

MODELING AND COMPRESSING GENERALIZED IMAGES USING ITERATED FUNCTION SYSTEMS (IFS) WITH A GENETIC ALGORITHM ON A NEURAL-NET-IFS SCHEME

Olympia Lilly Bakalis and Bryan J. Travis

Los Alamos National Laboratory, Los Alamos, NM 87545. olb@cnls.lanl.gov

ABSTRACT

We investigate the use of Iterated Function Systems (IFS) for modeling and compressing 2 dimensional fractal images by exploring solutions of the Inverse IFS Problem: Given a fractal image we are looking for parameters in 24 dimensions for a small set of affine maps and their associated probabilities which constitute the IFS. Upon iteration the IFS solution produces an attractor with the characteristics which describe the image under consideration. We use a Genetic Algorithm (GA) and a Neural Network (NN) scheme which simulates the IFS. A sample cross section of the error hypersurface within the "Mandelbrot set" in the parameter space of a 3-map IFS family is shown. The solution obtained with the GA, in the 18 or 24 dimensional parameter space, meet the desired specifications which describe the original image to within the given discretization. Applications of the inverse problem are aimed towards prediction of second order phase transitions, where scale invariance and power laws are encountered, as well as towards image compression.

1. INTRODUCTION

An abundance of natural systems exhibit complex behavior [1], [2] which often results in the formation of spatial or even temporal structures which cannot be described by static Euclidean geometry. Examples are objects whose fragmentation is preserved at all scales such that a small piece of them is structurally similar to the whole. Trees, clouds, fractures, and coastlines are among objects depicting spatial scale-free structure. Mandelbrot named these objects, whose dimension is most often *fractional* rather than integral, *Fractals*, and the geometry necessary for their description *Fractal Geometry* [3]. As he stated [4], *Fractal geometry is a workable new middle ground between the excessive geometric order of Euclid and the geometric chaos of general mathematics*. During critical phenomena, in addition to scale invariance, systems, however diverse from each other, also incorporate temporal scale-free behavior, while their macroscopic quantities exhibit power laws. Scale invariance and power laws are intrinsic to fractal geometry. The underlying mechanism responsible for such pattern formation is governed by nonlinear dynamics. Consequently, fractals are *attractors* of dynamical systems.

Due to the fact that there exist physically dissimilar systems which have similar properties, especially in critical phenomena, in order to characterize and study their phase space attractors, we are prompted to construct artificial models whose attractors belong in the same category. Such modeling may be applicable to *prediction* of continuous phase transitions. The model is therefore a mathematical tool which is a dynamical system in itself. An Iterated Function System (denoted as IFS) is an example of such a tool. As sources of deterministic fractals, IFSs were extensively studied by Michael Barnsley and his colleagues, and were used for image compression [5], [6].

An Iterated Function System consists of a specified number of appropriately chosen functions or transformations (linear or nonlinear) which as set operate iteratively on their own output in a metric space. An *IFS with probabilities* may be constructed when a probability-weight is associated with each of the functions. The IFS produces an orbit which converges within an object of fractal dimension. The functions used in this work are contractive affine maps acting in the Euclidean plane. A probability is associated with each of them. The object produced, namely the *attractor* of the prescribed IFS, is an image whose fractal density is distributed on top of an underlying fractal shape, i.e., its *support*.

In order to use IFS for modeling, the solution to the *Inverse Problem for IFS* becomes the subject of attention, for the answer to the following question is given by it. Given some fractal pattern corresponding to a complex physical system, can we find a simple IFS whose attractor incorporates properties and characteristics of the pattern in question? For this purpose, the inverse problem for IFS may be defined as finding parameters for a small number of affine maps and their associated probabilities.

1.1. The IFS

All terms, definitions, and theorems concerning Iterated Function Systems, used herein, may be found in the book *Fractals Everywhere* by Barnsley [5]. As mentioned earlier, our IFS consists of N affine transformations operating on shapes in the Euclidean plane. Under affine transformations, several properties of shapes are preserved: straight lines remain straight lines; parallel lines remain parallel; the ratio of volumes, areas and line segments is preserved. Thus, ellipses transform into ellipses, parabolas into parabolas, etc. In \mathbb{R}^2 , an affine map is representable by an 2×2

matrix together with a shift:

$$w \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} e \\ f \end{pmatrix}. \quad (1)$$

In our test case, the image to be modeled is a discretized attractor μ^* of an IFS. μ^* represents the density of visited points on its support \mathcal{A}^* (all the raised pixels). Using Barnsley's *collage theorem* we assume that we can find at least one IFS of contractive affine maps whose attractor \mathcal{A} is close in Hausdorff distance to \mathcal{A}^* , and whose density distribution μ (coded in color) is close in Hutchinson distance to μ^* ¹. The density of points on the attractor is distributed proportionally to the probability associated with each affine map [5]. The probabilities are adjusted *implicitly* to be proportional to the normalized areas of their associated transformation.

We explore solutions to the IFS inverse problem with the use of a Genetic Algorithm (GA), in the *forward iteration method* as defined in [5]. Forward iteration guarantees that every iteration on the attractor with the IFS which generated it, will map it onto itself with probability 1. A significant number of iterations (an average of 20 for images embedded in 128×128 grid) is necessary for determining whether a chosen parameter set actually causes the orbit of the IFS to converge on an attractor whose Hausdorff distance from \mathcal{A}^* is sufficiently small.

We are searching for solutions in the parameter space of the IFS, the hypercube $[-1, +1]^{6N}$, where each point represents an IFS attractor. We are interested in the neighborhood of the boundary of the Mandelbrot set (that is, the set of stable periodic points of the dynamical system) associated with our parametrized IFS family. The test images, embedded in the unit square, covered two categories of IFS attractors:

1. *Just Touching*, with test image the "fern" attractor of Figure 4(A).
2. *Minimally Overlapping*, with test image the "leaf" attractor of Figure 5(A).

1.2. The GA

We chose to use a GA in order to search for the $6N$ affine parameters of an IFS whose attractor is an acceptable solution, for the following reasons: First, it has been observed [7] that GAs consistently outperform other methods of stochastic search on problems involving discontinuous, noisy, high-dimensional, and multimodal objective functions, as is the case of the Inverse IFS problem. This is demonstrated in Figure 1: A projection of the Mandelbrot set in the 18-D parameter space of the 3-map IFS family is plotted (top). The projection was purposefully chosen to partly cover the niche containing the point "fern" (white regions). In the bottom section, the error landscape within the inner rectangle is plotted. Here, however, we observe that this projectional slice misses the "fern" minimum (0.067, 0.51) (due to the introduced randomness) because the largest peak is not located exactly on it. We conclude

¹The Hausdorff distance measures "nearness" between shapes of two images. The Hutchinson distance measures "nearness" between density distributions residing on top of those shapes.

that an automated search will be quite complicated and for that we prefer to represent the IFS with a binary string.

Genetic Algorithms were developed by Holland in the 1960s [8], [9] after the paradigm of natural evolution. Detailed description of GAs may be found in the books by Goldberg [10] and Michalewicz [11]. In brief, each generation of species is represented with a time step of an iterative procedure. The GA maintains a population P of n individuals s_i , $i = 1, \dots, n$ which, during each generation, undergo genetic operations and are evaluated according to their fitness in their environment thus producing a new population of candidate solutions to an objective (fitness) function f .

$$P(t) = \{s_1(t); s_2(t); \dots; s_n(t)\} \quad (2)$$

Each individual s_i is formed by a chromosomal binary string of length 1 in which the vector of parameters of the function $f(x)$ is encoded. A Simple GA procedure is shown in Figure 2. The genetic operator of **crossover**, applied with a probability **C.rate**, produces individuals representing new points in the search space. Additional variation in the genotype is achieved with the operator of **mutation**. Under mutation, a gene's value (bit) is altered with a very small probability **M.rate**, thus simulating a random walk through the string space. The fittest individual of the last population will be considered as the acceptable approximate solution to the objective function f .

2. ENCODING THE IFS ON A GA

Each IFS-chromosome consists of a binary string. We assigned 10 bits per parameter, of which one bit was reserved for the sign, to ensure an accuracy of 0.005. Encoding 6 parameters per affine transformation and up to N affines per IFS ($N = 4$), yield a string length $l = 60 \cdot 4 = 240$ bits.

2.1. Initializing the Population

In the real world the only available data is a discretized image which is not necessarily an IFS attractor. We therefore do not know of any particular niche in the parameter space where an IFS attractor (point) might resemble its characteristics. For an automated search, we must randomly choose a vast number of points from the entire hypercube $P = [-1, +1]^{24}$. To bypass this problem, we select *vital* points as follows: While randomly scanning the parameter hypercube, we construct a large number of chromosomal strings which loosely satisfy constraints derived from the Collage Theorem. For that to hold, the area generated after one iteration on \mathcal{A}^* should cover it almost exactly. We therefore require that each affine transformation should be contractive and map the image mostly within itself in one iteration. The individual affine maps should be arranged within \mathcal{A}^* such, that there is only minimal overlap between them. The IFS chromosome, then, consists of N such maps stacked in a single string. To reduce initialization time, we approximated the area of convex attractors with the area enclosed by only eight extremities. We then imposed the constraint that after being mapped once, the eight extremities should lie within the approximated area. The "gene-pool", described further below, is also formed by this method. Next, assuming that the area of each affine

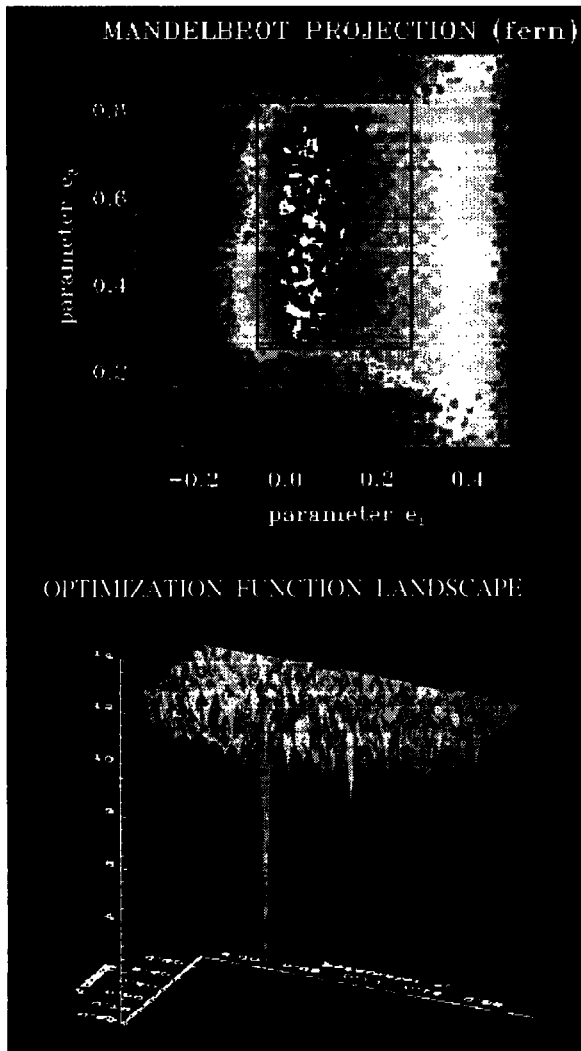


Figure 1: Projection of the Mandelbrot set (top) and error-landscape (bottom) for the 3-affine IFS family near the fern minimum. The slice is generated by randomly deviating two of the 18 parameters within the slice $[-0.01, 0.01]$, while all other 16 parameters remain unaltered at their original fern values. Solutions exist only in the white regions (Top).

```

t ← 0;
initialize P(t);
evaluate P(t);
while (not finished) do
    t ← t + 1;
    select P(t) from P(t - 1);
    operate on P(t);
    evaluate P(t);
end;

```

Figure 2: Sketch of a Simple GA procedure.

map is proportional to the determinant of the matrix [12], we required that each determinant is smaller than unity while their total sum nearly equals one, so that the following holds:

$$\text{Area of } \mathcal{A} = \sum_{k=1}^N (|\det \mathbf{a}_k| \cdot (\text{Area of } \mathcal{A})). \quad (3)$$

where \mathbf{a}_k is the linear part of the 2-D affine transformation $w_k(\mathbf{x}) = \mathbf{a}_k \mathbf{x} + \mathbf{b}$.

The population size n represents the number of sampling points in parameter space and is usually dependent on the chromosomal length l . For large l , n it can be of the order of thousands, resulting in slow evaluation. An immense population size would defeat the purpose of GAs who are meant to introduce new sampling points. A tiny one, on the other hand, could force the GA into premature convergence on the first “good enough” local minimum. The required size also depends on how diverse and fit the initial points are. To ensure a diverse selection of them, we introduced a *Gene Pool*. That consists of a very large number of individuals (sampling points) which satisfy the aforementioned constraints. The first n members of the pool constitute the initial population. Unacceptable chromosomes on every third generation are replaced with randomly chosen ones from the Gene Pool. This method permits the use of small enough populations, typically of 100 – 500 members, which minimizes computational time, while preventing premature convergence.

2.2. IFS-GA Genetic Operators

The fractal depends continuously on the IFS parameters. A flip of a sign on one of the parameters, however, which represents a jump in parameter space, may result in some minor alteration of the given image by causing, for example, a reflection on one of the transforms. Despite visual similarities of such attractors, the error between them may be quite large, while their surrounding neighborhoods may possibly constitute viable niches in the fitness landscape. The GA is capable of reaching such distant points in a single step with the crossover or with the mutation operators. The population size, and the crossover and mutation probabilities were adjusted empirically, and the values used for obtaining the minima of Figures 4(B) and 5(B) are summarized in Table 1.

Table 1: Genetic Algorithm parameters

Image	n	l	C.rate	M.rate
fern 4(B)	500	240	0.75	0.00067
leaf 5(B)	500	240	0.75	0.00067

3. THE GA SEARCH

The search aims towards recovering the fractal properties of the invariant support \mathcal{A}^* and the \mathbf{p} -balanced measure μ^* of

the IFS. The value of the invariant total measure is recovered in both categories of the test images. *Just-touching* IFS attractors (as is the fern) are more sensitive to the discretization than the *overlapping* IFS. It is therefore more difficult to recover the fractal dimension. For attractors of *overlapping* IFS (as is the leaf) the GA has a much better performance in approximately identifying intersecting areas and optimizing the total density distribution [Figure 4(B)].

3.1. Definition of symbols

- N^* : Total number of raised pixels in the test image.
- N : Total number of raised pixels in the constructed image.
- N_0 : Total number of raised pixels after 1 iteration on the test image with the IFS N_0 is used to verify the Collage theorem.
- μ^* : Total invariant measure of the test image.
- μ : Total invariant measure of the constructed image.
- s : Contractivity factor.
- not_raised**: Number of pixels in the test image that are not raised.
- plot_out**: Number of raised pixels in the constructed image that do not belong to the test image.
- coverage**: Fraction of the test image that is covered by the constructed image.

3.2. Objective (Fitness) Function

The objective function both specifies the features to be looked for in terms of their errors, and guides the algorithm towards an acceptable solution. We formulated a general function to accommodate both classes of images within the unit square. The exact location of the image inside the square is to be found. Flipping the sign of one of the parameters moves the search in a separate niche, which in general implies the existence of discontinuities. To ensure diversity in the chromosomal structure we maintained the seemingly uninteresting regions of the error landscape, as high plateaus, by imposing penalties.

Our fitness function [12] consists of the sum of the errors:

1. V_1 = The Hausdorff distance.
2. V_2 = The Hutchinson distance and the difference in the total invariant measure.
3. V_3 = The pixel overlap together with the difference in the total number of raised pixels between the test image and the constructed image.

The error landscape is smoothed with the sigmoidal filter (Eq. 4) whose constants were empirically chosen in the first few (< 10) evaluations. The three errors adjust relative to each other during each evaluation, thus preventing dominance of one of them, a primary cause of premature convergence. The overall form of the objective function is:

$$\text{ERROR} = \text{ERROR}_1 + \text{ERROR}_2 + \text{ERROR}_3 =$$

$$= \sum_{i=1}^3 \frac{\sqrt[3]{V_i}}{1 + \exp[\delta_i \cdot (V_{0i} - V_i)]} \quad (4)$$

The function's robustness is verified in the vicinity of the (3-affine IFS – no stem) fern parameters with the gradient descent (**powell** [13]) method. The initial condition is shown in Figure 3(A). Smoothing the landscape had a tremendous effect, as shown in Figure 3(C) which depicts the minimum where **powell** arrived and which is very close to the global [Figure 3(B)], (see Table 2).²

Table 2: The gradient descent algorithm **powell** performed remarkably well in minimizing our objective function, when starting from the Initial condition of Figure 3(A).

	true fern 3 affines	initial condition (IC)	powell on IC
N	1680	534	1695
N_0	1680	1428	1632
not_raised	0	1185	261
plot_out	0	39	276
coverage	100%	44.72%	84.09%
μ	20.54	13.26	20.54
D_f	1.638	1.629	1.638
s	0.850	0.75	0.854

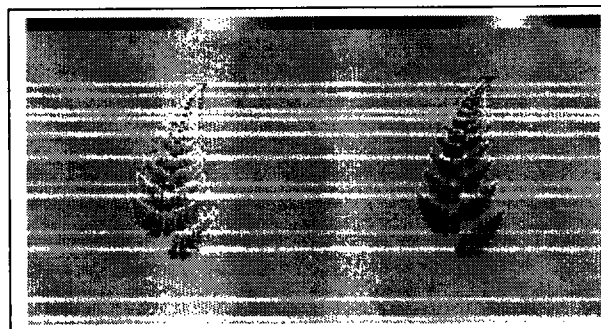
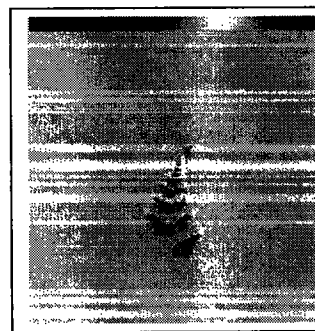


Figure 3: (A)–Top: Initial Condition for **powell**. (B)–Left: 3-affine (no stem) fern test image. (C)–Right: **powell** minimum.

²The IFS parameters for all the images may be found in [12].

Table 3: Summarized results for the fern attractor: The GA solution found on the hypercube $[-1.0, 1.0]^{24}$ of the parameter space P , overall recovers some fractal characteristics of the test image more accurately than the minimum in the neighborhood of the true fern, found with Powell's method (despite the apparent similarity of the patterns of Figure 3).

	true fern 4 affines	GA + powell on hypercube $[-1.0, 1.0]^{24}$
N	1723	1723
N_0	1723	1663
not_raised	0	255
plot_out	0	195
coverage	100%	86.71%
μ	20.70	20.70
D_f	1.638	1.920
s	0.850	0.652

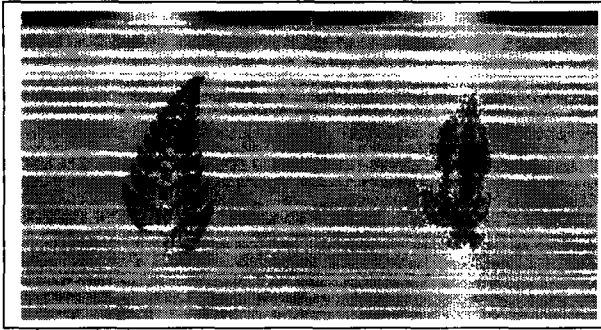


Figure 4: *Just Touching* case: (A)–Left: 4-affine fern test image. (B)–Right: GA minimum (initial points are randomly selected). These images, apparently dissimilar, share similar characteristics (Table 3) and coordinates.

3.3. The GA's search in the parameter hypercube $P = [-1.0, 1.0]^{24}$

3.3.1. *Just-touching IFS (fern)*

For *just-touching* IFS, a very fine grid should be used in order to preserve the intricate fine structure. Doubling the resolution increases the computation time 8-fold: each magnification is four times larger than the previous, and the iterations necessary for convergence double. Although parallel computing would be most appropriate, we performed all experiments serially. A grid of 128^2 pixels comprises the smallest size in which the most important characteristics of the image are still preserved. Many of the points of the fern in this size grid appear to be touching, which means that the GA will be looking for an overlapping IFS, and we do not expect the fractal dimension to be recovered in this class of images. In general, the fractal dimension is determined by the slope in the ratios

$$\frac{\log(\text{number of raised pixels})}{\log(1/\text{size of a pixel})}$$

calculated at various magnifications. This ratio on the 128^2 grid, however, is found exactly:

$$\begin{aligned} \frac{\log(N^*)}{\log(1/\text{pixel.size})} &= \frac{\log(1680)}{\log(128)} = 1.530 \\ \frac{\log(N)}{\log(1/\text{pixel.size})} &= \frac{\log(1663)}{\log(128)} = 1.528 \end{aligned} \quad (5)$$

From the above numbers and from the summarized results of Table 3 we conclude that the GA performed quite well on the 128×128 grid for $N = 4$ affines. But we also conclude that coarse grids may convey misleading information for *just-touching* IFS attractors. The fittest IFS chromosome, found by the Genetic Algorithm, produced the attractor shown in Figure 4(B).

Table 4: Summarized results for the leaf attractor: The IFS found by the GA on the hypercube $[-1, 1]^{24}$ of the parameter space P , satisfies the criteria for solution, and also approximates the mass distribution.

	true leaf	GA + powell on hypercube $[-1.0, 1.0]^{24}$
N	3563	3588
Collage N_0	3563	3569
not_raised	0	350
plot_out	0	375
coverage	100%	89.86%
μ	980.07	979.72
D_f	1.947	1.942
s	0.60	0.61



Figure 5: *Minimally Overlapping* case: (A)–Left: 4-affine leaf test image. (B)–Right: GA minimum (initial points are randomly selected). These images, apparently dissimilar, share similar characteristics (Table 4) and coordinates.

3.3.2. *Minimally Overlapping IFS (leaf)*

The *minimally-overlapping* leaf attractor is easier to solve than the *just-touching* fern attractor, because the overlap of its maps is preserved throughout various magnifications,

thus allowing for a larger selection of potential solutions. The minimum found by the Genetic Algorithm is shown in Figure 5(B).

The contractivity factor for this IFS ($N = 4$) is $s = 0.61$, which is very close to the original $s = 0.60$. Its fractal dimension $D_f = 1.942$, is almost exactly the same as the original $D_f = 1.947$. We note that the GA approximately recovers the density distribution. The summarized results for the leaf attractor are presented in Table 4.

4. APPLICATIONS AND CONCLUSIONS

In order to extract the essential properties which characterize the fractal image under study, we used a GA to find parameters for 4 affine maps explicitly and their associated probabilities implicitly. The properties are incorporated in the errors which constitute the objective function. The GA is found to be a very robust algorithm in the sense that it was easily adaptable to the IFS inverse problem and that it does find solutions as specified by the objective function. Final convergence on the GA minima is achieved with the use of a local optimizer, namely either the gradient descent algorithm `powell` or a *simplex method*, due to Nelder and Mead, namely the `amoeba` subroutine (Press *et al.* [13]). From the two, `powell` is sensitively dependent on the form of the evaluation function but is more accurate than `amoeba`. The fractal characteristics of the images to be modeled, especially in the *minimally overlapping IFS* case, were recovered to within 1% error of the total image-overlap, to less than 1% error of the Hausdorff distance, and to less than 1% error of the total measure of the fractal attractor. The evaluation function was modified with sigmoids so that the landscape of the error hypersurface is smoothed. This resulted in a better performance of the algorithm than the "raw" distance functions used in the literature on both GAs and SAs (Simulated Annealing algorithm) [14], [15].

Our technique aimed mainly towards modeling complicated patterns, associated with complex physical systems, whose structural complexity makes them difficult to study. In particular, the IFS-GA can correlate fractal spatio-temporal structures, found in nature, to simple hierarchical lattices from which the Renormalization Group (RG) transformation can be formulated. From the RG a nonlinear IFS is constructed whose attractor is the Julia set of the critical manifold. The critical exponents are extracted from the Julia set thus revealing the behavior of the macroscopic quantities near the critical region. This information is useful for prediction purposes. In addition, these models become, in essence, classifiers of universality classes for spatio-temporal fractal patterns encountered in criticality.

Finally, compression of generalized images with this method is a two-step process: A 2-dimensional image can be compressed into a graph of a fractal function. The task of the GA then, is to search in a $2N$ -D space, because four of the parameters are linearly related by the function's continuity condition. This time, the number of maps N is variable and can become large. Compression of fractal functions has been studied extensively [16].

5. REFERENCES

- [1] G.A. Cowan, D. Pines and D. Meltzer, editors. *Complexity. Metaphors, Models, and Reality*, Proceedings Volume XIX, Santa Fe Institute Studies in the Sciences of Complexity, p.8, p.492. (Adison-Wesley, 1994).
- [2] D. L. Stein, editor. *Lectures in the Sciences of Complexity*, Lectures Volume I, Santa Fe Institute Studies in the Sciences of Complexity (Adison-Wesley, 1989).
- [3] B. B. Mandelbrot. *The Fractal Geometry of Nature* (Freeman and Company, New York, 1983).
- [4] M. Fleischmann, D. J. Tildesley and R. C. Ball, editors. *Fractals in the Natural Sciences* (Princeton University Press, Princeton, 1990).
- [5] M. F. Barnsley. *Fractals Everywhere* (Academic, Boston, 1993), second edition.
- [6] M. F. Barnsley and L. P. Hurd. *Fractal Image Compression* (A. K. Peters, Wellesley, 1993).
- [7] N. Schraudolph and J. Grefenstette. *A User's Guide to GAUCSD 1.4* (UC San Diego, CA 92093-0114 Technical Report No. CS92-249, 1992).
- [8] J. H. Holland. *Adaptation in Natural and Artificial Systems* (MIT Press, Cambridge, 1992).
- [9] J. H. Holland. *Genetic Algorithms* (Sci. Am., pp. 66-72, July 1992).
- [10] D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning* (Addison-Wesley, Reading, 1989).
- [11] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs* (Springer-Verlag, Berlin, 1992).
- [12] O. L. Bakalis. *Encoding the Invariant Measure of Iterated Affine Transforms with a Genetic Algorithm* (University of New Mexico, Dissertation, 1996).
- [13] W. H. Press, B. R. Flannery, S. A. Teukolsky and W. T. Vetterling. *Numerical Recipes in C* (Cambridge University Press, Cambridge, 1991).
- [14] J. Lévy-Véhel and E. Lutton. *Optimization of Fractal Functions Using Genetic Algorithm* in *Fractals in the Natural and Applied Sciences*, edited by M. M. Novak (North-Holland, Amsterdam, 1994).
- [15] D. J. Nettleton. *Evolutionary Algorithms in Artificial Intelligence: A Comparative Study Through Applications* (University of Durham, Dissertation, 1994).
- [16] D. S. Mazel and M.H. Hayes, *Using Iterated Function Systems to Model Discrete Sequences*. IEEE Trans. on Signal Processing **40** (7), pp. 1724-1734 (1992).