual, in order to simulate a sequence of generations. If correctly designed, this dynamic stochastic process concentrates the population onto the global optimum of the search space.

EAs are useful for other purposes than pure optimisation and, for example, for the generation of artistic pictures. They act as an exploration tool in an image space, the implicitly optimised function being the "user's satisfaction." In ArtiE-Fract, the fitness function is made of two parts:

- an "internal" fitness, that depends only on the characteristics of the individual which represents an IFS: density, fractal dimension, brightness, contrast and lacunarity (provides informations about the distribution of the density of the attractor).
- an "external" fitness, which is set by the user during the run. Marks range from -1 (worst) to 6 (best), see figures 6 and 7.

The global fitness, that the algorithm maximises, is simply the sum of internal and external fitnesses.

The EA stops at each generation allowing user interaction: notations, direct modification, or a new generation run command. Figure 6 shows the main display window of ArtiE-Fract: it presents all the attractors of the population, so that the user can see them and eventually rank or capture them to make its own modifications. Here individuals 0, 1, 4, 7 received a good mark and 3 a negative one.

4.2 Genetic operators

Default parameters and operators are set in order to allow an efficient exploration of the image space. However the user has access to the majority of these parameters via some advanced parameters window. Because of the complexity of the individuals (IFS), there are many different operators which act at two levels:

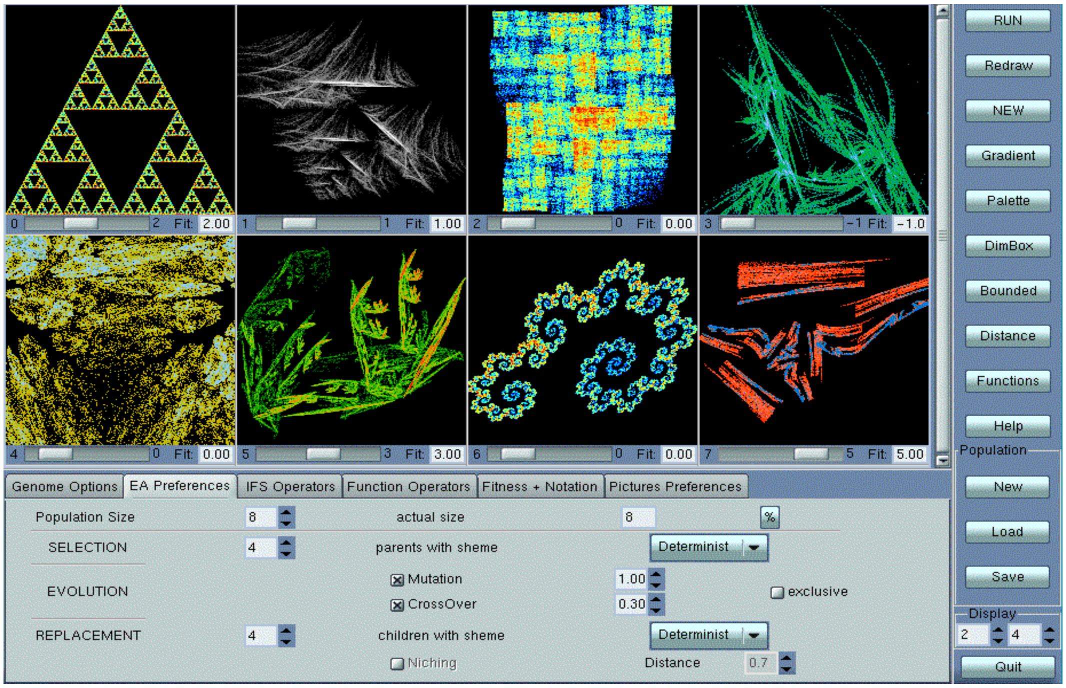


Figure 6: Display of ArtiE-Fract with various user's ranking. The population contains 8 individuals (with their corresponding number of functions of each type {affine, mixed, polar}): zero {3,0,0}, one {1,1,0}, two {0,2,0}, three {2,0,1}, four {2,0,0}, five {5,0,0}, six {2,0,0}, seven {1,1,0}.

at the IFS level: change the IFS structure by adding, deleting, exchanging a function, move one or more fixed points, change the toss-coin probabilities, modify the palette (it is composed of control points interpolated by splines or linear curves), rescale, center.

at the function level: gaussian mutation of constants, tree mutations (operator ↔ variable ↔ constant) and crossover, combination of functions.

Figure 7 displays one generation step for the population of figure 6: four new IFSs were obtained (top images: 0 - 3) from four parents (bottom: 4 - 7). A polar function was added to individual 0, a fixed point of individual 1 was moved and one function of each individual 6 and 7 was mutated to produce respectively individuals 2 and 3.

4.3 The user interaction tools

The user interaction tools of ArtiE-Fract are globally the same as the genetic operators described before, but are activated and controlled directly by the user: zooming, translation, change of the functions composition of the IFS, displacement of fixed point, modifications of the color palette. The user can pick up an individual, modify it according to his taste, and finally replace an individual or add it to the current population. This modified image may also be saved for further use.

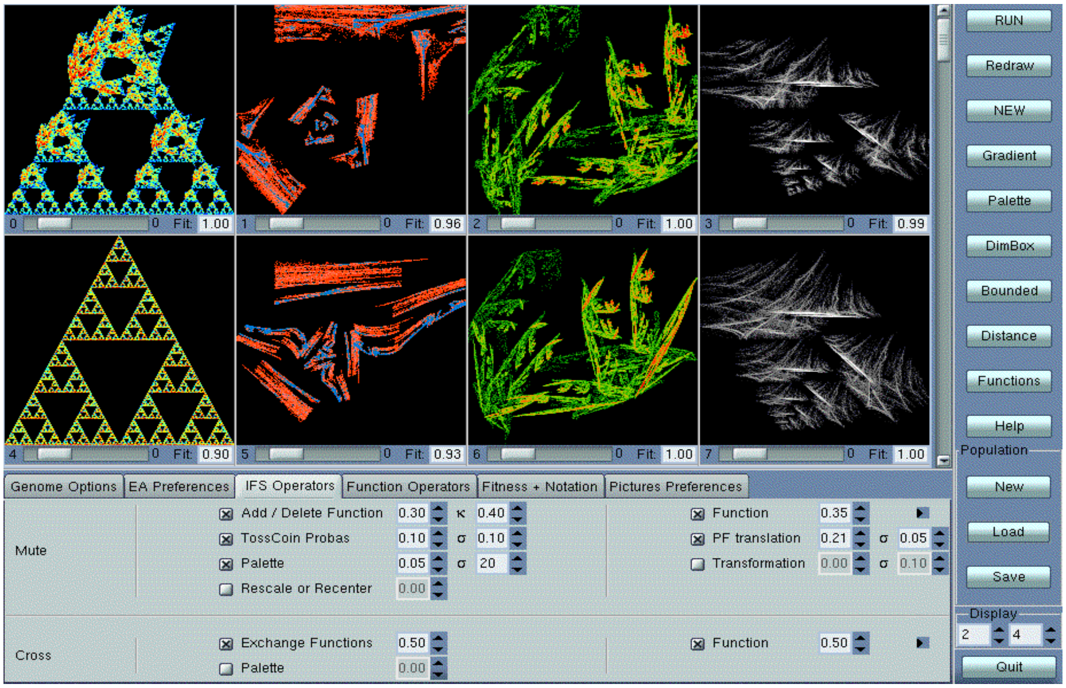


Figure 7: Evolution of the population of figure 6

The translation of a fixed point may have non trivial effects on the attractors shapes. Figure 8 shows its impact on a simple IFS made of two affine functions.

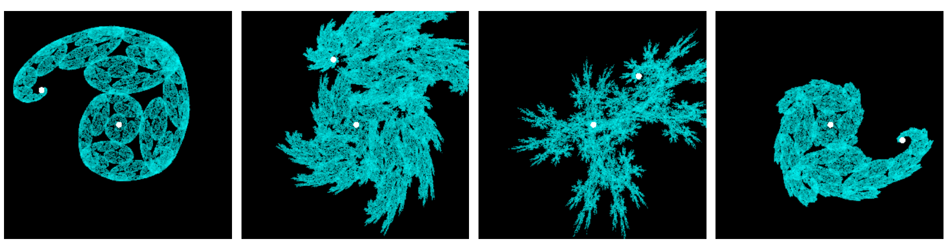


Figure 8: Effect of fixed point displacement for a two affine functions IFS: the external fixed point is moving around the central one (fixed point are white dots).

Other tools available are:

a function displayer so that the user may visualise the selected function in order to access and modify it directly as a formula. It also provides information about the distribution of a function inside the current population.

an initial population generator that provides many options: type and number of functions, specification of the tree components, color palette, attractor density.

At any time, the whole population can be saved and loaded again.

5 Conclusion

Figure 9 displays a sample of fractal images that may be generated with ArtiE-Fract. Although it is still under development (other interactive capabilities, new fractals models), ArtiE-Fract, in its present version, can already be considered as a flexible and efficient exploration tool of IFS fractal shapes. Experiments with some designers and advertisers tend to prove that it is an interesting tool for numerous artistic applications.

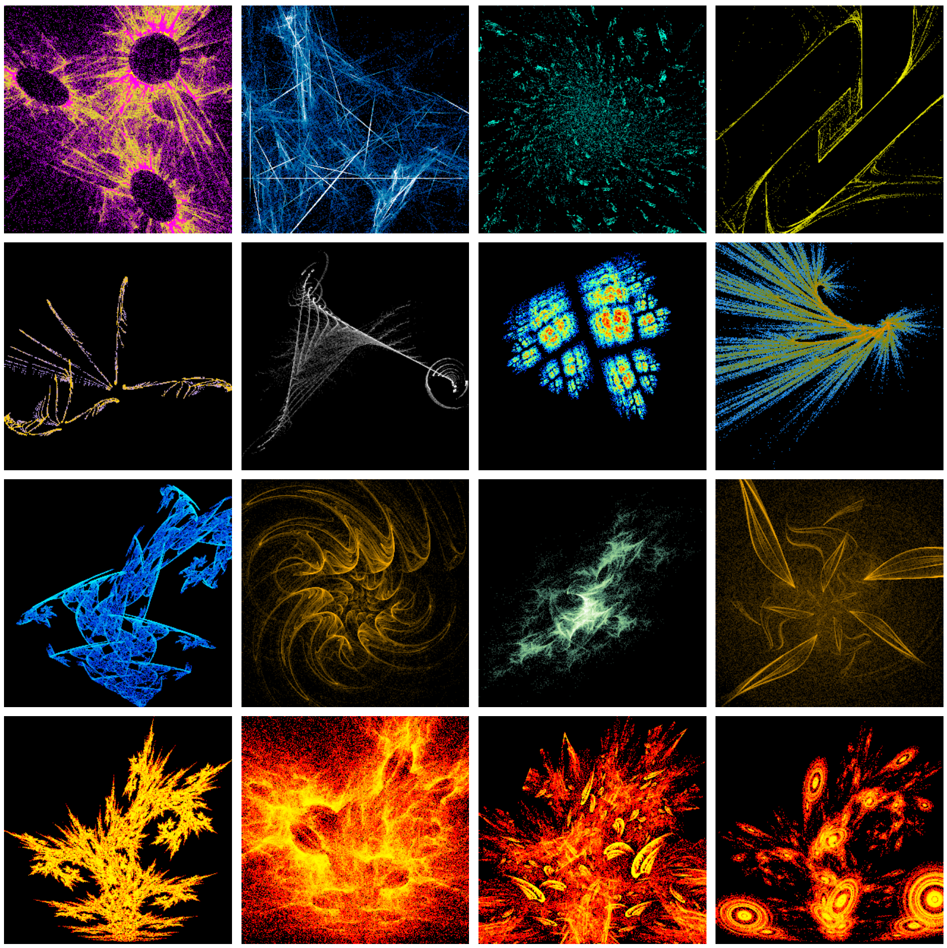


Figure 9: Gallery

References

[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. F. Barnsley, "Fractals Everywhere," *Academic Press*, New-York, 1988.

[3] M. Barnsley and S. Demko, "Iterated Functions System and the Global Construction of Fractals," *Proceedings of the Royal Society*, A 399:243–245, 1985.

[4] M. Barnsley, S. Demko, J. Elton, and J. Geronimo, "Invariant Measures for Markov Processes Arising from Iterated Functions Systems with Place-Dependent Probabilities", *Georgia Tech.* preprint.

[5] Pierre Collet, Evelyne Lutton, Frédéric Raynal, and Marc Schoenauer, "Individual GP: an Alternative Viewpoint for the Resolution of Complex Problems," GECCO 99, *Genetic and Evolutionary Computation Conference*, Orlando, Florida, USA, July 13-17, 1999.

[6] Pierre Collet, Evelyne Lutton, Frédéric Raynal, and Marc Schoenauer, "Polar IFS and "Individual" Genetic Programming," *Technical report*, INRIA Research Report, 1999.

[7] Pierre Collet, Evelyne Lutton, Frédéric Raynal, Marc Schoenauer, "Polar IFS + Parisian Genetic Programming = Efficient IFS Inverse Problem Solving", *Genetic Programming and Evolvable Machines Journal*, Volume 1, Issue 4, pp. 339-361, October 2000.

[8] J. H. Elton, "An Ergodic Theorem for Iterated Maps," in *Georgia Tech. preprint*, 1986.

[9] David A. Goldberg, "Genetic Algorithms in Search, optimisation, and Machine Learning," *Addison-Wesley Publishing Company, inc.*, Reading, MA, January 1989.

[10] J. Hutchinson, "Fractals and Self-Similarity," *Indiana University Journal of Mathematics*, 30:713–747, 1981.

[11] A.E. Jacquin, "Fractal Image Coding: a Review," *Proc. of the IEEE*, 81(10), 1993.

[12] 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.)

[13] Evelyne Lutton, Jacques Lévy Véhel, Guillaume Cretin, Philippe, Glevarec and Cedric Roll, "Mixed IFS: resolution of the inverse problem using Genetic Programming", *In Complex Systems*, vol. 9, number 5, October 1995 and *INRIA research report*, RR-2631, 17 p., August 1995.

[14] Frédéric Raynal, Evelyne Lutton, Pierre Collet, "Manipulation of Non-Linear IFS Attractors Using Genetic Programming" CEC99, *Congress on Evolutionary Computation*, Washington DC, USA, July 6-9, 1999.

[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", *In Fractals*, Vol. 2, Issue No. 3, pp. 379-382, June 1994.

[18] Some related web sites:
http://www.medialsace.fr/ifs/
http://www.cosy.sbg.ac.at/rec/ifs/