International Journal on Artificial Intelligence Tools

Evolution of Fractal shapes for artists and designers

Evelyne LUTTON

COMPLEX team - INRIA - Rocquencourt,
B.P. 105, 78153 LE CHESNAY Cedex, France,
Evelyne.Lutton@inria.fr
http://complex.inria.fr

Received (Day Month Year)
Revised (Day Month Year)
Accepted (Day Month Year)

We analyse in this paper the way randomness is considered and used in ArtiE-Fract. ArtiE-Fract is an interactive software, that allows the user (artist or designer) to explore the space of fractal 2D shapes with help of an interactive genetic programming scheme. The basic components of ArtiE-Fract are first described, then we focus on its use by two artists, illustrated by samples of their works. These "real life" tests have led us to implement additionnal components in the software. It seems obvious for the people who use ArtiE-Fract that this system is a versatile tool for creation, especially regarding the specific use of controlled random components.

1. Introduction

The programming of a software for artistic purposes is a challenging task: the computer software framework usually locks the user inside many interaction constraints that are sometimes considered as an obstacle to creativity.

Recent advances in interactive evolutionary algorithms (IEA) [2] have initiated many attractive works, mainly based on the idea of "maximising the satisfaction of the user" via a guided random search [1,12,14,15,17,20,22,21,23]. The use of randomness in this particular way yields an additional "creativity component," that may or may not be considered as helpful by the artist, with respect to the way this random component is controllable. The success of such approaches is thus strongly dependent at least on the choice of a convenient representation and of an adequate set of genetic operators.

Additionally, a problem to be considered very carefully when using randomness in an artistic process is linked to the use of randomness itself, which can be considered by the user as an uncontrollable component of the system, triggering a reflex of reject. A disturbing question can indeed be raised: who is the "artist": the user or the machine ? Both positions can be defended, of course, and several fully machine-driven artistic attempts have been performed.

In the ArtiE-Fract experiments [5], we have centered the system on the artist, and worked to limit this negative perception of the use of randomness. The system has been designed and programmed in order to let the user fully drive or partially interact with the evolutionary process at any moment (ArtiE-Fract is not an "artist"). The idea is to give the possibility to the human user to tame and to adapt the random behaviour of the system to his own artistic approach.

This paper is organised as follows: The ArtiE-Fract software is presented in section 2 the artistic experiments are then detailed in section 3 and 4. Main influences on the design of ArtiE-Fract and conclusions are presented in section 5.

2. ArtiE-Fract: Interactive Evolution of Fractals

Fractal pictures have always been considered as attractive artistic objects as they combine complexity and "hierarchical" structure [4,10]. The notion of fractals is born from the analysis of "strange" mathematical objects (such as an infinite length curve embedded in a finite surface, or as a continuous functions nowhere differentiable) at the end of the XIXth century. The so-called Fractal geometry became famous with the work of Benoit Mandelbrot [13] in the middle of the XXth century. The main argument was that these strange mathematical objects were actually objects that are currently found in nature: fern, coast of Britanny, romanesco caulifower, are examples of natural fractal shapes.

More generally fractal modelling and analysis help to approach the natural complexity: it has emerged from the necessity of building models able to cope with complex and irregular physical or biological phenomena. It has then been extended to the modeling of artifical ones [25], like finance or network traffic analysis.

Here we deal with a model of fractals that became famous in image compression applications [3], but that has many other applications (for example in speech signal processing [7]), namely Iterated Function Systems (IFS). The mathematical structure of iterated function systems attractors [3] let some more or less direct access to its characteristics and therefore, shape manipulation and exploration is possible [8,16].

In ArtiE-Fract, an Evolutionary Algorithm (EA) is used as a controlled random generator of fractal 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 "user satisfaction." This interactive approach is not new in computer graphics [23,20], but we extended it to the exploration of a fractal pictures space and carefully considered flexibility with the help of advanced interactive tools related to the specific fractal IFS model that is used.

The idea of evolving fractal shapes with interactive EA is not new also, since there exists a few similar works. These implementations considers however the problem at a lower complexity level, especially regarding the evolutionary models and the image models. For example Rowley[18] uses a simple genetic engine (mutation only) on images created with affine IFS, and a very simple user-interface. Ventrella[24] uses simple fractal models to produce random animations. Yoshiaki[26] evolves non-linear Julia sets using GP. ArtiE-Fract is indeed the result of a quite long evolution process (ArtiE-Fract is an offspring of a long serie of previous software versions !), and tries to embed a set of user-oriented tools. It intends to approach the interactive efficiency of a "photoshop" or a "gimp" software (of course in its own domain, that is fractal images design) .

2.1. *Interaction with the user*

Interaction with humans usually raises several problems, mainly linked to the "user bottleneck" [15]: human fatigue and slowness[a]. Solutions need to be found in order to avoid systematic and boring interactions. Several solutions have thus been considered [15,21,2]:

- reduce the size of the population and the number of generations,
- choose specific models to constrain the research in *a priori* "interesting" areas of the search space,
- perform an automatic learning (based on a limited number of characteristic quantities) in order to aid the user and only present him the most interesting individuals of the population, with respect to previous votes of the user.

Allowing direct interactions on the phenotype's level represents a further step toward efficient use of IEA as a creative tool for artists. The idea is to make use of the guided random search capabilities of an EA to *aid* the creative process. This is why in ArtiE-Fract, the user has the opportunity to interfere in the evolution at each of the following levels:

- *initialisation*: various models and parameters ranges are available, with some "basic" internal fitness functions (image density, for example);
- *fitness function*: at each generation, a classical manual rating of individuals may be assisted by an automatic learning, based on a set of image characteristic measurements (may be turned on or off), see figure 1;
- *direct interaction with the genome*: images can be directly manipulated via a specialized window and modified individuals can be added or replaced in the current population (a sort of interactive "local" deterministic optimisation). A large set of geometric, colorimetric and structural modifications are available. Moreover, due to the specific image model, some control points can be defined on the images that help distort the shape in a convenient, but non trivial manner, see figure 2;
- *parameter setting and strategy choices* are tunable at any moment during the run.

2.2. *Genome encoding: the IFS model*

An important aspect of ArtiE-Fract is the choice of the search space. As we have told before, a way to limit "user fatigue" is to reduce the size of the search space

[a]The work of the artist Steven Rooke [17] shows the extraordinary amount of work that is necessary in order to evolve aesthetic images from a "primordial soup" of primitive components.

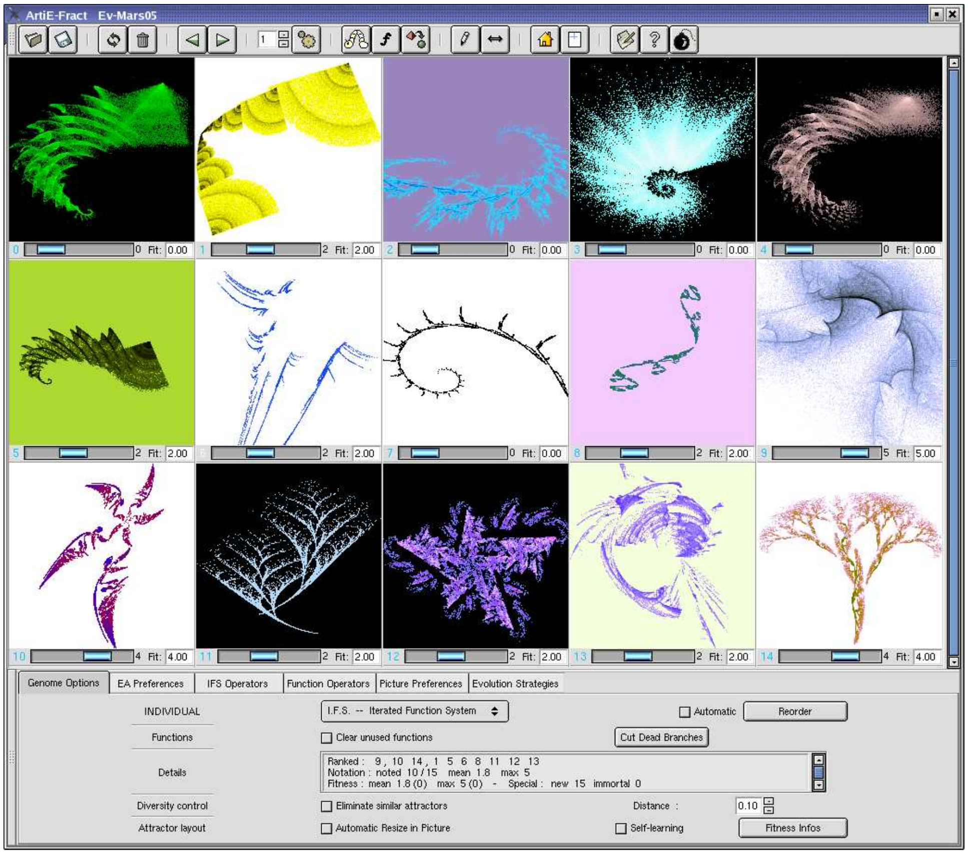

Fig. 1. Main Window of ArtiE-Fract

in order to navigate in a space of *a priori* interesting shapes. The choice made in ArtiE-Fract is the space of 2D fractal shapes encoded as iterated function systems (IFS). This gives access to a wide variety of shapes, that may appear more or less as "naive fractals."

ArtiE-Fract thus evolves a population of IFS attractor pictures, and displays it via an interface, see figure 1. More precisely, these IFS attractor pictures are encoded as sets of contractive non-linear 2D functions (affine and non-affine), defined either in cartesian or polar coordinates. A set of contractive functions represents an IFS, i.e. a dynamical system whose attractor can be represented as a 2D picture. These mathematical objects were considered as interesting as they allow to encode rather complex 2D shapes with a reduced number of parameters.

Iterated Function Systems (IFS) theory is a convenient mathematical tool to define "fractal" images, and has been used with success in many "fractal" image compression applications. The basis of this theory is the simple contractive mapping fixed point theorem. An image is defined as the fixed point of a specific

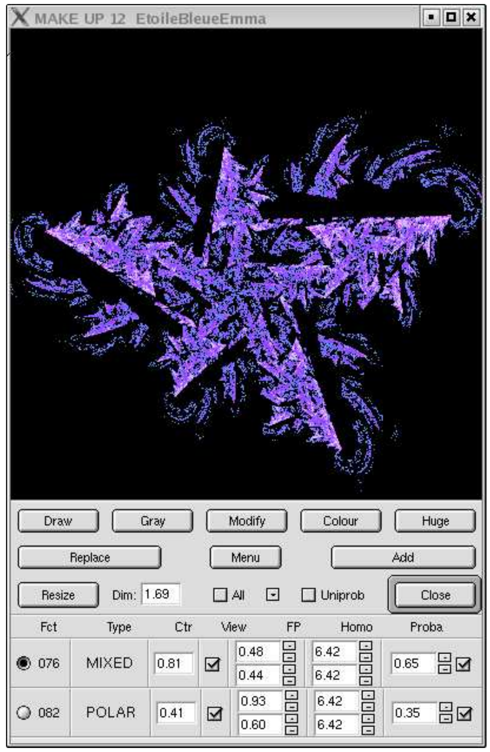

Fig. 2. Direct interaction window

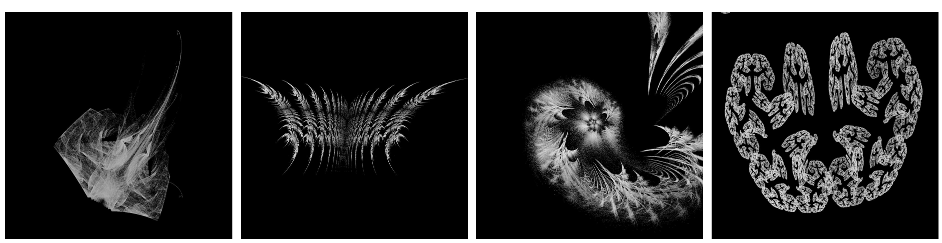

Fig. 3. Sample images obtained with ArtiE-Fract

2D transformation on the space of subsets of the plane. More precisely **an 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 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 $w : F \to F$, from a metric space (F, d) into itself, is called **contractive** if there exists a positive real number $s < 1$ such that:
>
> $$d\big(w(x), w(y)\big) \leq s \cdot d(x, y) \qquad \forall x, y \in F$$

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*:

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

- *Contractive Mapping Fixed Point Theorem*:

  if (F, d) is a complete metric space, and $W : F \to F$ 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", also called "chaos game"):**
  Let x_0 be the fixed point of one of the w_i functions. A points sequence x_n is generated 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 $\mho$. The larger n, the more precise the approximation.
- **Deterministic method:**
  From any initial set S_0, a sets sequence $\{S_n\}$ is built:

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

  When n tends to ∞, S_n is an approximation of the real attractor of $\mho$.

Both methods are used in various applications, but usually the stochastic method is prefered for the generation of 2D IFS attractors as it is less time consuming (the points sequence of the toss-coin algorithm scans the attractor in a random way, i.e. each computation provides a new point of the attractor, while the deterministic method performs an approximation of the attractor, by computing a sequence of complete images). However deterministic methods are used for applications like image and signal compression, fractal zoom, or fractal interpolation.

IFS were extensively studied in the framework of image and signal compression [8,11,19,3], however all IFS models explored in fractal compression were based on affine sets of contractive functions.

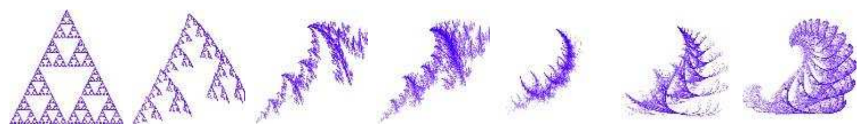

Fig. 4. An IFS morphing built with ArtiE-Fract

From an artistic standpoint, affine IFS give access to an interesting variety of shapes (the "self-affine" fractals). But the use of non-affine functions, beside the scientific interest of exploring this rather unknown space, yields a variety of shapes that may look "less directly" fractal. This is another of the specifics of ArtiE-Fract: three models of IFS are used (affine, mixed and polar), separately or in combination. Each of them induces a slightly different topology on the search space, which gives privileged access to various image types.

Figures 3 and 4 present a set of images created by several users (non-necessarily artists !), that suggest biological images or vegetation, as well as some very "geometric" ones.

This additional freedom, based on the use of non-linear functions seems to be experimentally attractive to artists, as it allows the expression of various inspirations.

In ArtiE-Fract, genomes are represented as trees, which correspond to part of functions, functions, or group of functions (i.e. IFS), depending on the strategy chosen for the genetic engine (classical or parisian, see section 2.4). The tree encoding is a typical genetic programming one, i.e. with constant values (reals), functions (simple ones as $+$, $\times$, $\div$ and more complex as sin, cos, ch, sh, exp, log, $\ldots$) and parameters (x and y for images) .

2.3. *Genetic operators*

The interactive evolution process has been designed in order to perform a constrained and oriented random search in the search space of IFS attractors. The main ingredients of this process are the following:

- An initialisation procedure that produces an initial population, a careful design of this initialisation improves the efficiency of the search, while implicitely restricting the a priori set of shapes to be accessed during the search. An interface yields various parameters settings for the population initialisation. Choices range from fully random to pure deterministic initialisations. The user has the possibility to constrain the set of initial functions and terminals, as well as to bias the random shots in this set. Predefined

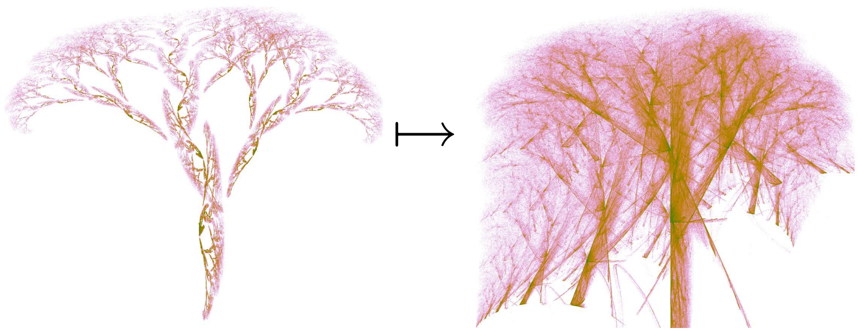

Fig. 5. A mutation

individuals (images or functions) can also be introduced in the population at any time.

- Various genetic operators, acting at various levels, that modify interactively or automatically the composition of the population (mainly mutations and crossovers to generate offspring for the next generation), direct interactions with the user can also be considered as an operator (a so-called "local optimisation") from the viewpoint of the EA. Figures 5 and 6 show examples of genetic operators. Genetic changes at the genome level (in a simple GP style) result in non trivial image modifications.

2.4. *Genetic engine and advanced evolutionary strategies*

Another specific component of ArtiE-Fract is the Parisian approach implementation, which also can be turned on or off at any moment of the evolution. This component has been designed to favour exploration and genetic diversity.

The Parisian approach has been designed relatively recently [6] and is actually a generalisation of the classifier systems approaches [9]. It is based on the capability of an EA not only to push its best individual toward the global optimum, but also to drive its whole population in attractive areas of the search space. The idea is then to design a fitness landscape where the solution to the problem is given by the whole population or at least by a set of individuals, and not anymore by a single individual. Individuals do not encode a complete solution but a part of a solution, the solution to the problem being then built from several individuals that "collaborate."

This approach is to be related to the spirit of co-evolution: a population is a "society" that builds in common the solution that is search for, but on the contrary to co-evolution, the species are not specifically identified and separated. Of course the design of such algorithms becomes more complex than for a direct –standard– EA approach, and the diversity of the population is a crucial factor in the success of a Parisian approach. Moreover, splitting the problem into interconnected sub-

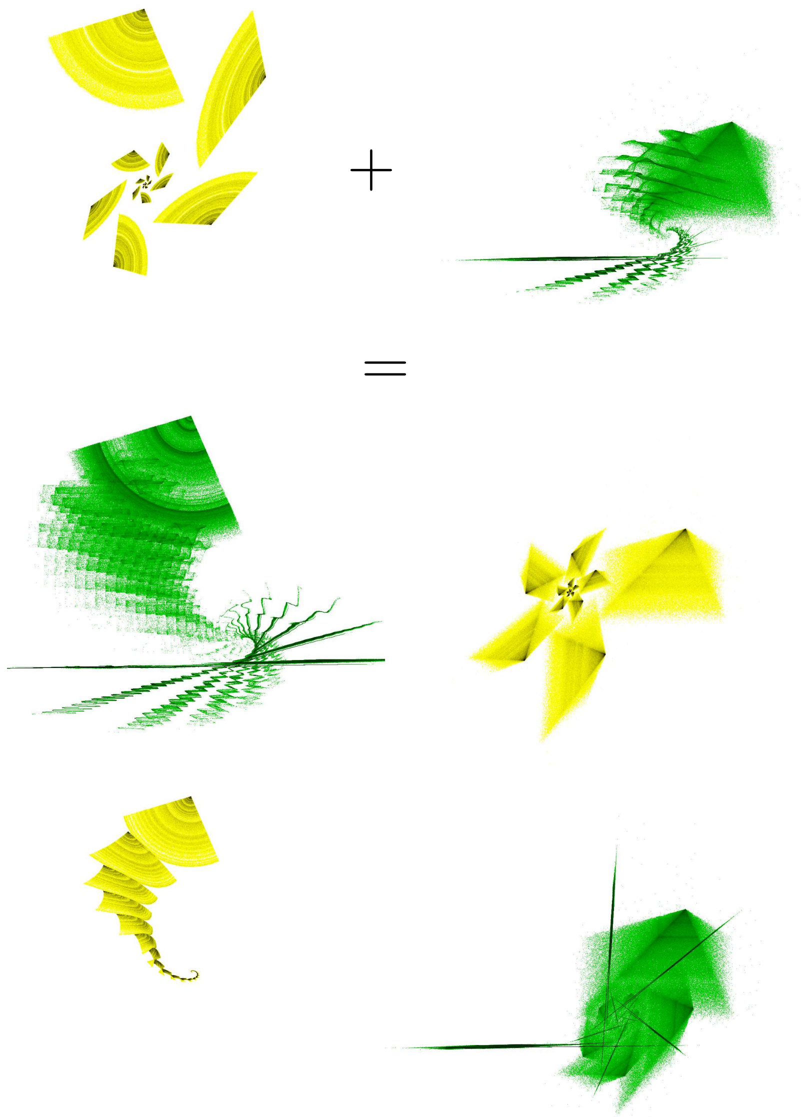

Fig. 6. Four possible crossover offspring (bottom) from 2 parents (top)

problems is not always possible. However, when it is possible to do so, the benefit is great: a Parisian approach limits the computational waste that occurs in classical EA implementations, when at the end of the evolution, the whole final population is dumped except the best individual only. Experiences and theoretical developments have proved that the EA gains more information about its environment than the only knowledge of the position of the global optimum. The Parisian approach tries to use this important feature of EAs.

A Parisian EA may have all the usual components of an EA, plus the following additional ones:

- *two* fitness functions: a "global" one that is calculated on the whole population or on a major portion of it (after a clustering process or elimination of the very bad individuals, for example), and a "local" one for each individual, that measures how much this individual contributes to the global solution.
- a distribution process at each generation that shares the global fitness on the individuals that contributed to the solution,
- a diversity mechanism, in order to avoid degenerated solutions where all individuals are concentrated on the same area of the search space.

Developing a Parisian EA for interactive creative design tools is based on observation of the creative process. Creation cannot be reduced to an optimisation process: artists or creative people usually do not have precisely in mind what they are looking for. Their aim may fluctuate and they sometimes gradually build their work from an exploration. "User satisfaction" is a very peculiar quantity, very difficult to measure, and to embed in a fitness function. This is the reason why ArtiE-Fract has been equipped with a Parisian approach mode that can be activated at any time during the run of the system using a translation module between classical and Parisian populations.

For instance, ArtiE-Fract propososes a Parisian mode to evolve individuals made of functions : several functions are necessary to build an image (= an IFS attractor). The Parisian engine thus builds the set of images presented to the user from k randomly aggregated indiduals of the population. The user evaluation is then shared between functions that were involved in each image (distribution of global fitness), and a local fitness component is derived from the contractivity ratio of each function (a function that is too contractive may yield less interesting features). This scheme is particularly useful for introducing new components combinations in an evolved population, or when the user has the feeling to get stuck in a local optimum during his search. A Parisian approach can thus be used here like a functions mixing tool, or even like what we could call a "lateral thinking" tool, as it helps the user to get a feeling about the graphical role of some functions.

3. Example 1: how Emmanuel Cayla uses ArtiE-Fract

Emmanuel Cayla's first approach to ArtiE-Fract is precisely this flexible access to fractals. Actually the artist advocates that a new intimacy is built between mathematics and painting as it happened during the "Renaissance" with geometry, proportion and perspective. This new relationship doesn't imply only fractals but those are certainly meant to play a key role in this reunion.

To start his exploration of the fractals' universe on the ArtiE-Fract software, the painter first decided to set the graphical parameters of a set functions and to stick with those. He had indeed the feeling that it is too difficult to properly browse such a search domain without defining a static reference framework. The first of these decisions was to put colours aside and to work only with black ink on a white background. The second parameter had to do with noise, also known as "grain" or "distorsion" whose level was uniquely set over the whole initial population.

Indeed, the "noisify" operator of ArtiE-Fract, is an important component of this artistic approach. This operator adds a random noise to the value returned by the function during the drawing of the attractor image, see figure 7. This is an important factor of visibility of scattered attractors, as it conditions the thickness of the simulated "paintbrush."

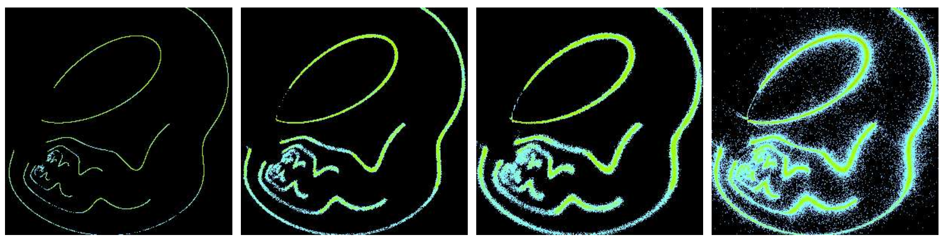

Fig. 7. Original IFS and 3 mutated IFS with various noise levels.

With these two settings, the painter produced the first generations of images having approximately (as it is however produced by a random process, even if it is strongly limited in this particular case) the desired characteristics.

The third of these *a priori* constraints parameters is the image format, in other words the graphical proportion and size of the generated images. For the moment, the default setting (square images) has been used. This point has however been considered as a limitation by the artist, and has been integrated in a new version of the software.

Experience shows that extraction of individuals that are interesting from the pictural standpoint is indeed a directed process. All the individuals that are acknowledged as "picturally effective" by the artist, i.e. all the individuals he identifies as matching his artistic visions, are the outcome of a process based on selection and gradual construction. Selection of individuals during the successive generations is a bit more sophisticated of a process than a simple "good"/"bad" categorisation. The painter is familiar with images and the criteria used for conservation and "reproduction" of individuals in the creation process of ArtiE-Fract are going to be based on observations such as: "This individual is interesting for the feeling of movement it produces, I will mate it with that one whose use of blacks and blurs is appealing; I will also add this other one, it is a bit less interesting but the other two could greatly benefit from the way it plays with lines..." As one can see, the artist indeed plays the role of constructor. ArtiE-Fract might bring a great deal of randomness in the process but painters have always worked with randomness and taken advantage

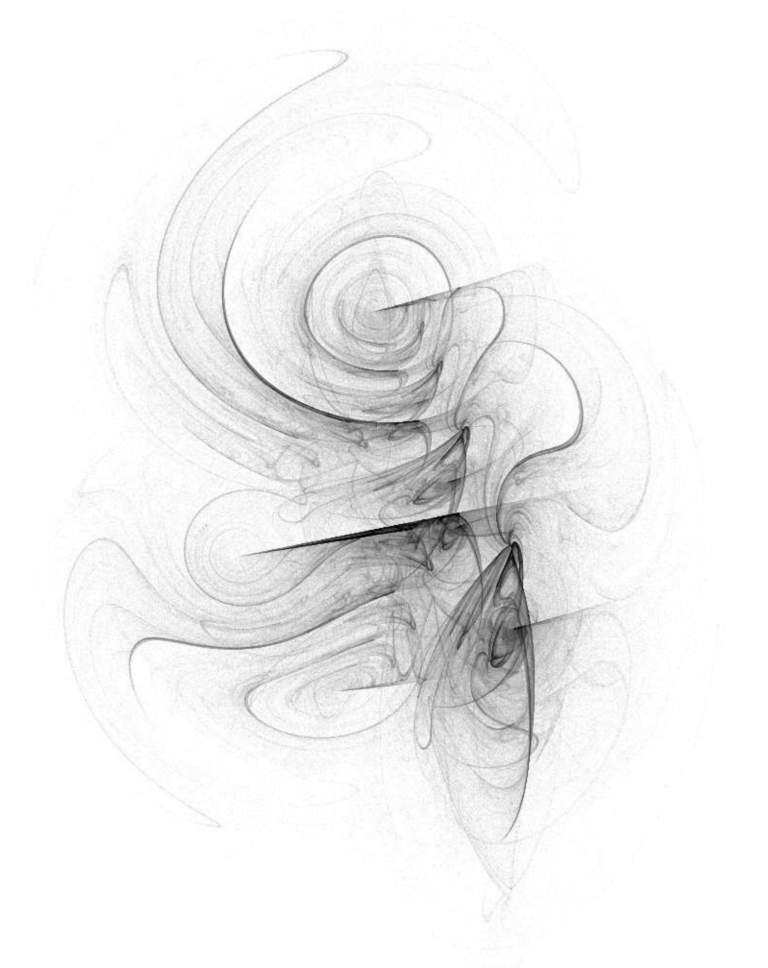

Fig. 8. DANCING WOMAN *(Danseuse)* – Emmanuel Cayla

of "artistically interesting incidents."

Another interesting aspect is the tuning of the various "usual" genetic parameters, for instance, population size. It appeared pointless for the painter to work with generations rich of hundreds or thousands of individuals for the five following reasons:

- Computation time increases with the number of individuals.
- In any case the population will need sorting.
- Whatever happens, one will remain within the same family of shapes.
- There are enough individuals exhibiting originality, even in smaller sized populations.
- One is dealing with infinite spaces and may we work with 20, 40, 100 or

Fig. 9. SWIMMING FOSSILS *(Fossiles nageant)* – Emmanuel Cayla

3000 individuals that we would only encompass an infinitely small space of possibilities with respect to the potential creativity of the system.

What the artist is looking for, before anything else, is expression, more precisely poetic expression. And with ArtiE-Fract, it works. This means that with this software, one is able to browse and discover meaningful shapes like those our mind enjoys flying over: lakes, mountains piercing the clouds, forgotten cities, trees taking over old walls, people standing on icebergs near the North Pole... And it is essentially here, although not only, that this approach of numeric arts moves us.

Fig. 10. HARSH HORIZON *(Horizon violent)* – Emmanuel Cayla

4. Example 2: how Marie-Amelie Porcher uses ArtiE-Fract

Marie-Amelie Porcher is a young designer, who is currently studying at a french art school (Ecole des Beaux arts de Rennes), and who is familiar with computer image design tools. She become rapidly efficient in using ArtiE-Fract for her own creative work, and focussed on the creation of repetitive textures. Texture creation tools have thus been added to the software, as well as other image post-processing tools she was asking for. We are now currently working on the integration of these additionnal features into ArtiE-Fract's genomes.

Figures 14 to 17 shows some of her creations. She was very satisfied with the tool and has brilliantly proved that this software is in adequation to the demand of the new generation of digital image designers.

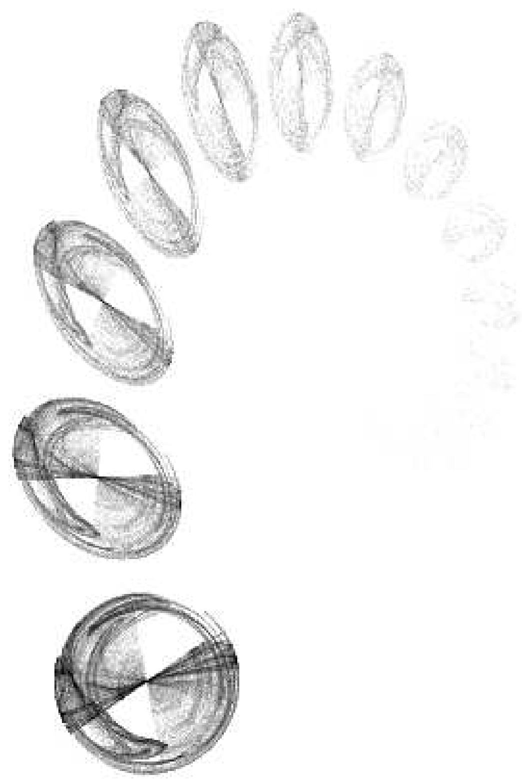

Fig. 11. FUNNY CIRCLES *(Drôles de cercles)* – Emmanuel Cayla

5. Conclusions and Future developments

Of course this fruitful collaboration has deeply influenced the design of many components of ArtiE-Fract, as an artist has sometimes a completely different viewpoint on the software tools.

For example, besides the specific way he selects and notates the images, strong control tools were considered as crucial. Direct interactions were thus designed[b], such as simply "killing" an individual and controlling the "reproduction-elimination" step of the evolutionary process. Artists also stressed on the fact that the visual evaluation of a picture is strongly dependent on the surrounding images

[b]A "strongly controlled evolution mode" is now available in ArtiE-Fract.

Fig. 12. Fox *(Renard)* – Emmanuel Cayla

and background, point that was neglected by the ArtiE-Fract developers so far.

Another point has to do with the creation of repetitive sequences such as borders: The "modulo," "mirror" and "symmetries" effects have been for instance programmed to produce shapes that can be continuously juxtaposed, as the one produced my Marie-Amelie Porcher.

Until now, Emmanuel Cayla based his work mainly on the "global evolution" tools of ArtiE-Fract, i.e. the Parisian evolution modes were used only as harsh exploration tools, to open new research directions at some stages, when the population becomes too uniform, for instance. He however produced unexpected shapes, in comparison to what was produced before by inexperienced designers. His use of black and white was also noticeable (figures 8 to 13). It stressed on the importance

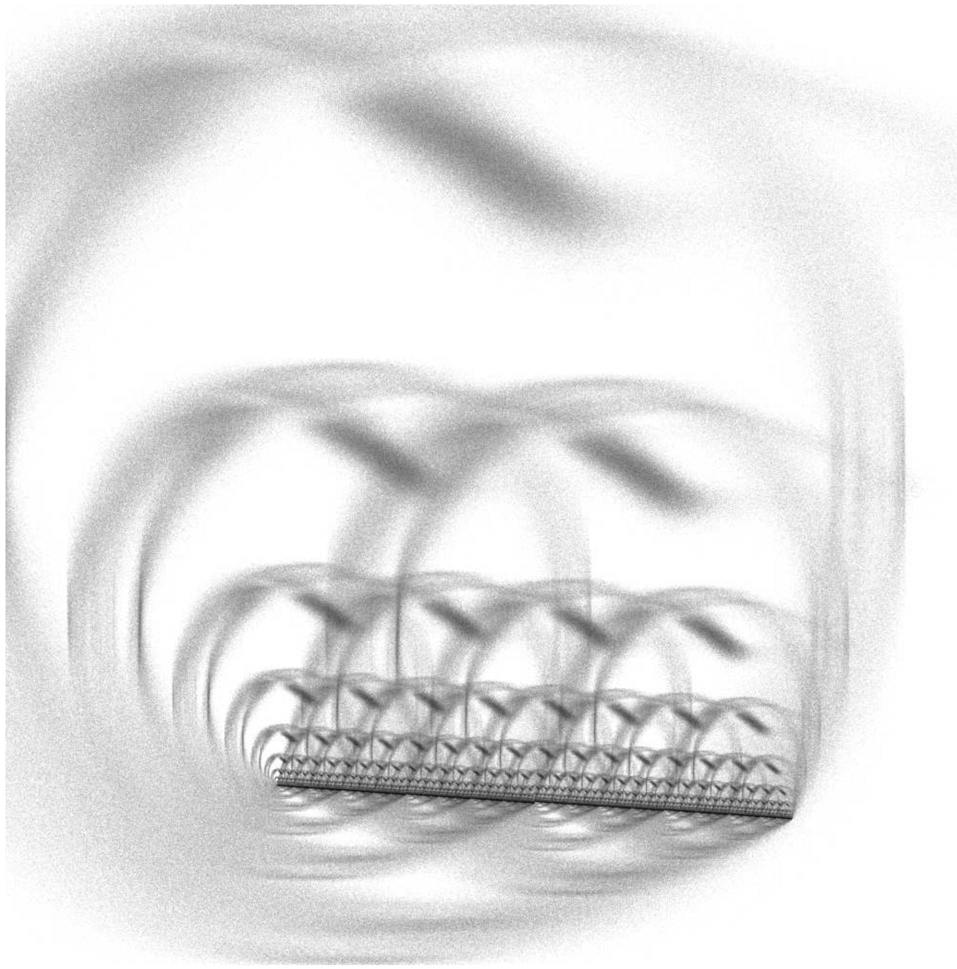

Fig. 13. WATER WHEEL *(Roue d'eau)* – Emmanuel Cayla

of predefined simple evolution modes:

- that concentrate the search on some specific aspects of the design, such as "evolution of color only," "evolution of shape only,"
- that tune the degree of randomness in the evolution process, like "strong control" or "weak control."

To sum up, interactions with the "real life" of art and design has led us to integrate many additionnal features to ArtiE-Fract. Among them, the "noise" component (in functions encoding and in post-processing) is currently raising theoretical questions about the convergence of IFS involving such functions. Image limitation tools (repetitive images, non-square images), predefined tunings for the EA engine

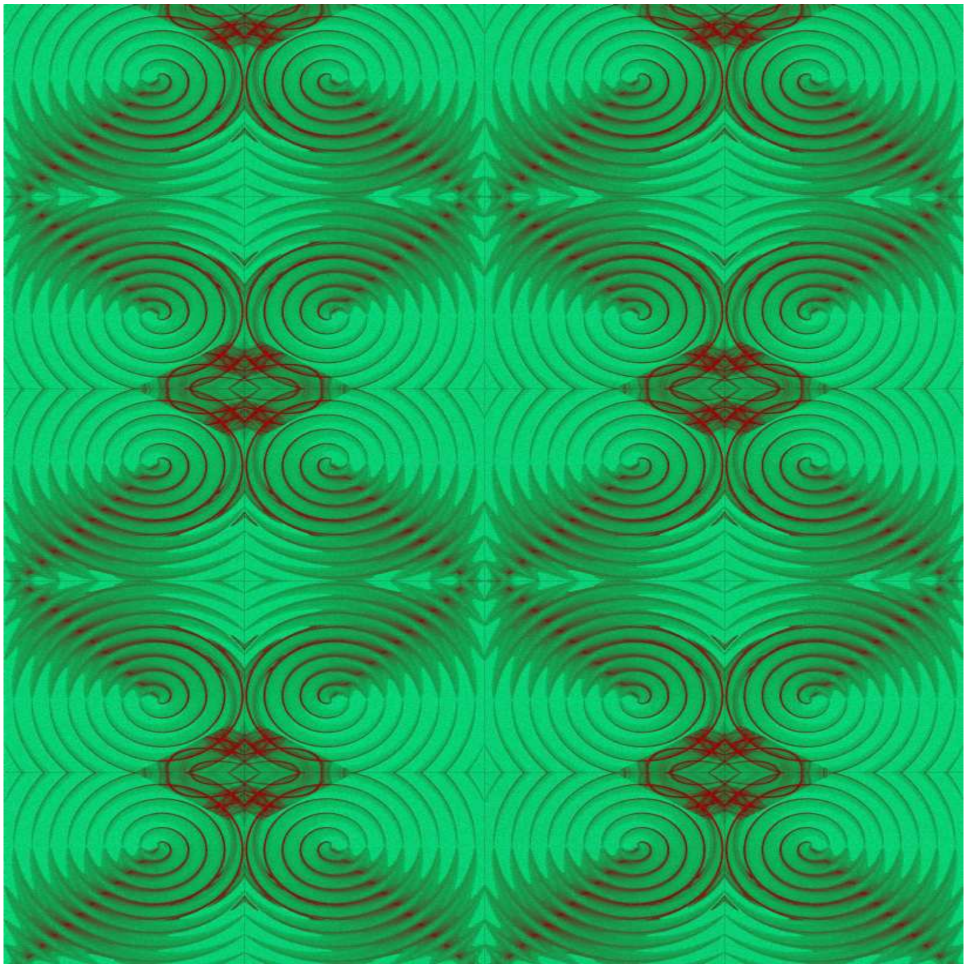

Fig. 14. Celtic arabesques *(Arabesques cetliques)* – Marie-Amelie Porcher

(including the translation of populations between Parisian and non-Parisian mode), and image post-processing tools have also an important impact on application capabilities of the system.

Figure 18, shows an example of textile design outputs. As Emmanuel Cayla says, “ArtiE-Fract is a tool that should help us come up with new shapes in the world of artistic drawing.”

Acknowledgments

The author is grateful to Emmanuel Cayla and Marie-Amelie Porcher, for their patient and enthousiastic contribution to this work, to Jonathan Chapuis, who programmed most of the ArtiE-Fract software, and to Yann Semet for his kind help

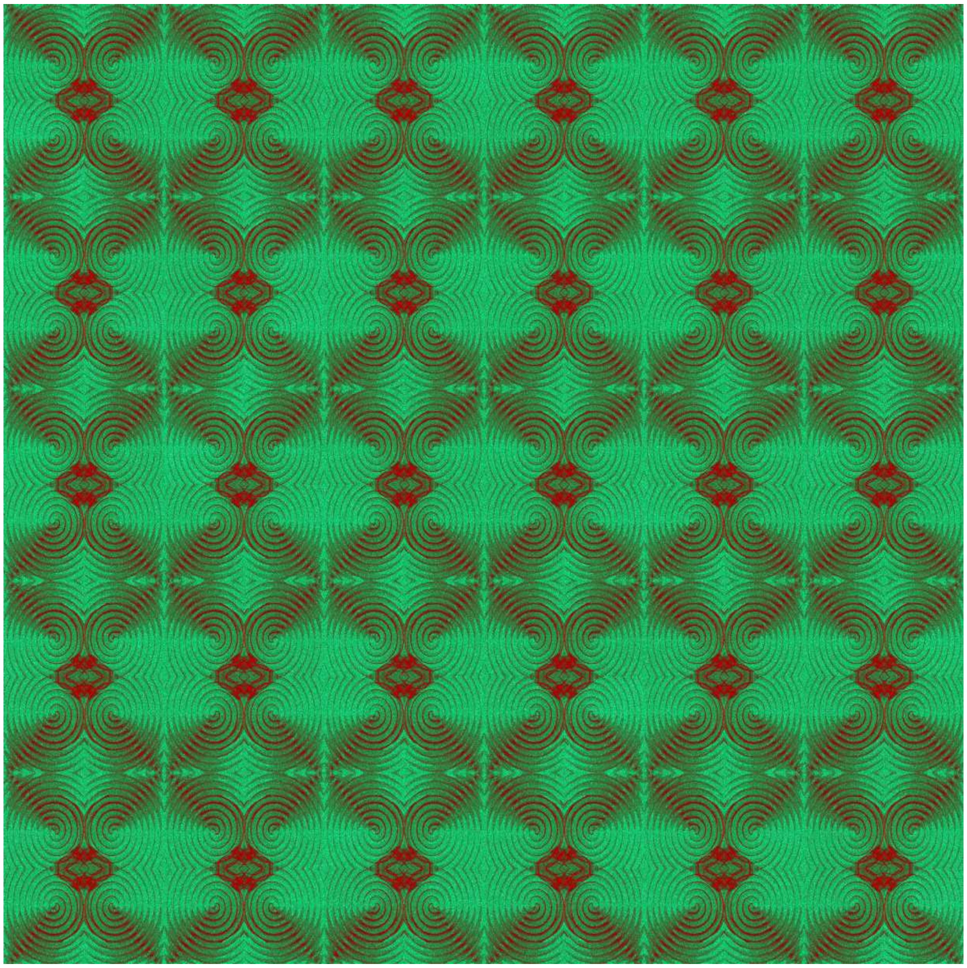

Fig. 15. Celtic arabesques - smaller motives *(Arabesques cetliques)* – Marie-Amelie Porcher

during the writing of the first version of this paper.

References

1. P. J. Angeline. Evolving fractal movies. In *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)*, pages 503–511, 1996.
2. W. Banzhaf. *Handbook of Evolutionary Computation*, chapter Interactive Evolution. Oxford University Press, 1997.
3. M. Barnsley and S. Demko. Iterated function system and the global construction of fractals. *Proceedings of the Royal Society*, A 399:243–245, 1985.
4. M. F. Barnsley. *Fractals Everywhere*. Academic Press,N Y, 1988.
5. J. Chapuis and E. Lutton. Artie-fract: Interactive evolution of fractals. In *4th International Conference on Generative Art*, Milano, Italy, December 12-14 2001.

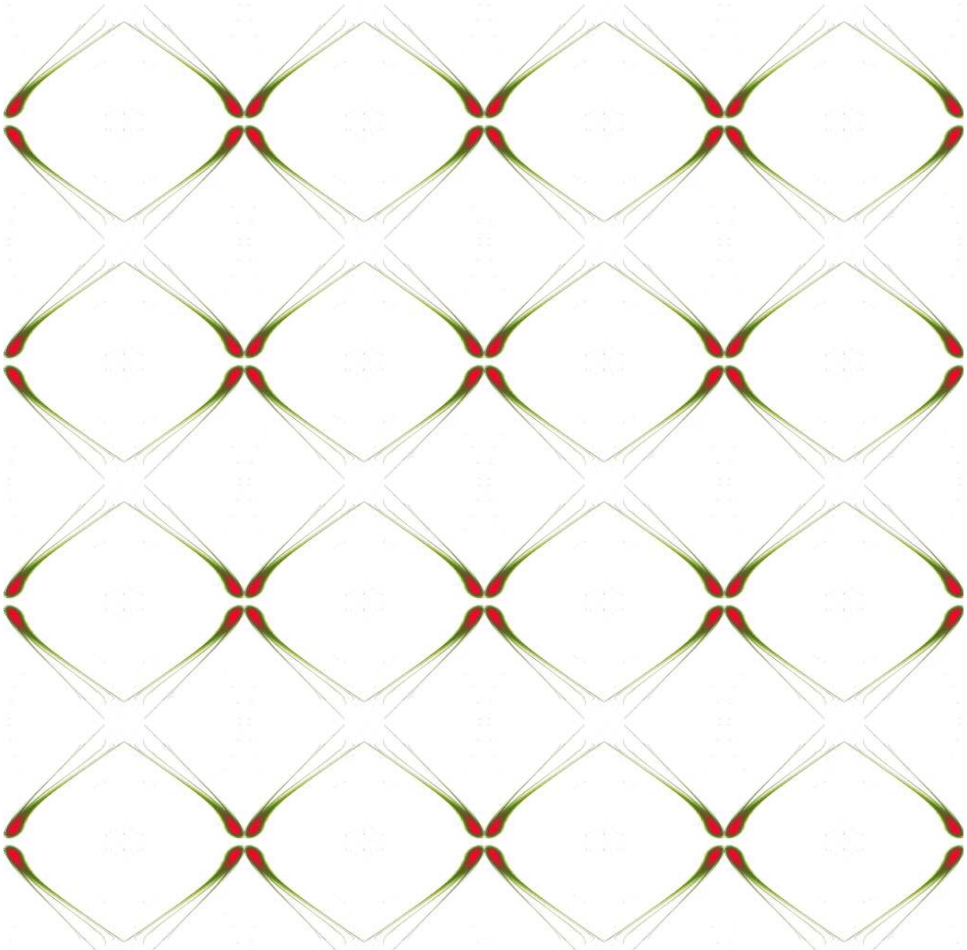

Fig. 16. Losange matches *(Allumettes en losange)* – Marie-Amelie Porcher

6. P. Collet, E. Lutton, F. Raynal, and M. Schoenauer. Polar ifs + parisian genetic programming = efficient ifs inverse problem solving. *Genetic Programming and Evolvable Machines Journal*, 1(4):339–361, 2000. October.
7. K. Daoudi, E. Lutton, and J. Levy Vehel. Fractal modeling of speech signals. In *Fractals in Engineering*, 1994. 1-4 June, Montreal.
8. B. Forte, F. Mendivil, and E. R. Vrscay. "chaos games" for iterated function systems with grey level maps. *SIAM Journal on Mathematical Analysis*, 29(4):878–890, 1998.
9. D. A. Goldberg. *Genetic Algorithms in Search, Optimization, 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):1451–1465, 1993.

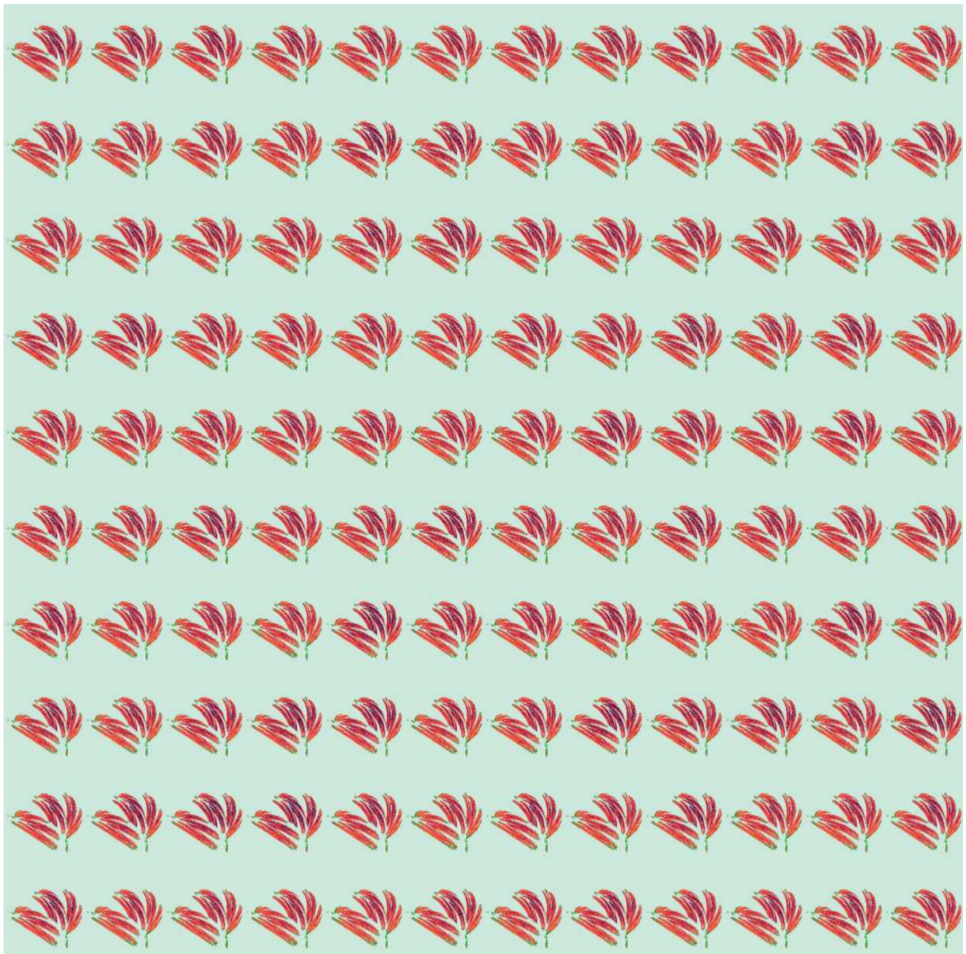

Fig. 17. Wall paper *(Papier peint)* – Marie-Amelie Porcher

12. S. Kamohara, H. Takagi, and T. Takeda. Control rule acquisition for an arm wrestling robot. In *IEEE Int. Conf. on System, Man and Cybernetics (SMC'97)*, volume 5, Orlando, FL, USA, 1997.
13. B. B. Mandelbrot. *The Fractal Geometry Of Nature*. W.H.Freeman and company, 1977.
14. N. Monmarche, G. Nocent, G. Venturini, and P. Santini. *Artificial Evolution, European Conference, AE 99, Dunkerque, France, November 1999, Selected papers,*, volume Lecture Notes in Computer Science 1829, chapter On Generating HTML Style Sheets with an Interactive Genetic Algorithm Based on Gene Frequencies. Springer Verlag, 1999.
15. R. Poli and S. Cagnoni. Genetic programming with user-driven selection: Experiments on the evolution of algorithms for image enhancement. In *2nd Annual Conf. on Genetic Programming*, 1997.

Fig. 18. Two scarves prototypes created by Emmanuel Cayla with ArtiE-Fract. (`http://cetoine.com`)

16. F. Raynal, E. Lutton, P. Collet, and M. Schoenauer. Manipulation of non-linear ifs attractors using genetic programming. In *CEC99, Congress on Evolutionary Computation, July 6-9, Washington DC. USA.*, 1999.
17. S. Rooke. The evolutionary art of steven rooke. http://www.azstarnet.com/∼srooke/.
18. H. Rowley. Interactive mutation of IFS Fractals. http://www.cs.cmu.edu/∼har/GeneticArt.html
19. D. Saupe and R. Hamzaoui. A bibliography for fractal image compression, 1996.
20. K. Sims. Artificial evolution for computer graphics. *Computer Graphics*, 25(4):319–328, July 1991.
21. H. Takagi. Interactive evolutionary computation: System optimisation based on human subjective evaluation. In *IEEE Int. Conf. on Intelligent Engineering Systems (INES'98)*, Vienna, Austria, Sept 17-19 1998.
22. H. Takagi and M. Ohsaki. Iec-based hearing aids fitting. In *IEEE Int. Conf. on System, Man and Cybernetics (SMC'99)*, volume 3, Tokyo, Japan, Oct. 12-15 1999.
23. S.J.P. Todd and W. Latham. *Evolutionary Art and Computers*. Academic Press, 1992.
24. J. Ventrella, Tweaks. http://www.ventrella.com/Tweaks/FractalStudio/index.html
25. J. Lévy Véhel and E. Lutton (Eds). *Fractals in Engineering: New Trends in Theory and Applications*. Springer Verlag, 2005. ISBN-10: 1846280478.
26. I. Yoshiaki, Interactive Evolution of Genetic Fractals. http://www.asahi-net.or.jp/∼hq8y-ishm/gp.html http://www.bekkoame.ne.jp/∼ishmnn/gallery/gp-image2.html
27. Some related web sites:
http://www.genarts.com/karl
http://www.accad.ohio-state.edu/∼mlewis/AED/Faces/
http://draves.org/flame/