Evolutionary multifractal signal/image denoising

Evelyne Lutton, Jacques Lévy Véhel

12 Evolutionary multifractal signal/image denoising

Evelyne Lutton and Jacques Lévy Véhel

This chapter investigates the use of Evolutionary techniques for multifractal signal/image denoising. Two strategies are considered: using evolution as a pure stochastic optimiser, or using interactive evolution for a meta-optimisation task. Both strategies are complementary as they allow to address different aspects of signal/image denoising.

12.1. Introduction

We deal with enhancement – or denoising – of complex signals, based on the analysis of the local Hölder regularity. Our methods do not make explicit assumptions on the type of noise nor on the global smoothness of original data, but rather supposes that signal enhancement is equivalent to increasing the Hölder regularity at each point. Such methods are well adapted to the case where the signal to be recovered is itself very irregular, e.g. nowhere differentiable with rapidly varying local regularity.

We describe two techniques. The first one tries to find a signal close to the observations and such that its *local Hölder function* is prescribed. A pure optimisation approach is convenient in this case, as this problem does not admit a closed form solution in general (although attempts have been previously done on an analytical basis for simplified cases [17, 19]). In addition, the number of variables involved is huge. Genetic Algorithms have been found to be efficient in this case, and yield better results than other algorithms. The principles and example results are presented in section 12.2.

However, it appears that the question of results evaluation is critical: A precise (and general !) definition of what good denoising – or enhancement – is, is questionable. Medical doctors indeed may have different opinions on the quality of a given result, as well as remote sensing specialists, or art photographers. The perception of quality is extremely dependent on

the end-user, the context and the type of application. A simple signal-to-noise-ratio (when computable) is certainly not able to capture the subtle perceptive judgment of a human end-user.

To investigate this issue, we describe another regularity-based enhancement technique: Multifractal Bayesian denoising acts by finding a signal close to the observations and such that its *multifractal spectrum* is prescribed. This method relies on the tuning of a small set of parameters that are able to provide various improvements of the observed noisy image. An interactive evolutionary approach has been designed in order to cope with the meta-optimisation problem of tuning the parameters set, and is described in section 12.7.

In order to get acceptable computation times, the underlying optimisation problem and its parameters have been designed to be solved by a deterministic method. The evolutionary approach is used in an interactive way, at a meta-level.

Going further into this direction, a scheme has been designed (and tested!) in order to reduce the number of user interactions, in other words to limit the famous "user fatigue," see section 12.9.

The schemes and tools developed on the signal and image denoising problem can be extended to other image analysis tasks, such as multifractal image segmentation (see section 12.11).

12.2. Signal enhancement/denoising

The problem may be set in the following way: Someone observes a signal Y which is a certain combination $F(X, B)$ of the signal of interest X and a "noise" B. Making various assumptions on the noise, the structure of X and the function F, one then tries to derive a method to obtain an estimate $\hat{X}$ of the original signal which is optimal in some sense. Most commonly, B is assumed to be independent of X, and, in the simplest case, is taken to be white, Gaussian and centred. F usually amounts to convoluting X with a low pass filter and adding the noise. Assumptions on X are related to its regularity, e.g. X is supposed to be piecewise C^n for some $n \geq 1$. Techniques proposed in this setting resort to two domains: functional analysis and statistical theory. In particular, wavelet based approaches, developed in the last ten years, may be considered from both points of view [7, 8].

In this work, we do not make explicit assumptions on the type of noise and the coupling between X and B through F. Furthermore, we are not interested in the global smoothness of X, but rather concentrate on its *local* regularity: Enhancement will be performed by increasing the Hölder function α_Y (see next section for definitions) of the observations. Indeed, it is generally true that the local regularity of the noisy observations is smaller than the one of the original signal, so that, in any case, $\alpha_{\hat{X}}$ should be greater than α_Y. We thus define our estimate $\hat{X}$ to be the signal "closest" to the observations which has the desired Hölder function. Note that since the

Hölder exponent is a local notion, this procedure is naturally adapted for signals which have sudden changes in regularity, like discontinuities. From a broader perspective, such a scheme is appropriate when one tries to recover signals which are highly irregular and for which it is important that the denoising procedure yields the right regularity structure (i.e. preserves the evolution of α_X along the path).

12.3. The local Hölder exponent

We shall measure the local irregularity of signals with the help of the Hölder exponent. To simplify notations, we assume that our signals are nowhere differentiable. Generalisation to other signals require technicalities which are unessential to our purposes.

Let $\alpha \in (0,1)$, $\Omega \subset \mathbf{R}$. One says that $f \in C_l^{\alpha}(\Omega)$ if:

$$\exists\, C : \forall x, y \in \Omega : \frac{|f(x) - f(y)|}{|x-y|^{\alpha}} \leq C$$

Let: $\alpha_l(f, x_0, \rho) = \sup\{\alpha : f \in C_l^{\alpha}(B(x_0, \rho))\}$. Note that $\alpha_l(f, x_0, \rho)$ is non increasing as a function of ρ.
We are now in position to give the definition of the local Hölder exponent :

Definition 1. *Let f be a continuous function. The* local Hölder exponent *of f at x_0 is the number $\alpha_l(f, x_0) = \lim_{\rho \to 0} \alpha_l(f, x_0, \rho)$.*

Since α_l is defined at each point, we may associate to f the function $x \to \alpha_l(x)$ which measures the evolution of its regularity.

This regularity characterization is widely used in fractal analysis because it has direct interpretations both mathematically and in applications. For instance, the computation of the Hölder exponent at each point of an image allows to perform edge detection[16].

12.4. Signal enhancement based on increasing the local Hölder function

Let X denote the original signal and Y the degraded observations. We seek a regularized version $\hat{X}$ of Y that meets the following constraints: a) $\hat{X}$ is close to Y in the L^2 sense, b) the (local) Hölder function of $\hat{X}$ is prescribed.

If α_X is known, we choose $\alpha_{\hat{X}} = \alpha_X$. In some situations, α_X is not known but can be estimated from Y. Otherwise, we just set $\alpha_{\hat{X}} = \alpha_Y + \delta$, where δ is a user-defined positive function, so that the regularity of $\hat{X}$ will be everywhere larger than the one of the observations. We must solve two problems in order to obtain $\hat{X}$. First, we need a procedure that estimates the local Hölder function of a signal from discrete observations. Second, we need to be able to manipulate the data so as to impose a specific regularity.

We will use a wavelet based procedure for estimating and controlling the Hölder function. We let $\{\psi_{j,k}\}_{j,k}$ be an orthonormal wavelet basis,

where as usual j represents scale and k position. Denote $c_{j,k}$ the wavelet coefficient of X. Results in [11] and [10] indicate that, assuming that ψ is regular enough and has sufficiently many vanishing moments, one may estimate $\alpha_X(t)$ by linear regression of $\log(|c_{j,k}|)$ w.r.t. to the scale j (log denotes base 2 logarithm), considering those indices (j,k) such that the support of $\psi_{j,k}$ is centred above t: Roughly speaking, those coefficients should decay in scale as $2^{-j(\alpha+1/2)}$ (more precisely, all the $|c_{j,k}|$ are bounded by $C2^{-j(\alpha+1/2)}$ for some $C > 0$, and some of the coefficients $|c_{j,k}|$ are of the order of $C2^{-j(\alpha+1/2)}$).

The use of wavelets then allows to perform the reconstruction in a simple way: Starting from the coefficient $(d_{j,k})$ of the observations, we shall define a procedure that modifies them to obtain coefficients $(c_{j,k})$ that fulfil the decay condition with the desired α, and then reconstruct $\hat{X}$ from those $(c_{j,k})$.

We may now reformulate our problem as follows: For a given set of observations $Y = (Y_1, \ldots, Y_{2^n})$ and a target Hölder function α , find $\hat{X}$ such that $||\hat{X} - Y||_{L^2}$ is minimum and the regression of the logarithm of the wavelet coefficients of $\hat{X}$ above any point i w.r.t. scale is $-(\alpha(i) + \frac{1}{2})$. Note that we must adjust the wavelet coefficients in a global way. Indeed, each coefficient at scale j subsumes information about roughly 2^{n-j} points. Thus we cannot consider each point i sequentially and modify the wavelet coefficients above it to obtain the right regularity, because point $i+1$, which shares many coefficients with i, requires different modifications. The right way to control the regularity is to write the regression constraints simultaneously for all points. This yields a system which is linear in the logarithm of the coefficients:

$$\Delta L = A$$

where Δ is a $(2^n, 2^{n+1} - 1)$ matrix of rank 2^n, and

$$\begin{aligned} L &= (\log|c_{1,1}|, \log|c_{2,1}|, \log|c_{2,2}|, \ldots \log|c_{n,2^n}|), \\ A &= -\frac{n(n-1)(n+1)}{12}\left(\alpha(1) + \frac{1}{2}, \ldots, \alpha(2^n) + \frac{1}{2}\right) \end{aligned}$$

Since we use an orthonormal wavelet basis, the requirements on the $(c_{j,k})$ may finally be written as:

$$\text{minimize } \sum_{j,k_n} (d_{j,k} - c_{j,k})^2 \text{ subject to: } \forall\, i = 1, \ldots, 2^n,$$
$$\sum_{j=1} s_j \log(|c_{j,E((i-1)2^{j+1-n})}|) = -M_n(\alpha(i) + \frac{1}{2}) \qquad (12.1)$$

where $E(x)$ denotes the integer part of x and the coefficients $s_j = j - \frac{n+1}{2}$, $M_n = \frac{n(n-1)(n+1)}{12}$ and equation (2) are deduced from the requirement that

the linear regression of the wavelet coefficients of $\hat{X}$ above position i should $-(\alpha(i) + \frac{1}{2})$.

Finding the global solution to the above program is a difficult task; in particular, it is not possible to find a closed form formula for the $c_{j,k}$. In [19], a method is described, that allows explicit computations under simplifying assumptions. In the following, we show how this problem can be addressed with an evolutionary algorithm.

12.5. Evolutionary signal enhancement with EASEA

An evolutionary technique seems to be appropriate for the optimisation problem described in equation (12.1): a large number of variables are involved, and the function to be optimised as well as the constraint are non linear. We describe in this section an implementation based on the EASEA [5] language and compiler.

EASEA (EAsy Specification of Evolutionary Algorithms) is a language dedicated to evolutionary algorithms. Its aim is to relieve the programmer of the task of learning how to use evolutionary libraries and object-oriented programming by using the contents of a user-written `.ez` source file.

EASEA source files only need to contain the "interesting" parts of an evolutionary language, namely the fitness function, specification of the crossover and mutation operators, the initialisation of a genome plus a set of parameters describing the run. With this information, the EASEA compiler creates a complete C++ source file containing function calls to an evolutionary algorithms library (either the GALIB or EO for EASEA v0.6). Therefore, the minimum requirement necessary to write evolutionary algorithms is the capability of creating non-object-oriented functions, specific to the problem which needs to be solved.

In our case, the evolutionary optimisation involved to enhance a signal (1D or 2D) was implemented using a simple structure on which genetic operators were defined. We used GALib [35] as the underlying evolutionary library.

We describe below the implementation for 1D signals. An implementation for image denoising was also produced based on the same principles [23].

The Haar wavelet transform has been used to produce the $d_{j,k}$ associated to the observed signal Y. We also suppose that we know the desired Hölder exponents $\alpha(i)$ (either $\alpha(i) = \alpha_Y(i) + \delta$ where the $\alpha_Y(i)$ are the Hölder exponents of Y and δ is a user defined regularisation factor, or $\alpha(i)$ is set a priori).

Our unknowns will be the multiplicative factors $u_{j,k}$ such that $c_{j,k} = u_{j,k} * d_{j,k}, j \in [0..n-1], k \in [0..2^j - 1]$. As is usual in wavelet denoising, we leave unchanged the first l levels and seek for the remaining $u_{j,k}$ in $[0, 1]$. The genome is made of the $u_{j,k}$ coefficients, for $j \in [l..n-1]$ and

$k \in [0..2^j - 1]$. These coefficients are encoded as a real numbers vector of size SIZE_MAX $= 2^n - 2^l$, which can be written using EASEA syntax as:

```
GenomeClass { double    U[SIZE_MAX]; }
```

The EASEA `Standard functions` sections contain the specific genetic operators, namely:

(1) **The initialisation function:** Each $u_{j,k}$ coefficient is randomly set to a value in $[0, 1]$. Two initial solutions are also put in the initial population: $u_{j,k} = 1.$ and $u_{j,k} = 2^{-k\delta}$.

(2) **The crossover function:** a barycentric crossover has been easily defined as follows : Let `parent1` and `parent2` be the two genomes out of which `child1` and `child2` must be generated, and let `alpha` be a random factor:

```
\GenomeClass::crossover:
  double alpha = (double)random(0.,1.);
  if (&child1) for (int i=0; i<SIZE_MAX; i++)
     child1.U[i] = alpha*parent1.U[i]
                   + (1.-alpha)*parent2.U[i];
  if (&child2) for (int i=0; i<SIZE_MAX; i++)
     child2.U[i] = alpha*parent2.U[i]
                   + (1.-alpha)*parent1.U[i];
\end
```

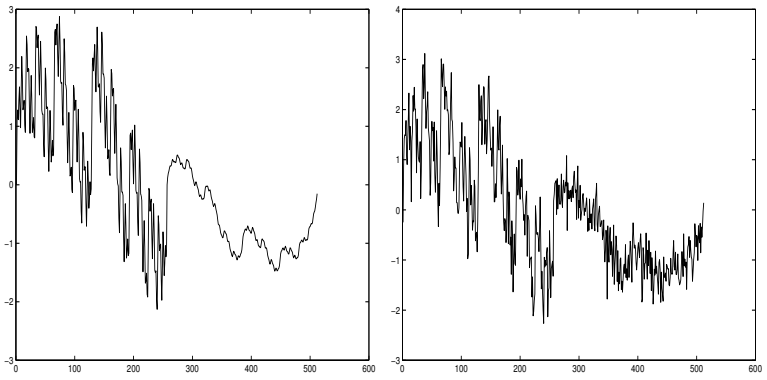
(3) **The mutation function:** Mutation is a random perturbation of radius $SIGMA = 0.01$, applied with probability `PMut` on each gene.

```
\GenomeClass::mutator: // Must return the number of
                       // mutations as an int
  int NbMut=0;
  for (int i=0; i<SIZE_MAX; i++)
     if (tossCoin(PMut)){    NbMut++;
        Genome.U[i]+=SIGMA*(double)random(-1.,1.);
        Genome.U[i] = MIN(1.,Genome.U[i]);
        Genome.U[i] = MAX(0.,Genome.U[i]);}
  if (NbMut==0) identicalGenome=true; // saves evaluation
                                      // time
  return NbMut;
\end
```

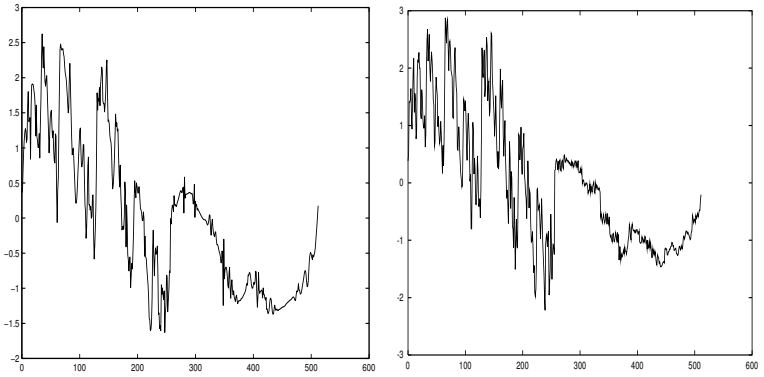
(4) **The evaluation function:** The fitness function has two aims: minimise $\sum((1 - u_{j,k}) * c_{j,k})^2$, making sure constraint (12.1) is satisfied, i.e. the Hölder exponents are the ones we want. The constraint is integrated into the fitness function using a high penalty factor W:

$$Fitness = \sum_{j,k}((1 - u_{j,k}) * c_{j,k})^2 + W * \sum_i |\alpha_u(i) - \alpha(i)|$$

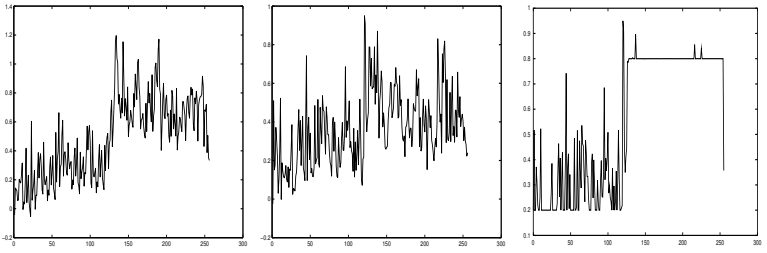
Genome Size	SIZE_MAX = 496
Population Size	25
Number of generations	50000
Computation time	1438.52 seconds for 744897 evaluations



Generalised Weierstrass function (left) + noise (right).



Denoising using wavelet thresholding (left), and using the evolutionary scheme with prescribed $\alpha(t) = Step(0.2, 0.8)$ (right).



Left: Estimated Hölder exponents of the original function (left), of the function + noise (middle) and of the reconstructed function (right).

Figure 12.1. Results on a generalised Weierstrass function $\alpha(t) = Step(0.2, 0.8)$

We use the GALib steady state genetic engine with replacement percentage of 60% and selection by ranking. Crossover and mutation probabilities are fixed respectively to 0.9 and 0.001. Genome size, population size, and number of generations are fixed for each experiment, see section 12.6.

12.6. Numerical Experiments

Results of enhancement on synthetic 1D data are shown in figure 12.1. The original signal is a generalised Weierstrass function [6] with $\alpha_X(t) = 0.2$ for $t < 0.5$, $\alpha_X(t) = 0.8$ for $t > 0.5$ that has been corrupted by additive white Gaussian noise. Figure 12.1 shows the original signal, the noisy one, and the result of the enhancement procedure. For comparison, a denoising using a classical wavelet hard thresholding is also displayed (*i.e.* all coefficients with absolute value smaller than a given threshold are put to 0). For both procedures, the parameters were set so as to obtain the best fit to the known original signal. It is seen that, for such irregular signals, the Hölder regularity based enhancement yields more satisfactory results. One should however remark that we have placed ourselves in a favourable situation for the evolutionary algorithm, since the constraint was to find the (generally unknown) correct Hölder function. Parameters of the evolutionary algorithm are given in figure 12.1.

12.7. Interactive schemes and multifractal Bayesian denoising

There are several ways to improve the method. To begin with, more precise estimations of Hölder exponents yield more accurate results. For instance, in [21], an estimation based on the analysis of local oscillations of the signal has been used. The associated inverse problem is then more complex, and necessitates the use of specific genetic operators. Improved results are obtained at the expense of more complex computations.

Progress on this topic are however more related to the design of an efficient analysis of the quality of the results. But computational measurement of denoising results are difficult to design, as the evaluation of a good denoising is strongly dependent on the end-user as well as the application framework. Signal-to-noise ratio is unable to reflect all the subtle components of a human expert appreciation on a denoising result. Remote sensing, medical imaging, sound restoration, data filtering have very different constraints, and expert users of these domains have different needs.

A way to cope with this discrepancy is to involve the human user in the optimisation loop in order to let him accurately guide the search mechanism towards what he wishes. The artificial evolution framework allows one to introduce human evaluation in the algorithmic loop, and to cope with human judgment irregularity (or even inconsistency). Actually interactive evolution is a research topic that is rapidly growing : first attempts were oriented toward artistic applications [31, 30, 34, 1], but now many

other applications domains are explored: Hearing Aids fitting [33], smooth, human-like, control rules design for a robot arm [13], or design of HTML style sheets [25]. An overview of this vast topic can be found in [32].

Interaction with humans raises several problems mainly related to human fatigue. Three types of solutions have been considered [27, 32, 2] : (1) reduce the size of the population and the number of generations, (2) choose specific models to constrain the research in a priori "interesting" areas of the search space, or (3) perform automatic learning (based on a limited number of characteristic quantities) in order to assist the user and only display the most interesting individuals in the population, with respect to previous votes of the user.

In order to implement (1) and (2), we adopt an approach different from the one in the previous sections: Instead of prescribing the Hölder exponent at each point, we shall rather try to control the multifractal spectrum of the denoised image. This allows to reduce dramatically the number of variables. A first experiment where a small population is evolved using a multifractal scheme is presented in section 12.8. We then experimented an approach integrating item (3), *i.e.* we extend fitness rating to individuals in a larger population *via* the analysis of the user judgment on a small sample of individuals (section 12.9).

12.7.1. Description of the multifractal Bayesian denoising method

The multifractal analysis of a signal consists in measuring its regularity at each sample point, in grouping the points having the same irregularity, and then in estimating the "size" (through some "fractal dimension") of each iso-regularity set. Irregularity is measured via the Hölder exponent

The multifractal spectrum f is a representation of the irregularity of the signal over its definition domain: For each irregularity value, i.e. for each α, one estimates the speed of exponential decay of the probability of finding a point with regularity α as resolution tends to infinity. In some cases, this speed is also the Hausdorff dimension of the corresponding iso-α set (see [3] for details).

For example, for an image, a value of $f(\alpha) \simeq 1$ corresponds to a set of points with same regularity and dimension 1 (*i.e.* it will most of the times look like a set of lines), $f(\alpha) \simeq 0$ is a set of scattered points (singular points), and $f(\alpha) \simeq 2$ is a typically an area of positive measure.

Multifractal analysis is a tool widely used in image and signal analysis, as it provides at the same time a local (α) and a global ($f(\alpha)$) viewpoint on data. It has been exploited with success in many applications where irregularity bears some important informations (image segmentation [15], signal and image denoising [16, 20], etc ...)

The principle of the denoising method is the following: For a noisy image I_1, we search for a denoised image I_2 that satisfies two conditions:

- I_2 has a given multifractal spectrum,

- the probability that the addition of a white Gaussian noise (with variance σ) to I_2 produces the observed image I_1, is maximal.

As we mentioned before, the wavelet transform is a convenient tool for the estimation of the Hölder exponents. This second denoising method is thus also based on the discrete wavelet transform.

12.7.2. The search space is the set of free parameters of the method

We explain here the algorithm for image denoising. Recall that the aim is to find a denoised image I_2 close to the noisy observations I_1, under the constraint that I_2 has a given multifractal spectrum g. The noise is assumed to be white, centred, and Gaussian with variance σ.

The problem may be reformulated as follows: If we denote by y a wavelet coefficient of the noisy image at scale j, then the corresponding wavelet coefficient $\hat{x}$ of the denoised image at the same scale j can be calculated by solving the following equation (for details, see [18]):

$$\hat{x} = \arg\max_{x>0} \left(j \cdot g \left(\frac{\log_2 \left(\hat{K} \cdot x \right)}{-j} \right) - \frac{(|y| - x)^2}{2\sigma^2} \right) \operatorname{sgn}(y)$$

where

- $\hat{K}$ is a constant for which $\hat{\mathrm{K}} \cdot |\mathrm{y}| < 1$ holds and may be set independently for each scale. In what follows, $\hat{K}$ has been taken as the inverse of the maximum coefficient in each scale j.
- g is a function which defines the multifractal spectrum of the denoised image. If we choose to represent it by a linear-by-parts function, its shape is determined by 5 values $\alpha_{\min}$,α_{nod} ,$\alpha_{\max}$, $g(\alpha_{\min})$ and $g(\alpha_{\max})$. More precisely, the spectrum has been chosen to fulfil the following constraints:
  - g is defined on the interval $[\alpha_{\min}, \alpha_{\max}]$,
  - $g(x) \in [0, 1]$,
  - $\alpha_{nod} \in [\alpha_{\min}, \alpha_{\max}]$ and $g(\alpha_{nod}) = 1$,
  - g is affine on $[\alpha_{min}; \alpha_{nod}]$ and on $[\alpha_{nod}; \alpha_{max}]$.

In most cases, but not necessarily, the multifractal spectrum calculated from the denoised coefficients $\hat{x}$ should show a slight spectral shift to the right. This shift is a sign of an overall increase of regularity.

Consequently, we have 7 free parameters:

- the 5 values defining the a priori spectrum g,
- the variance σ of the noise,
- the wavelet used for the discrete (inverse) wavelet transformation.

The choice of the wavelet is less critical than the choice of the other 6 parameters. Usually Daubechies 6 to 12 offer equivalent denoising results in terms of visual reception whereas Daubechies wavelets with smaller supports yield unsatisfactory results in some cases.

Especially in cases where we want to treat very noisy images and subsequently have to set the parameters σ and α_{nod} to relatively high values, the denoising algorithm leads to artefacts in the denoised image when using wavelets with a small support, see figure 12.2. The regularity of those wavelets is low. They are therefore not able to model very irregular parts of an image.

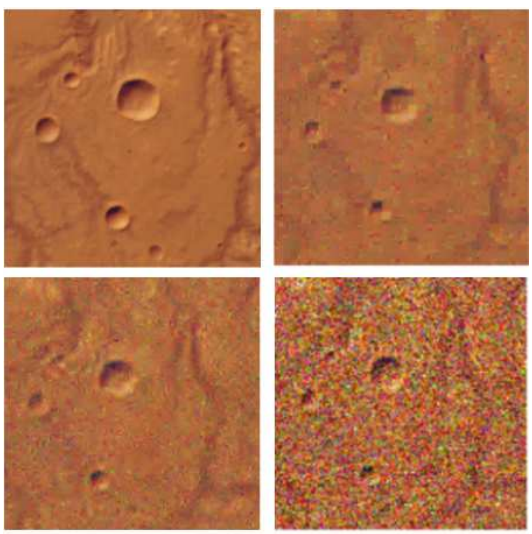


Figure 12.2. **Original Image without noise** (u.l.), Multifractal denoising using wavelet **Daubechies 2** (u.r.), Multifractal denoising using wavelet **Daubechies 18** (l.l.), **Noisy Image** (l.r.), all parameters except wavelets are constant.

It should be mentioned that the number of calculated wavelet scales is fixed to a value obtained from the image dimensions $[N \times M]$:

$$scales = \lfloor \log 2(\max(\mathrm{N}, \mathrm{M})) \rfloor$$

The setup of the 7 resulting free parameters is nontrivial in the sense that they are strongly dependent on the amount of noise in the noisy image and the subjective opinion of the human observer about which result reflects best the desired denoised image.

A solution is therefore to build an interactive evolutionary algorithm (IEA) to interactively find suitable settings of the free parameters.

12.8. An interactive approach with a small population

This first implementation does not include the choice of the wavelet basis as a free parameter but considers a shift to the a priori spectrum g for diagonal wavelet coefficients. It has been noticed that the diagonal wavelet coefficients are more sensitive to additive noise and therefore may need a different spectrum g.

The genomes that are evolved by this IEA are made of 7 real genes:

- 5 values to define the g function for the horizontal and vertical wavelet coefficients: $\alpha_{min} \in [0, 0.5]$, $g(\alpha_{min}) \in [0, 1]$, $\alpha_{nod} > \alpha_{min}$, $\alpha_{nod} \in [0, 2]$, $\alpha_{max} > \alpha_{nod}$, $\alpha_{max} \in [0.01, 20]$, $g(\alpha_{max}) \in [0.2, 1]$
- the shift of the g function for the diagonal coefficients (range $[0, 0.5]$),
- the variance of Gaussian noise, σ (range $[0, 100.0]$).

Fitness and user interaction

The fitness function is given by the user via sliders attached to each denoised sample. Evaluations range from -10 to $+10$. The default value 0 corresponds to a neutral judgment.

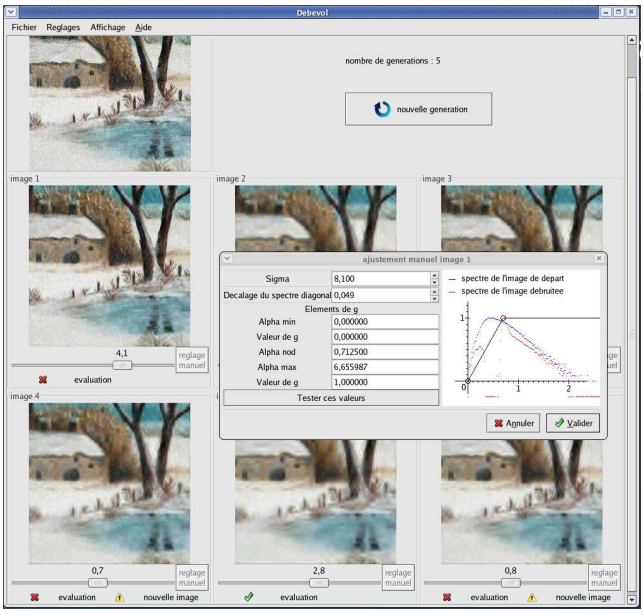


Figure 12.3. The interface of the small population IEA, written in C++.

Additionally, the user may directly edit the genotypes of each image, see figure 12.3, and thus participate in the evolutionary loop as an additional genetic operator.

Genetic Engine

The population has a fixed size of 6 individuals. Each individual carries a set of 7 parameters and therefore represents a potential solution for the aforementioned optimisation problem. All individuals are presented as an image, resulting from the denoising algorithm with corresponding free parameters. The basic evolutionary cycle is presented in figure 12.4, and the operators are the following:

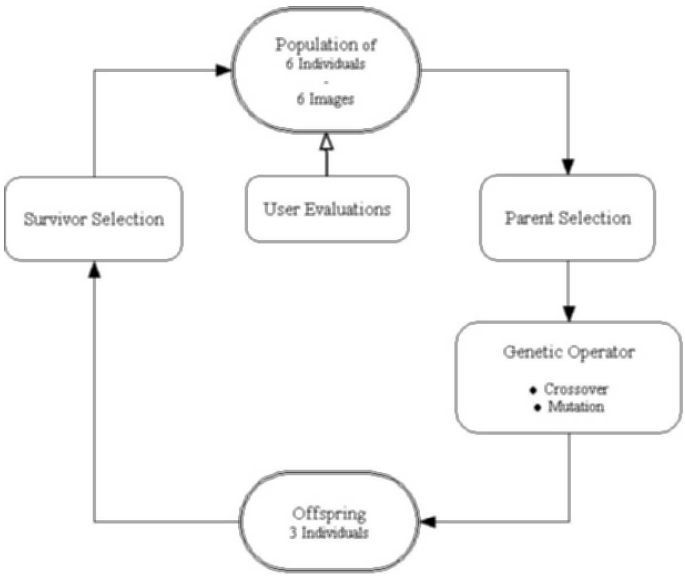


Figure 12.4. The small population denoising IEA genetic engine.

- **Parents Selection** is performed by deterministic selection of the 3 best individuals in the population.
- **Genetic Operators:**
  - **Barycentric crossover** is performed by weighted combination of parents with a randomly chosen weight in $[0, 1]$.
  - **Mutation** is an independent perturbation of each gene value by addition of a Gaussian noise with a given variance.
- **Survivor Selection** scheme replaces the 3 worst parents by offsprings.

A sharing scheme is applied before parent selection: the user marks are weighted to maintain diversity inside the small population. The sharing is based on a genotypic distance. The parent selection then chooses the 3 individuals with the best weighted fitness and is therefore fully deterministic. Crossover and mutation operators then produce 3 children. The survivor selection substitutes parents with offspring and thereby closes the evolutionary cycle.

12.9. An interactive approach with a large population

In the previous interactive scheme, the user has access to 6 individuals (or images) per generation, and the genetic engine only considers the current user evaluations to calculate the next generation. This IEA is driven by a fitness sample – or let us say *a fitness map* – made of only 6 points.

In order to increase the reactivity of the system while being able to handle populations of any size (this increases the search capabilities), we consider the use of a larger fitness sample, while considering techniques to approximate user evaluation. However, a dynamic approximation of the interactive fitness is a delicate task, and necessitates rather large samples. We have proposed a method based on the use of past user marks, collected in a set, the *fitness map*. The fitness of new individuals produced by the genetic engine can be preliminarly estimated from the fitness map by smooth interpolation (flat or polynomial, see below). This preliminary fitness estimation can serve as a preselection tool in order to show to the user only the 6 best individuals of a larger current population.

The use of larger population sizes offers some major advantages, of which an obviously easier maintenance of diversity, a more extensive exploration of the given search space and a possible speedup of convergence are the most significant.

Genome

The genomes are made of 7 genes:

- 5 values to define the g function for the horizontal and vertical wavelet coefficients: $\alpha_{min} \in [0, 0.5]$, $g(\alpha_{min}) \in [0, 1]$, $\alpha_{nod} > \alpha_{min}$, $\alpha_{nod} \in [0, 2]$, $\alpha_{max} > \alpha_{nod}$, $\alpha_{max} \in [0.01, 20]$, $g(\alpha_{max}) \in [0.2, 1]$
- the wavelet used for the discrete wavelet transformation (Daubechies 2 to 20),
- the variance of Gaussian noise, $\sigma \in [0, 100]$.

Fitness and user interaction

The user evaluations are given the same way as in 12.8 with range: $[-6(verybad), ...0(neutral), ... + 6(verygood)]$.

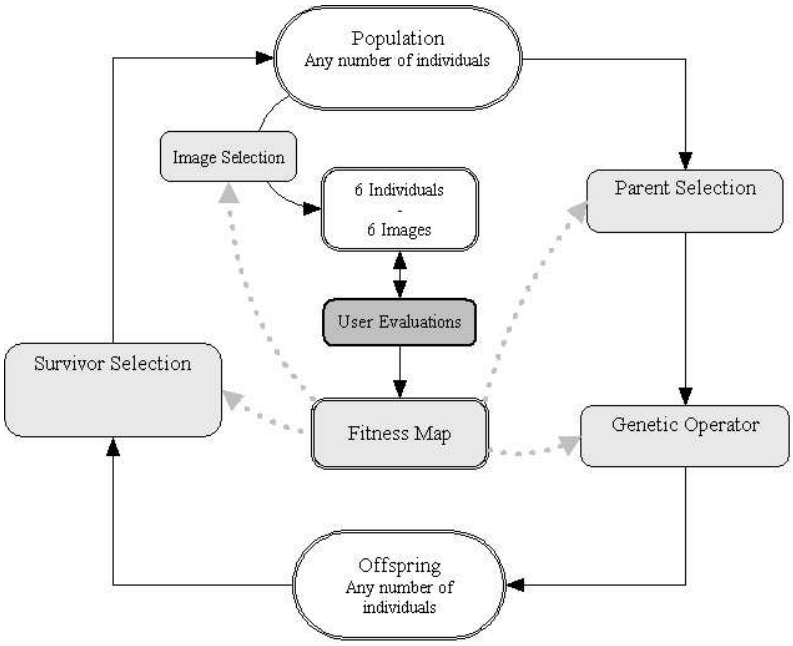


Figure 12.5. **The extended genetic engine** supports a fitness map.

The genetic engine is highly customisable for parameters setting. In contrast to the small population IEA, it is possible that all 6 images in the user interface are changed from a generation to the next. As loosing good images may be frustrating for the user, it is possible to mark images as "Super Individuals" that remain in the user interface and in the population. The user may toggle this state at any time see figure 12.6.

Additionally, to increase user interactivity, two new dialogues have been created: to view and manipulate the individuals in the population (figure 12.7), as well as the samples in the fitness map (figure 12.8). These dialogues both provide plots of the gene values of the individuals in the current population, or in the fitness map. By toggling checkboxes, additional curves, such as an interpolation of fitness and sharing values, are available.

"User ranges" (figure 12.9) have been introduced, as soft thresholds that constrain the search space of genes, and can be set independently for each of the 7 genes.

Genetic Engine

Enlarging the population size and using a fitness map requires major changes: The selection step now strongly depends on the fitness map, as well as crossover and mutation operators. The generation cycle includes an "image

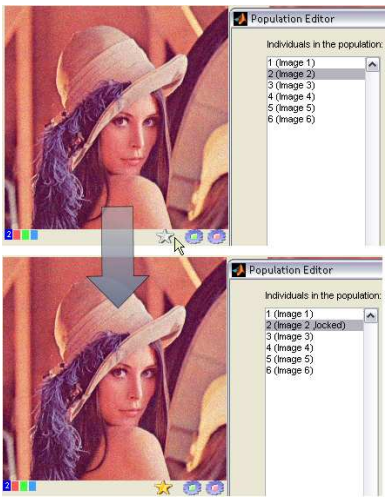


Figure 12.6. Clicking the star button toggles an individual as "Super Individual". From thereon it is "immortal" in the population.

selection" step, i.e. 6 individuals are selected to be shown to the user, see figure 12.5.

The fitness map is a matrix of size [8xN]. N is the number of samples that are saved in the fitness map. These samples are vectors of size [8x1], and include a genotype and its corresponding fitness value. The fitness map is used to interpolate between the available samples in order to predict the fitness values of unknown genotypes. Two interpolation methods have been implemented:

- **"nearest'**: The fitness value of the nearest sample in the fitness map is returned as the fitness value of the unknown sample.
- **"interpolation"**: Interpolating polynomials of order 8 are calculated for each gene using the samples of the fitness map (small stars interpolated by smooth curves in figure 12.10). The approximated fitness value for an unknown sample (see vertical markers in figure 12.10) is the mean value of the 7 polynomials for the genes values of the unknown sample.

Various selection algorithms have been implemented. These selection operators can be deployed by the parent-, offspring- and image selection. The selection operator that is actually used in a certain stage of the genetic cycle is set off-line with help of a configuration file.

The available selection methods are the following:

- **"fittest"**: The individual with the best fitness value is selected.
- **"cycle"**: n individuals are selected by cycling through m individuals among the fittest. This method can be used to generate

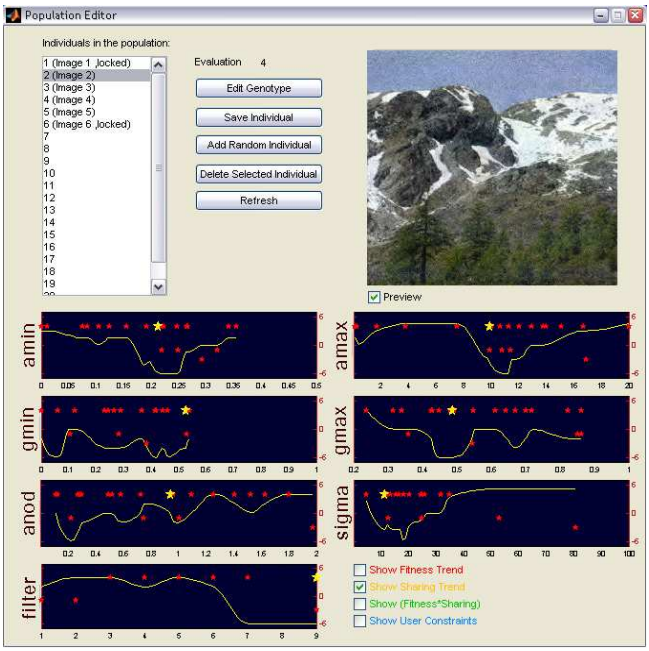


Figure 12.7. **The population editor**. Individuals may be added to the population, existing individuals may be deleted and their genotype can be manipulated. The gene values are plotted on 7 curves as small stars. The plotted curves are interpolations of the fitness (or shared fitness) samples values.

an offspring from a small number of parent individuals (as in the small population IEA).

- **"roulette"**: Randomized variant of fitness-proportionate selection.
- **"rank"**: Randomized variant of rank-proportionate selection. The selection probability for an individual is $pressure^{-rank}$, where "pressure" adjusts the strength of selection and "rank" is the position of the individual inside the population (sorted by decreasing fitness values).

A sharing algorithm has been implemented. Similarly to the sharing algorithm of the small population IEA, fitness values are weighted with a sharing factor that is calculated from mean genotype distances within the population. Genotypes with a high mean distance to other genotypes in the population consequently have a bigger increase in their fitness. The pressure of this sharing method can be set in a configuration file, independently for each selection method. Distinct selection of individuals is also implemented and configurable.

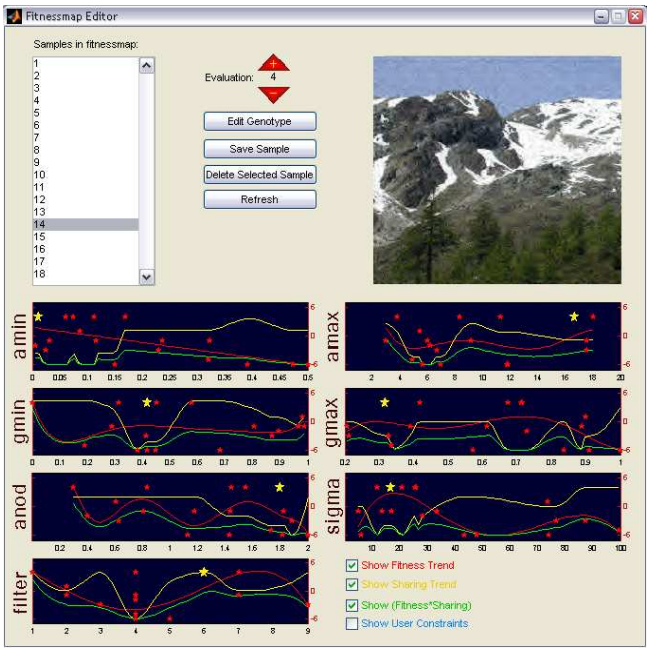


Figure 12.8. **The fitness map editor**. Samples may be deleted and their fitness can be reevaluated.

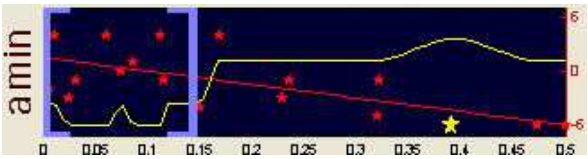


Figure 12.9. Plot of sample fitness for the values of α_{min}. An interpolation of the fitness values is plotted (line). The sharing estimation is plotted as a curve. Setting an **user preferred range** for individual genes is done by drag and drop of the thick vertical brackets.

Different versions of the genetic operators (crossover and mutation) have been implemented:

- **crossover**: **"random":** New individuals are a weighted combination of their parents. The weights are randomly chosen in $[0, 1]$. **"swap":** Special case of random crossover. Parent genes are randomly swapped to generate children genotype. **"factory":** This method builds new genotypes out of the best genes from two parent individuals. The necessary fitness for individual genes is taken from the interpolating polynomials described earlier.

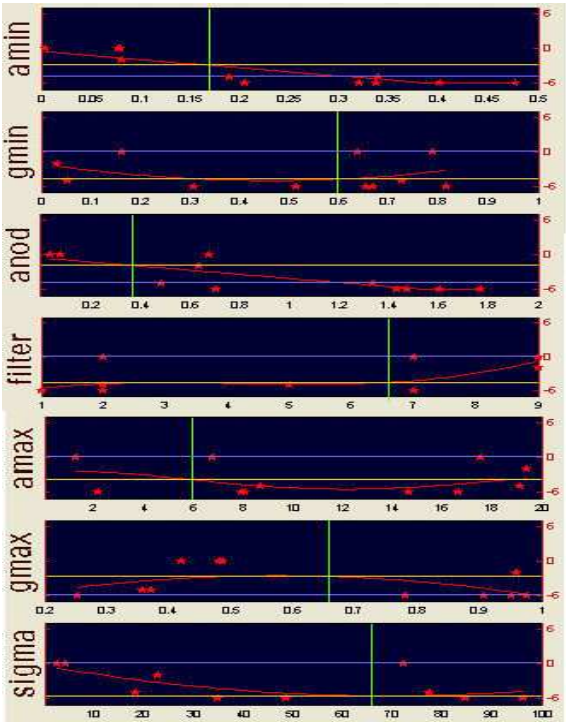


Figure 12.10. The 2 fitness estimation methods illustrated for a sample genotype (vertical markers): **nearest method** (horizontal grey line), **interpolation method** (horizontal light line). The y axes represent the fitness values, while the x axes represent the gene values

- **mutation**: **"random":** Gaussian perturbation of each gene with a given σ. **"preferred area":** Gaussian perturbation of each gene towards it's user range. There is no effect on a gene when it is already located inside the area set by the user.

12.10. Experiments

Quantitative evaluations are rather difficult to perform on interactive evolutionary algorithms. To be able to evaluate the efficiency of the fitness map scheme or, to some extent, compare the small population IEA with the large population IEA, experiments were made in a non-interactive way.

The two algorithms were run on several noisy images, for which the original "non-noisy" images were available, and for various parameter settings.

The non-interactive software

For these tests, the software was slightly modified. User evaluations were replaced by automatic evaluations. A user fitness is therefore imitated by the calculation of a phenotypic distance between the noisy images and their corresponding original images (typically a L2 distance between images). The two presented versions of the IEA were set to run 30 generations on every noisy image and for every parameter setting. In each generation the minimum phenodistance was collected in order to produce a convergence curve. This was repeated for at least 30 times. Afterwards a mean curve of convergence was calculated. The 2 IEAs have been compared on the basis of these average curves.

Parameters

The influence of the population size parameter has been analysed, the parameter setting used for the tests is the following:

- Large population IEA:
  - Population size: 16, 32, 64 and 128 Individuals.
  - Parent selection: Rank selection (as presented in 12.9).
  - Offspring size: 90% of parent generation.
  - Image selection: Fittest selection.
  - Fitness map interpolation: nearest.
  - Use of "Super Individual": in each generation the image with the lowest phenodistance to the original image is set as super individual.
  - *One generation is equivalent to 5 user interactions.*
- Small population IEA:
  - Population size: 6 Individuals
  - Parent selection: Fittest 3,
  - Offspring size: 3 Individuals,
  - *One generation is equivalent to 3 user interactions.*

Results

To ensure a fair comparison between the two algorithms, the average curves of convergence are plotted with respect to the number of user interactions (i.e. user evaluations) instead of the generations number.

Figures 12.11, 12.12 and 12.13 show a clear improvement of the minimisation behaviour for the fitness map scheme, the larger the population, the more efficient.

The loss of precision of the fitness calculation based on the fitness map, which is a very rough approximation of the user – or phenotypic (for the automated version) – fitness, is compensated by the exploration capabilities of a larger population.

This improved exploration capability has also been noticed in a qualitative manner by users.

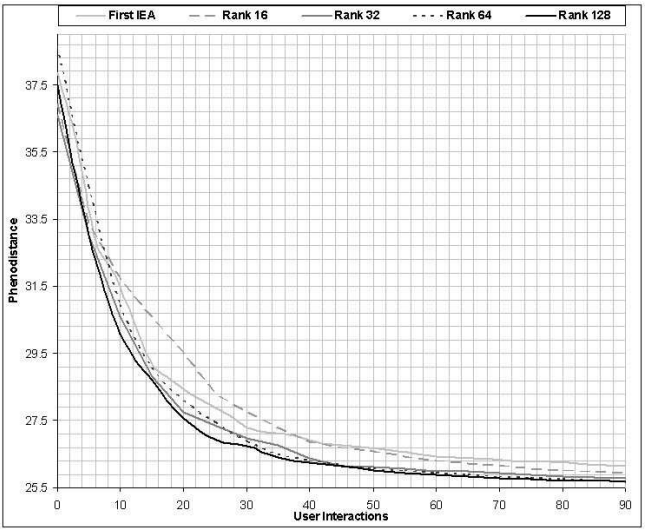


Figure 12.11. Comparison of mean convergence for different population sizes. Original Image: Sommet 256. Noisy Image: Sommet 256 with Gauss $\sigma = 20$.

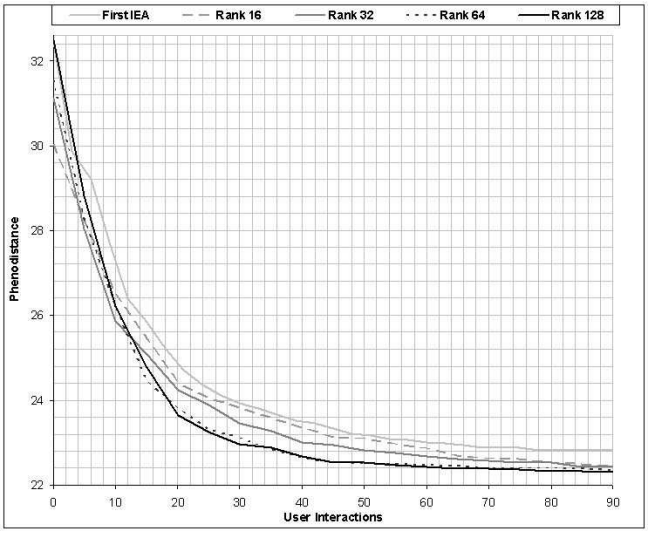


Figure 12.12. Comparison of mean convergence for different population sizes. Original Image: Lena 256. Noisy Image: Lena 256 with Gauss $\sigma = 25$.

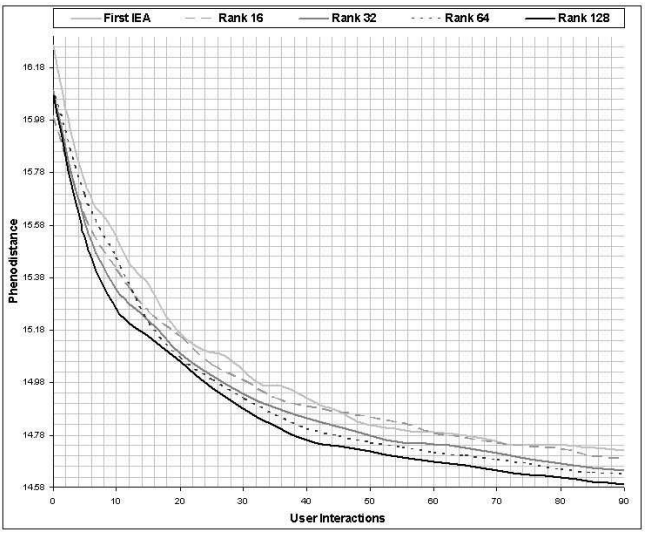


Figure 12.13. Comparison of mean convergence for different population sizes. Original Image: Mars 256. Noisy Image: Mars 256 with Gauss $\sigma = 30$.

12.11. Conclusion

This work about the use of evolutionary computation schemes for signal and image denoising leads us to several conclusions that may be considered from a wider point of view. First of all, it has been made evident – if necessary – that evolutionary schemes are efficient in signal and image analysis basic tasks, as far as we deal with complex optimisation problems. But of course, as this choice implies heavy computational costs, such a technique may not be convenient in cases where short response time is required.

Another point that has been raised, is the interest and efficiency of interactive schemes in image and signal processing: For subtle tasks where computational measurement cannot accurately reflect the judgment of the end-user, which is actually the case for image denoising, an IEA can be a solution. Once again, however, a careful design of the EA components and user interaction schemes, are necessary. For instance, the manipulation of a much larger population in conjunction with the use of rough approximations of the user fitness provides a solution to the "user bottleneck" problem.

This work also defends a viewpoint on signal and image analysis tasks, in terms of semi-automatic procedures where an end-user is involved in order to constrain the analysis towards aims for which numerical models are not available. Such an analysis may additionally have backward consequences into non-interactive procedures. For example, the fitness map scheme can be easily generalised to other applications, including non-interactive ones where exact fitness calculation is computationally expensive.

The IEA presented in this work is freely distributed in the Fraclab Toolbox, see figure 12.14, available for download at `http://complex.inria.fr`.

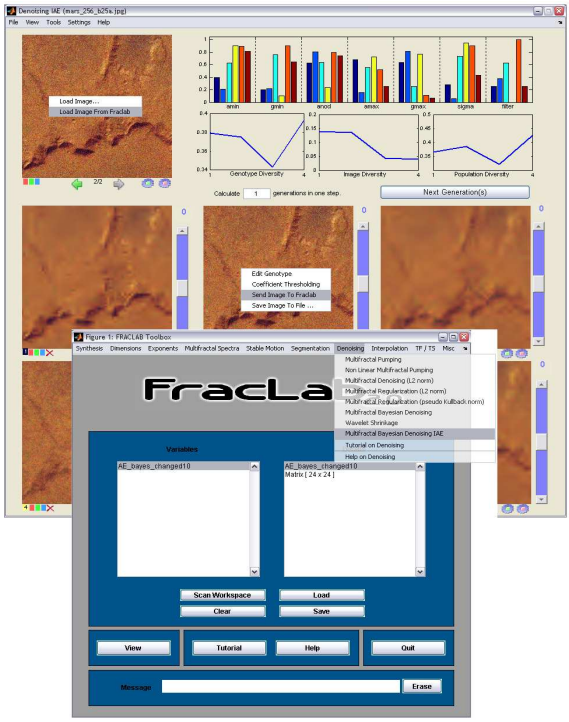


Figure 12.14. The presented IEA integrated with Fraclab.

Acknowledgment

The authors are grateful to Mario Pilz, Pierre Grenier, Yann Landrin-Schweitzer and Pierrick Legrand for their contribution, technical support, and help to integrate the software into Fraclab.

Bibliography

[1] P. 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] W. Banzhaf, "Interactive Evolution", in *Handbook of Evolutionary Computation*, 1997, Oxford University Press.

[3] G. Brown, G.Michon, J. Peyrière, *On the multifractal analysis of measures.* J. Statist. Phys., 1992, 66 (3-4), 775–790.

[4] J. Chapuis and E. Lutton, "ArtiE-Fract: Interactive Evolution of Fractals", 4th International Conference on Generative Art, Milano, Italy, December 12-14, 2001.

[5] P. Collet, E. Lutton, M. Schoenauer, J. Louchet, "Take it EASEA," Parallel Problem Solving from Nature VI, vol 1917, Springer Verlag pp 891-901, Paris, septembre 2000. *EASEA home page:* `http://sourceforge.net/projects/easea`

[6] K. Daoudi, J. Lévy Véhel, Y. Meyer, *Construction of functions with prescribed local regularity,* Constructive Approximation, 1989.

[7] R.A. Devore, B. Lucier *Fast wavelet techniques for near optimal image processing.* 1992 IEEE Military Communications Conference, 2–12 (1992).

[8] D.L. Donoho, *De-noising by soft-thresholding.* IEEE Trans. Inf. Theory 41, No. 3, 613–627 (1994).

[9] L. Gagnon, F. Drissi Smaili, *Speckle noise reduction of airborne SAR images with symetric Daubechies wavelets,* Signal and Data Processing of Small Targerts, Proc. SPIE 2759, 1996.

[10] B. Guiheneuf, J. Lévy Véhel, *2 microlocal analysis and applications in signal processing,* Int. Conf. on Wavelet, Tangier, 1997.

[11] S. Jaffard, *Pointwise smoothness, two-microlocalization and wavelet coefficients.* Publ. Mat. 35, No. 1, 155–168 (1991).

[12] Y. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," in Soft Computing, vol 9, pp 3-12, 2005.

[13] 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), vol 5, Orlando, FL, USA, pp 4227-4231, 1997.

[14] Y. Landrin-Schweitzer, P. Collet and E. Lutton, "Interactive GP for Data Retrieval in Medical Databases," in EUROGP'03, 2003.

[15] J. Lévy Véhel, "Fractal Image Encoding and Analysis: A NATO ASI Series Book", Fisher, Yuval (ed.) Springer Verlag, 1996

[16] J. Lévy Véhel, *Fractal Approaches in Signal Processing,* Fractals, **3** (4), pp 755-775, 1995.

[17] J. Lévy Véhel, B. Guiheneuf "Multifractal Image Denoising," *SCIA*, 1997.

[18] J. Lévy Véhel and P. Legrand "Bayesian Multifractal Signal Denoising", IEEE ICASSP Conference, 2003,

[19] J. Lévy Véhel, P. Legrand, Hlderian regularity-based image interpolation, *ICASSP06, IEEE International Conference on Acoustics, Speech, and Signal Processing,May 14-19, Toulouse, France,* 2006.

[20] J. Lévy Véhel and E. Lutton, "Evolutionary signal enhancement based on Hölder regularity analysis", EVOIASP2001 Workshop, Como Lake, Italy, Springer Verlag, LNCS 2038, 2001.

[21] P. Legrand, E. Lutton and G. Olague "Evolutionary denoising based on an estimation of Hölder exponents with oscillations" EVOIASP Workshop, Budapest, 2006.

[22] E. Lutton, P. Collet, Y. Landrin-Schweitzer and T. Prost, "Introducing Lateral Thinking in Search Engines with InteractiveEvolutionary Algorithms," in Annual ACM Symposium on Applied Computing (SAC 2003), Special Track on "Computer Applications in Health Care" (COMPAHEC 2003), March 9 to 12, Melbourne, Florida, U.S.A., 2003.

[23] E. Lutton, P. Grenier and J. Lévy Véhel "An Interactive EA for Multifractal Bayesian Denoising", EVOIASP Workshop, Lausanne, 2005.

[24] J. J. Merelo, *EO home page:* `http:/geneura.ugr.es/ merelo/EO.html`, Granada University.

[25] N. Monmarche, G. Nocent, G. Venturini, and P. Santini. "On Generating HTML Style Sheets with an Interactive Genetic Algorithm Based on Gene Frequencies", Artificial Evolution, European Conference, AE 99, Dunkerque, France, November 1999, Selected papers, Springer Verlag, LNCS 1829, 1999, C. Fonlupt and J. K. Hao and E. Lutton and E. Ronald and M. Schoenauer (Eds).

[26] C.J. Oliver, *Information from SAR images,* J. Phys. D, **24**, 1493-15144, 1991.

[27] 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, pp 269-277, 1997.
[28] S. Rooke, "The Evolutionary Art of Steven Rooke" http://www.azstarnet.com/∼srooke/
[29] Y. Semet, Y. Jamont, R. Biojout, E. Lutton and P. Collet, "Artificial Ant Colonies and E-Learning: An Optimisation of Pedagogical Paths," in HCII'03 - Human Computer Interaction International, 2003.
[30] K. Sims, "Interactive evolution of dynamical systems," in *First European Conference on Artificial Life, Paris, December*, pages 171–178, 1991.
[31] K. Sims, "Artificial Evolution for Computer Graphics," *Computer Graphics*, 25(4):319–328, July 1991.
[32] H. Takagi, " Interactive Evolutionary Computation : System Optimisation Based on Human Subjective Evaluation", IEEE Int. Conf. on Intelligent Engineering Systems (INES'98), Vienna, Austria, pp 1-6, Sept 17-19, 1998.
[33] H. Takagi, M. Ohsaki, "IEC-based Hearing Aids Fitting", IEEE Int. Conf. on System, Man and Cybernetics (SMC'99), Tokyo, Japan, vol 3, pp 657-662, Oct. 12-15, 1999.
[34] S.J.P. Todd and W. Latham, "Evolutionary Art and Computers", Academic Press, 1992.
[35] M. Wall, *GAlib home page:* `http://lancet.mit.edu/ga/` , MIT.

Evelyne Lutton, Jacques Lévy Véhel, COMPLEX team, INRIA, Domaine de Voluceau, 78153 Le Chesnay Cedex, France