Applied Intelligence 16, 101–118, 2002


# All the Truth About NEvAr*

PENOUSAL MACHADO AND AMÍLCAR CARDOSO
*Centro de Informática e Sistemas da Universidade de Coimbra (CISUC), Pinhal de Marrocos, 3030 Coimbra, Portugal*
machado@dei.uc.pt
amilcar@dei.uc.pt

**Abstract.** The use of Evolutionary Computation approaches to generate images has reached a great popularity. This led to the emergence of a new art form—Evolutionary Art—and to the proliferation of Evolutionary Art Tools. In this paper, we present an Evolutionary Art Tool, NEvAr, the experimental results achieved, and the work methodology used to generate images. In NEvAr, useful individuals are stored in a database in order to allow their reuse. This database is playing an increasingly important role in the creation of new images, which led us to the development of automatic seeding procedures, also described. The automation of fitness assignment is one of our present research interests. We will, therefore, describe some preliminary results achieved with our current approach to automatic evaluation.

**Keywords:** evolutionary art, genetic programming, computational creativity

## 1. Introduction

Creativity is often regarded as one of the most impressive features of the human mind, which may explain the current interest of the Artificial Intelligence community in the study of computational creativity. Several other factors, however, contribute to this rising interest: artificial creative systems may prove useful in a wide range of artistic, architectural and engineering domains where conventional problem solving techniques have failed; its study may bring insight to the overall understanding of human creativity; the study of artificial creativity can be viewed as the logical next step in AI research, i.e. if we can already build systems capable of performing tasks requiring intelligence, can we build systems able to perform tasks requiring creativity?

The development of artificial creative systems is often inspired in models of the human creative process (e.g. [1–4]). There are, however, other possible sources of inspiration. Evolution is responsible for the development of an incredible amount and variety of solutions, *species*, to a specific problem, *survival*. It is, therefore, unquestionable that this process can give rise to innovative solutions [5].

In the past few years two Evolutionary Computation (EC) techniques, namely Genetic Algorithms [6] (GA) and Genetic Programming [7] (GP), have been used as means to implement computational creativity, resulting in the development of several applications in fields such as music and image generation, architecture and design.

GA are the most common EC approach in the musical field (e.g. [8–11]). However, according to [12], and in spite of the numerous applications, GA are not ideal for the simulation of human musical thought.

In the image generation field, GP is the most used EC approach, some examples being [13–16], which evolve images, and [17], where GP is used to evolve human faces. GP has also been successfully applied in the fields of design [18, 19] and animation [14, 16, 20].

The main difficulty in the application of EC approaches to fields such as image and music generation is the development of an appropriate fitness function. As a result, most systems rely on Interactive Evolution (IE),

*This work was partially funded by the Portuguese Ministry of Science and Technology, under Program PRAXIS XXI.

i.e. the user evaluates the individuals and thus guides evolution. There are several systems in the musical field that perform automatic evaluation (e.g. [21–25]). In the field of image generation, however, there was only one attempt to fully automate fitness assignment [26].

The use of IE for image generation has achieved a great popularity. The roots of these applications can be found in Richard Dawkins book "The Blind Watchmaker" [13], in which the author suggests the use of a GA to evolve the morphology of virtual organisms, *biomorphs*. This work was, apparently, the source of inspiration for the systems developed by K. Sims [14] and W. Latham [27], which can be considered as the first applications of IE in the field of the visual arts, and which are usually considered as the most influential works in this area. The success of these approaches has led to the emergence of a new art form, "Evolutionary Art", and also to the proliferation of IE applications in this field, usually called Evolutionary Art Tools.

In this paper we will make an in-depth description of an Evolutionary Art Tool, NEvAr (Neuro Evolutionary Art). Our objective is twofold: provide useful information on the development of an Evolutionary Art Tool; present our current research ideas, which we consider that can enrich nowadays systems.

The paper is structured as follows: in Section 2 we make an overview of the system, which comprises the description of NEvAr's evolutionary model, used representation (2.1), and genetic operators (2.2); we proceed by presenting experimental results (Section 3) and the process used to produce them (3.2); in Section 4 we describe our current research efforts, which are related with the development of seeding procedures (4.1) and automatic evaluation (4.2), and present some preliminary experimental results; finally, we draw some conclusions and point directions for future research.

## 2. Overview of the System

Fitness assignment plays a key role on EC algorithms since it guides the evolutionary process. Consequently, the quality of the results is deeply connected with the quality of the evaluation. In its present form, NEvAr is mainly an IE system, therefore, the user plays a key role in the process.

The interaction between human and computer has some advantages, but also poses some problems. It is, for instance, virtually impossible to use large population sizes or to perform extended runs. It was clear from the beginning of its development that if NEvAr were to succeed, i.e. produce appealing images, it would have to do it in few evolutionary steps and with a low number of individuals' evaluations. On the other hand, a skilled user can guide the evolutionary process in an extremely efficient way. She/he can predict which images are compatible, detect when the evolutionary process is stuck in a local optimum, etc. In other words, the user can change its evaluation criteria according to the specific context in which the evaluation is taking place.

It is crucial to consider these idiosyncrasies in the development of an Evolutionary Art Tool. In Fig. 1 we show the evolutionary model of NEvAr. From now on, we will designate by *experiment* the set of all populations of a particular GP run.

NEvAr implements a parallel evolutionary algorithm, in the sense that we can have several different and independent *experiments* running at the same time. It is also asynchronous, meaning that one *experiment* can be in its first population, while another can be in its hundredth. Additionally, we can transfer individuals between *experiments* (migration).

The use of migration allows the construction of image databases. We can, for instance, create an empty *experiment* and transfer to it individuals that we find interesting or useful. The images belonging to this database can be used to initialise new *experiments*, or added to the current population of an *experiment* when the user finds that this addition may improve the evolutionary process.

NEvAr also allows the migration within the same *experiment*. This feature is important due to the limited size of the populations, allowing the revival of images from previous ones. It is also possible to go to a previous population and change the evaluation of the individuals, which is extremely useful since it allows the exploration of different evolutionary paths.

Through time, we have constructed a large database of individuals. Nowadays, the extensive use of this database plays a key role in NEvAr, as will be shown in Section 3. Through the use of the databases and migration, we try to overcome one of the main weaknesses of EC approaches: the lack of long term memory mechanisms (although *multiploidy* can be viewed as a limited memory mechanism). Although the use of migration and databases is not particular to NEvAr (e.g. systems like [14, 15] also possess these features), the emphasis we give to its use is. Additionally, we have also developed automatic seeding mechanisms, which will be explained in Section 4.1.

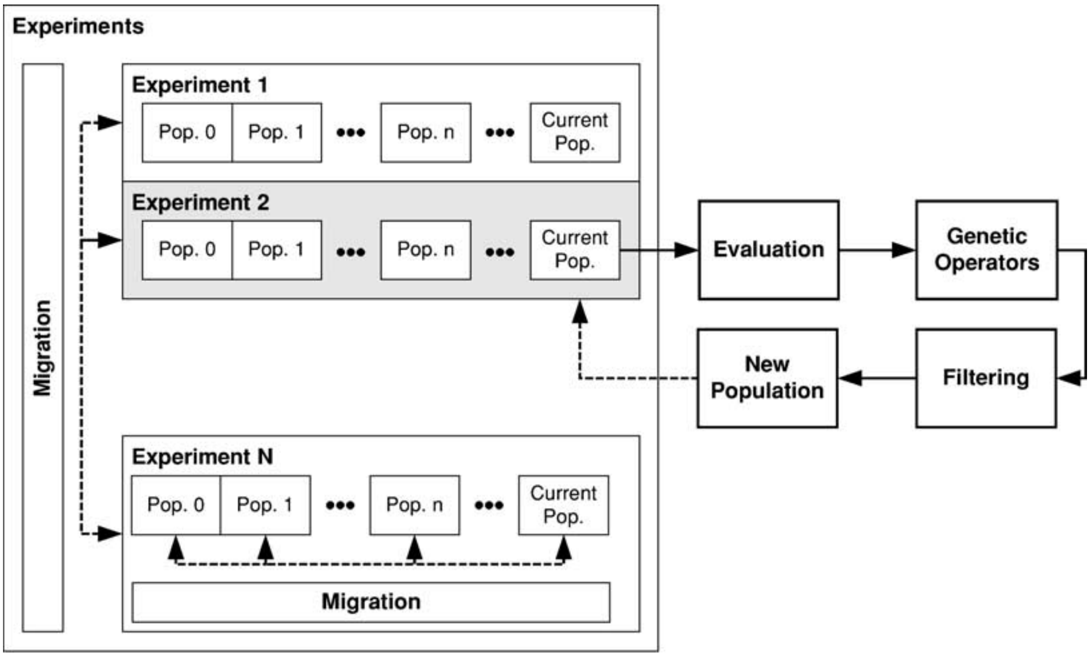


*Figure 1.* Evolutionary model of NEvAr. The active experiment is depicted in grey.

A final word goes to our efforts to automate fitness assignment. Automatic Evaluation is still under development; in Section 3 we will present our current approach and the experimental results achieved so far. The filtering module is linked with the idea of automatic evaluation and will be explained in Section 4.2.

### 2.1. Representation

In NEvAr, like in most GP applications, the individuals are represented by trees. Thus, the genotype of an individual is a symbolic expression, which can be represented by a tree. The trees are constructed from a lexicon of functions and terminals. The internal nodes are functions and the leafs terminals. We use a function set composed, mainly, by simple functions such as arithmetic, trigonometric and logic operations. The terminal set is composed by the variables $x$ and $y$, and by constants which can be scalar values or 3d-vectors.[1]

The interpretation of a genotype (an individual) results on a phenotype, which in NEvAr's case is an image. To generate an image, we evaluate the corresponding expression for each pixel coordinate and the output is interpreted as the greyscale value of the pixel. In Fig. 2 we present some examples of genotypes and their corresponding phenotypes.

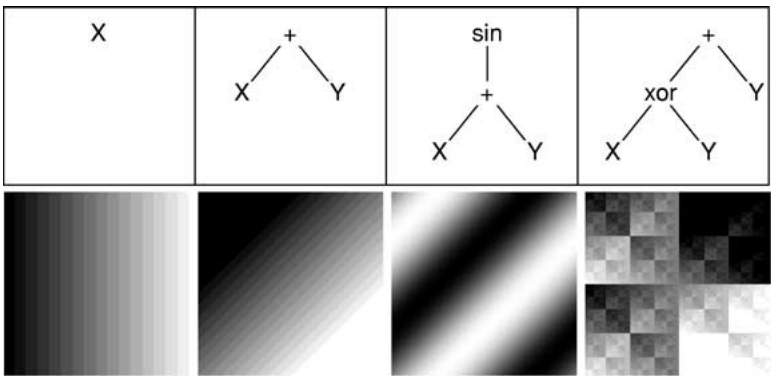


*Figure 2.* Some simple functions and the corresponding images.

NEvAr allows the evolution of true-colour images, which is achieved by the use of the 3d-vector terminals. Each of the components of these vectors correspond to a different colour channel (Red, Green and Blue or, alternatively, Hue, Saturation and Value). We will resort to an example in order to explain how the calculations are performed. Consider the following expression: $\sin(([0, 1, 0.5] \times ([1, 0.5, 0.5]) + X)$. The first operation to execute is the multiplication of the two vectors; the multiplication is not performed in the standard way. Instead, each of the components of the first vector is multiplied by the corresponding component of the second, i.e. $[0 \times 1, 1 \times 0.5, 0.5 \times 0.5] = [0, 0.5, 0.25]$. Next we add the variable $X$ to this vector, which yields $[0 + X, 0.5 + X, 0.25 + X]$. Finally, we apply the sin operator to each of the components, thus obtaining: $[\sin(0 + X), \sin(0.5 + X), \sin(0.25 + X)]$. By using this approach, we avoid having to develop special operators designed to manipulate vectors; instead each operator is applied to a scalar value that represents a colour component of the image.[2]

### 2.2. Genetic Operators

We use two kinds of genetic operators: recombination and mutation. For the recombination, we use the standard GP crossover operator [7], which exchanges sub-trees between individuals. In GP mutation is, usually, considered less important than recombination [7]. In NEvAr, however, the picture is quite different. Conventional GP systems use a small function set and large population sizes. In NEvAr this situation is inverted. Therefore, mutation becomes necessary, in order to allow the reintroduction of genetic material that would be otherwise lost.

We resort to five mutation operators (see Fig. 3 ):

- Sub-tree swap—randomly select two mutation points and exchange the corresponding sub-trees.
- Sub-tree replacement—randomly select a mutation point and replace the corresponding sub-tree by a randomly created one.
- Node insertion—randomly select an insertion point for a new, randomly chosen, node. If necessary, create the required arguments randomly.
- Node deletion—the dual of node insertion.
- Node mutation—randomly select a node and change its value.

The genetic operators induce changes at the phenotype level. In Fig. 4, we show examples of the application of the crossover operator. As can be seen, the crossover between two images can produce interesting

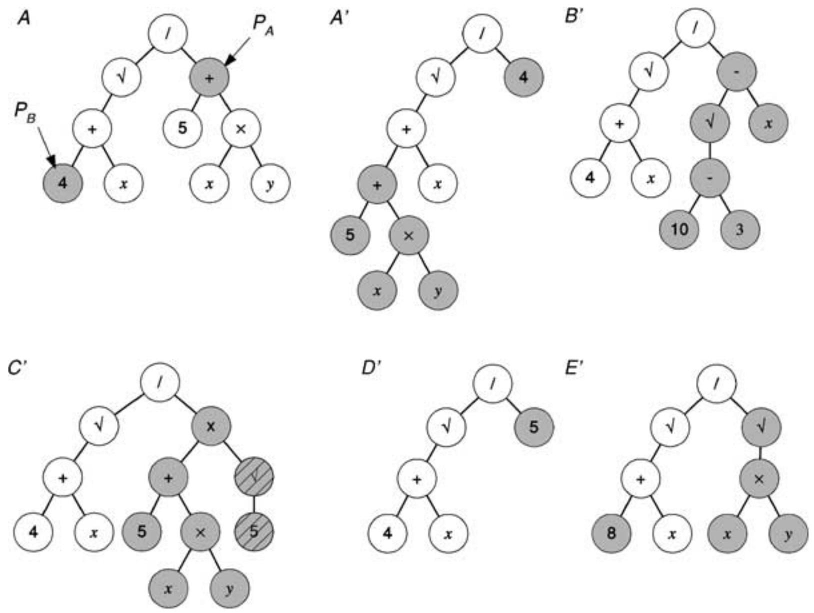


*Figure 3*. Examples of the application of the mutation operators on individual $A$. The individual $A'$ was generated by sub-tree swap, $B'$ by sub-tree replacement, $C'$ by node insertion, $D'$ by node deletion and $E'$ by node mutation.

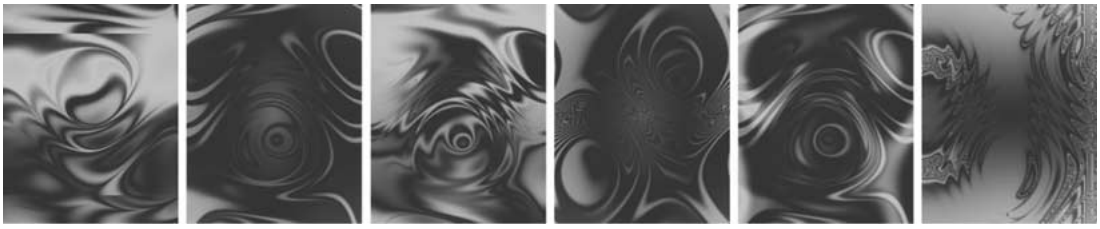

*Figure 4*. On the left, the two progenitor images. On the right, some images resulting from their crossover.

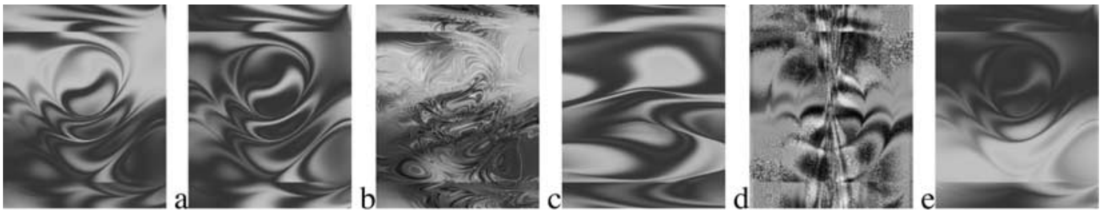


*Figure 5*. On the left, the original image. On the right, several mutations of this image. Image *a* was generated through node mutation, *b* through node insertion, *c* through node deletion, *d* through sub-tree swap and *e* by sub-tree replacement.

and unexpected results. Additionally, there are cases in which the images seem to be incompatible, i.e. images that, when combined, result in "bad" images.

In Fig. 5 we give examples of images generated through mutation. Once again, the results of this operation can give quite unexpected results.

## 3. Experimental Results

As an Evolutionary Art Tool, the main goal of NEvAr is the production of artworks. Its analysis must be performed with this in mind. A strictly objective analysis of the results achieved with NEvAr is impossible, due to the nature of the results (images). The most obvious way to assess the performance of the system is to ask to a set of people to rate the images generated by the system. This type of analysis has a certain appeal and appears to be a valid way to evaluate the system's results. However, this idea encompasses a fundamental error.

NEvAr is guided by a user. Therefore, its goal is to produce images compatible with the aesthetic and/or artistic principles of the user. As such, the evaluation by third parties is secondary. Thus, the really important issue is the level of user satisfaction.

It is important to notice that the fact that a tool can generate "interesting" images is irrelevant from the artistic point of view. What is really important is that the produced artworks convey the artistic ideas of the artist using the tool. In other words, the artist must be able to express her/himself through the use of the tool.

The images generated with NEvAr during the early stages of experimentation were clearly disappointing. This failure didn't result from the lack of power of the tool, but from our lack of expertise in its use. Like any other tool, NEvAr requires a learning period. To explore all the potential of a tool, the user must know it in detail and develop or learn an appropriate work methodology. The results, and user satisfaction, depend not only on the tool but also on its mastering. In other words, the evaluation of a piano as a tool can only be made by someone that knows how to play it well. This rises a problem: unfortunately few people know how to play NEvAr.

Due to these factors, instead of presenting charts evaluating the performance of NEvAr, we will present some results, i.e. images. Our intention is to show the type of artwork that can be produced by the system. Additionally, we will describe the work methodology that we currently use to generate images with NEvAr.

### *3.1. The Results*

Next we will present several examples of images generated with NEvAr. The images presented in Fig. 6 are a subset of the ones presented at the "*Art and Aesthetics of Artificial Life*" exhibit, that took place at the Centre for the Digital Arts of the UCLA, during 1998.

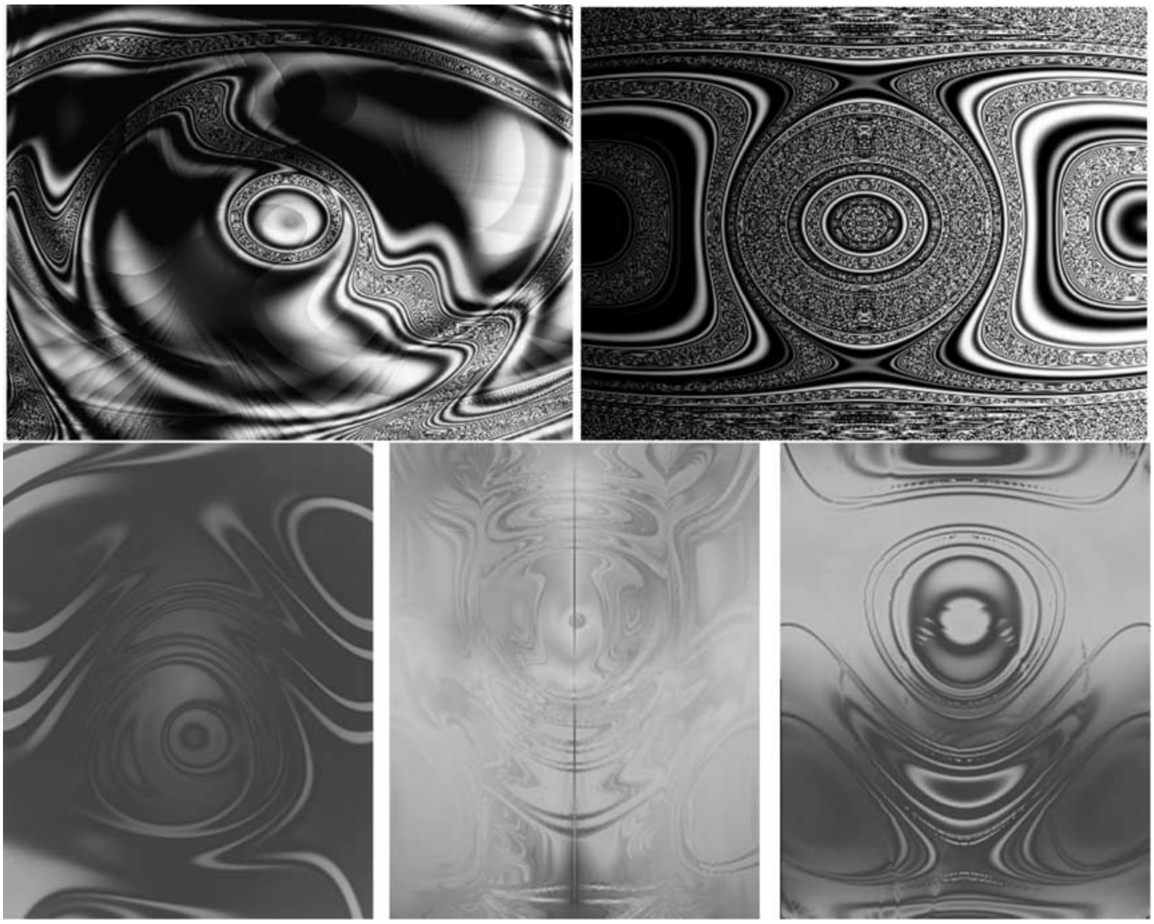

*Figure 6*. Some of the images presented at the "*Art and Aesthetics of Artificial Life*" exhibit, curator: Nicholas Gesseler, 1998. Further images can be found in the CD-ROM accompanying [19].

From the presented examples, it is clear that the images generated with NEvAr are typically of abstract nature. From the theoretical[3] point of view, it is possible to construct pictorial images, but in practice that is hard to achieve.

### *3.2. The Process*

The creation of an artwork encompasses several stages, such as: genesis of the idea, elaboration of sketches, exploration of the idea, refinement, and artwork execution. The methodology that we propose can be considered, in some way, analogous. It is composed by four main stages: Discovery, Exploration, Selection and Refinement.

These stages can be described, concisely, as follows: the stage of Discovery consists on finding a promising evolutionary path, which, typically, corresponds to evolving a promising set of images from an initial random population (generation of the ideas); in the second stage, Exploration, the "ideas" evolved on the previous stage are used to generate images of high aesthetic value (exploration of the ideas); the Selection stage involves choosing the best produced images; the selected images, when necessary, will be subjected to a process of Refinement, whose goal is the alteration of small details (i.e. it corresponds to the final execution of the artwork).

Next, we will describe how we use NEvAr in each of the mentioned stages, referring the most important issues to take into account in each of them and the skills involved. In this description we will consider that we start with a randomly generated population. The use of the image database will be described afterwards.

***3.2.1. Discovery.*** Our empirical experience allows us to classify the Discovery stage as the most crucial of the process, and, together with the Exploration stage, the one in which the faculties of the user are more important.

Discovery corresponds to the genesis of the idea, and is therefore inappropriate to approach this stage with pre-conceived ideas regarding the final aspect of the artwork. In other words, it is impossible in practice (yet tempting) to think on an image and use NEvAr to evolve it. This is probably the most important aspect to retain, because it contrasts with what is usually expected in a tool, i.e. that it allows the implementation of an idea. This aspect can be viewed as a weakness, but it is also the distinguishing feature and strength of NEvAr (and other evolutionary art tools). A conventional tool only plays an important role in the artistic process in stages posterior to the generation of the idea. NEvAr, however, plays a key role in the generation of the idea itself. Its influence is noticeable throughout all the artistic process and in its main creative stage. The artist is no longer responsible for the creation of the idea; instead, she/he is responsible for the recognition of promising concepts. More precisely, the idea results from an evolutionary process, and is created by the artist and the tool, in a (hopefully) symbiotic interaction.

The most important principle that we have identified is the prevalence of the form in relation to the colour. Thus, during the initial stages of the evolutionary process the user should concentrate in the evolved forms and "forget", for the time being, hers/his chromatic preferences. One way to achieve independence from colour is to use, exclusively, greyscale images during the first populations, until appropriate forms are found, and only allow the generation of full colour images afterwards. This methodology proved to be extremely efficient, allowing a greater systematisation of the evolutionary process. The way NEvAr handles colour is adequate to this form of operation, since it typically allows the alteration and incorporation of colour without significantly changing the images' form.

The images belonging to the first populations usually fall in one of two extremes, being either too simple or too complex. The discovery stage is characterised by the combination of simple images, in order to gradually increase image complexity until reaching an appropriate level. It is advisable to avoid "noisy" images, since the removal of noise is usually difficult. In Fig. 7 we present the first populations of an experiment.

In which concerns fitness assignment, the images have by default a fitness value of 0, and we usually give a classification greater than 0 to a very restrict set of images (typically 1 to 3). The next population will be, therefore, composed by the combination of these images. We use roulette wheel instead of tournament selection. Although tournament selection is usually preferred in GP systems, in our opinion roulette selection is more adequate to IE systems, since it is more intuitive and allows a greater degree of control by the user.

During the exploration stage, we use high crossover and mutation rates (e.g. 90% crossover and 20% mutation probabilities). The objective is to increase population diversity, thus avoiding the loss of interest by the user. In Fig. 8, we present the best images from population 7 to 30. Colour was introduced in the 10th population. By the 21st population, the images were sufficiently interesting to allow a transition to the Exploration stage.

***3.2.2. Exploration.*** The goal of this stage is to explore the ideas present in the current population in order to produce something close to an artwork. When we reach this stage, we are already dealing with images of high aesthetic value. Through the recombination of these images, we explore a space of forms smaller than the one explored in the discovery stage, and which may be, therefore, more thoroughly searched.

Ideally, the quality of the populations should increase steadily. In practice this is unachievable. The Exploration stage can prolong itself conducting the artist to a point that, at least apparently, has nothing to do with the original one. Sometimes, the path to this point is relatively direct. There are cases, however, in which a dead end is reached. In these cases, it is necessary to descend the hill, which usually implies a deliberate action by the user. This descent may cause a return to the Discovery stage.

Apparently, the discovery of new interesting images, even when they have a completely different aspect from the previously generated ones, is usually faster than the first discovery process. This may indicate that some type of learning occurs during the evolutionary process, i.e. that, during evolution, useful combinations of primitives are built, and that these sub-trees can be recombined allowing the generation of interesting images in few populations.

Additionally, the high plasticity inherent to the used representation method allows the evolution of radically different phenotypes from resembling genotypes.

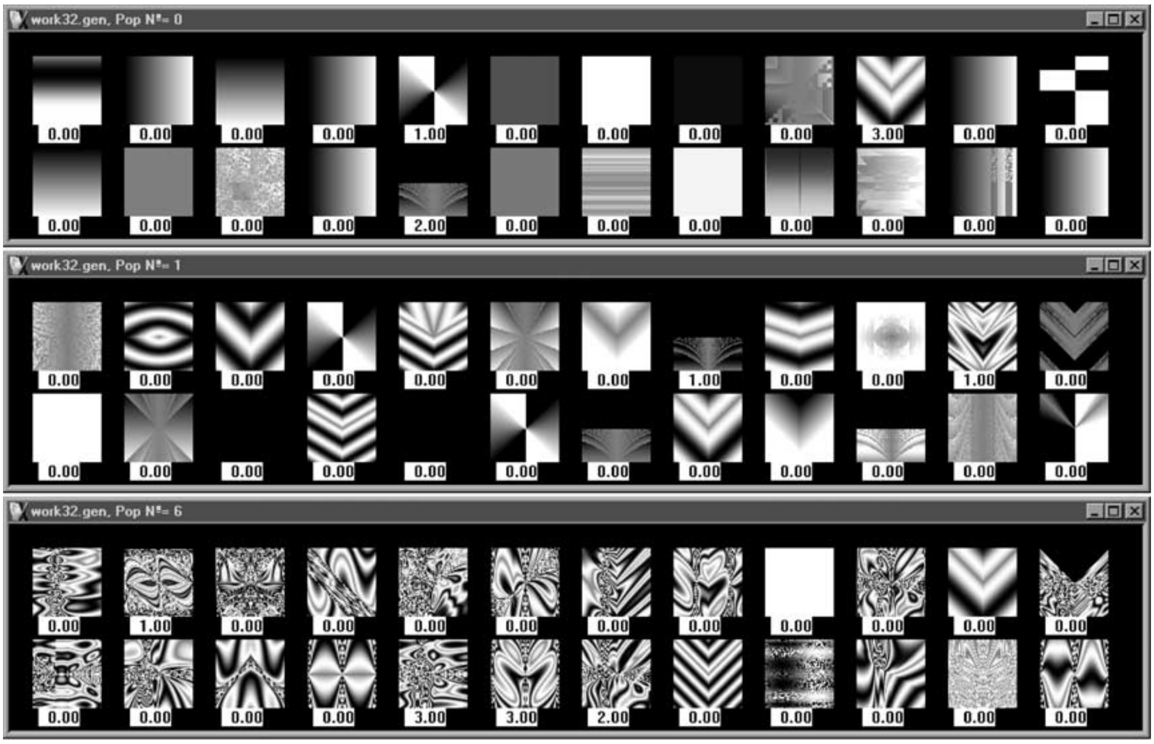


*Figure 7*. Populations 0, 1 and 6 of an experiment. The numbers bellow the images correspond to the fitness given by the user. The increase in the complexity of the images is evident. From the 6th population on the user gave preference to organic and fluid forms in detriment of geometric ones.

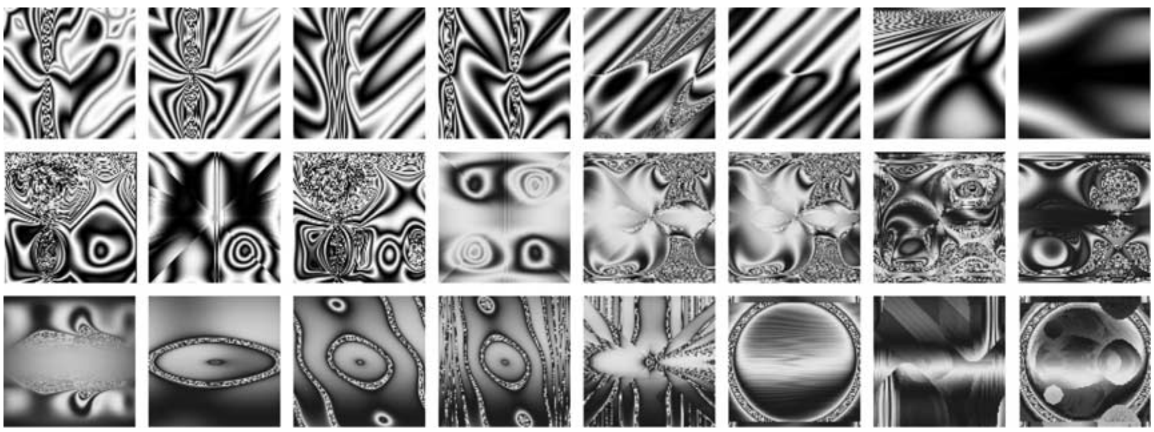

*Figure 8*. The best images, according to the fitness assignment, of populations 7 to 30 (from left to right and top to bottom).

It is important to notice that an individual is more than its phenotype. During evolution there is an accumulation of genetic material that is not expressed in the phenotype. This genetic material can become active at any time due to crossover or mutation. A striking evidence of this fact is the reappearance of images that where already abandoned by the evolutionary process (e.g. the reappearance in population 30 of an image from population 5).

Like in the Discovery stage, the expertise of the user is determinant to the success of the Exploration stage. With the accumulation of experience, the user learns how to distinguish between promising paths and the ones that lead nowhere, to predict which combinations of images produce best results, how to manipulate crossover and mutation rates in order to produce best results, etc.

In Figs. 9 to 11 we present some populations belonging to this stage.

***3.2.3. Selection.*** The Selection stage can be divided in two different ones: one that is concurrent with the evolutionary process, and one that is posterior. During the stage of Exploration the best images (according to the user criteria) are added to a different *experiment*, which works as a gallery. As stated before, NEvAr stores all populations, which allows the review of the

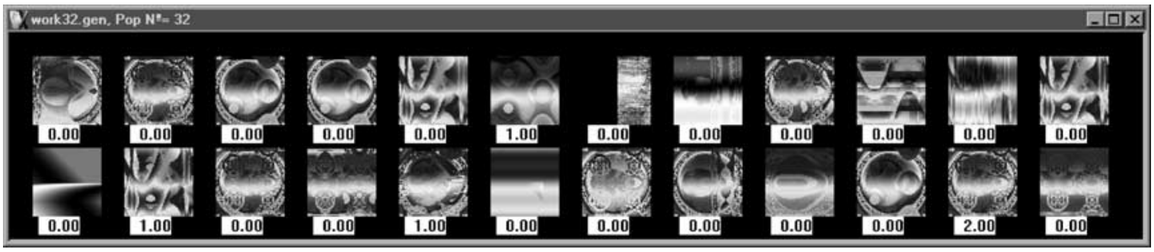


*Figure 9*. The user is exploring images generated during Discovery. The connection between the images of population 32 and the best images of populations 29 and 30 is evident (see Fig. 8)

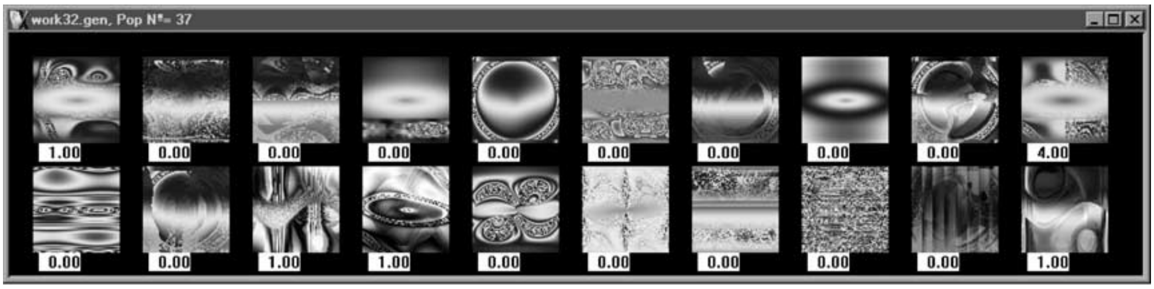


*Figure 10*. At this stage the user chooses to abandon the idea that she/he was exploring. The circular images that dominated previous populations where neglected, with the objective of introducing a change of path.

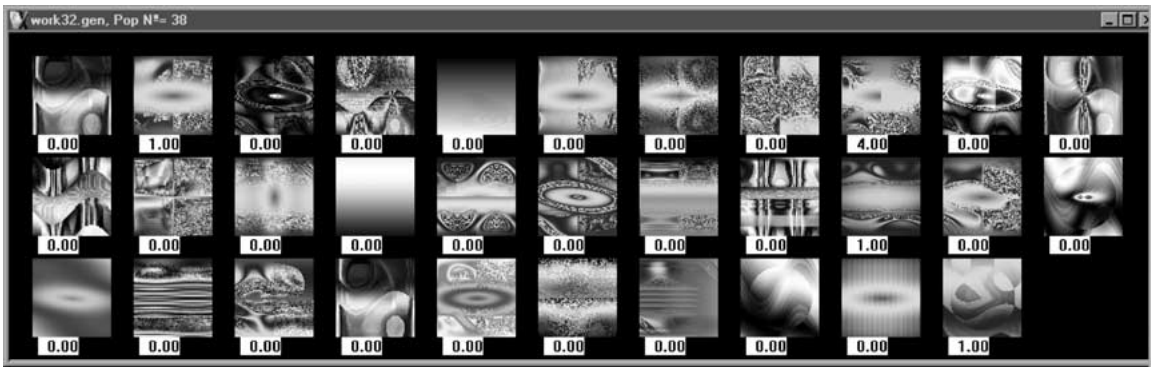


*Figure 11*. As can be seen there was a significant increase in population diversity. Population size was increase in order to allow a greater search width. The user continues to force a change of evolutionary path through hers/his choices, which will result in a decrease of the average quality of the images, and a transition to the Discovery stage. There where several other transitions between this stages, and the exploration stage was ended in the 57th population.

evolutionary process and the addition to the gallery of images that were previously neglected. This revision is highly recommended, and a substantial amount of time should separate the generation of the images and its review in order to allow the necessary distance between generation and criticism. In Fig. 12, we show the image gallery resulting from the experiment that we have been using to illustrate our description.

The images belonging to the gallery will be subjected to a process of analysis, which will lead to their division in four groups:

- Discard—Images that are considered irrelevant.
- Useful—Images that, at least apparently, represent good ideas and that should be stored in the databases of NEvAr.
- Refine—Images that still need some work to achieve the status of artworks.
- Artworks—Composed, ideally, by images that fully satisfy the aesthetic and/or artistic criteria of the user.

In Fig. 13 we present the results of the classification of the images of Fig. 12 in the above mentioned groups.

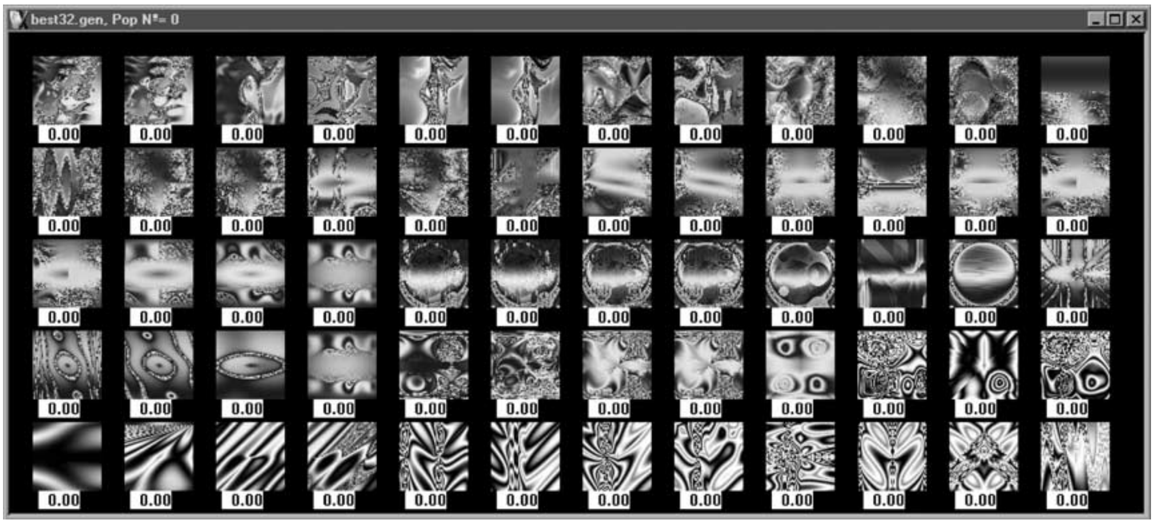


*Figure 12*. Image gallery.

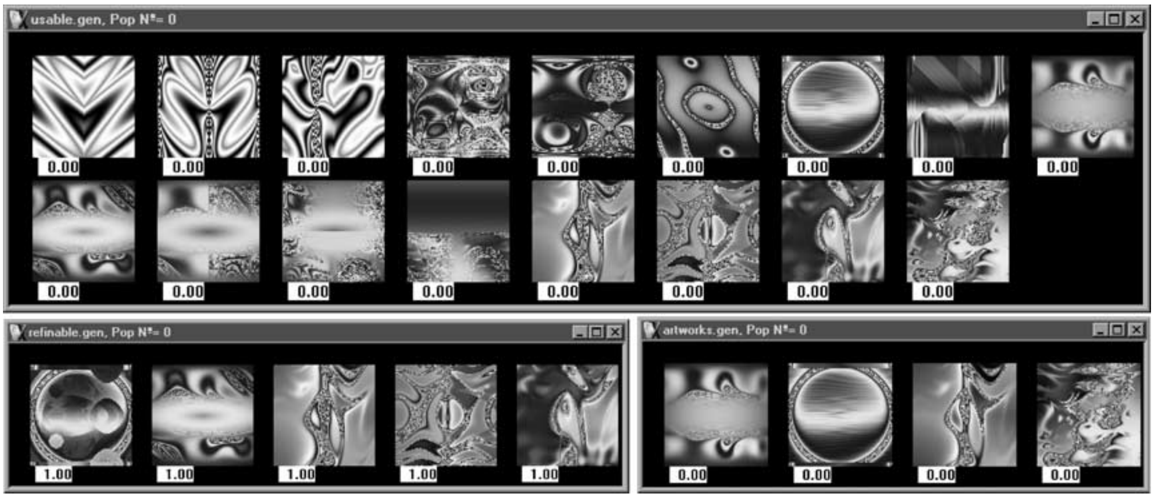


*Figure 13*. The Useful, Refine and Artwork groups. An image can belong to several groups.

***3.2.4. Refinement.*** The Refinement process usually occurs separately from the *experiment* that generated the image. The common procedure is to initialise a new *experiment* with the image we want to refine (i.e. the initial population of this *experiment* will be composed by the image and, in some cases, similar ones). The generation of new populations, from this initial one, allows the exploration of a search space in the vicinity of the image that we want to refine.

It is important to notice that there is a difference between the refinement of an idea and the retouching of an image. Inducing specific changes in an image (e.g. in order to correct an imperfection) may prove difficult, and NEvAr doesn't seem to be the right tool for that kind of job.

***3.2.5. Image Database.*** The database has been used mainly in two situations to initialise new *experiments* and to add individuals to the current population of an *experiment*.

The goal of the first form of use is to shorten, or even avoid, the initial stages of the evolutionary process (Discovery and Exploration). In Fig. 14 we present an example of this type of operation, and in Fig. 15 the best individuals of the first 20 populations of this *experiment*. We are currently working on the development of automatic seeding methods. Our current approach is CBR inspired and will be described in Section 4.1.

The addition of previously generated individuals to the current population usually follows an opportunistic reasoning. There are several situations in which this may be useful, for instance to avoid a local optimum, or when we find an image whose combination with a previously created one is previewed as promising.

The image database is playing an increasingly important role in the process of image generation, and is currently a priceless feature of the system.

One of the misconceptions about Evolutionary Art Tools is that the quality of the generated images is deeply connected with the used primitives, hence the emphasis on the development of "high level" functions (e.g. fractals) that are able to generate interesting images on their own. The experimental results achieved with NEvAr, i.e. the generated images, show that a set of "basic" primitives, which can be combined in powerful ways, is enough to produce high quality results.

Introducing "high level" primitives can be seen as a way to incorporate knowledge into the system. By resorting to these primitives, interesting images can be

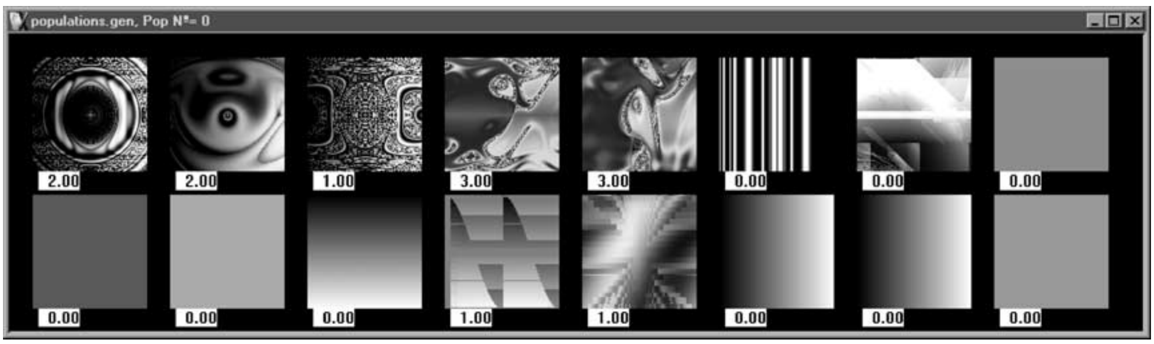


*Figure 14*. The initial population, composed by five individuals belonging to the database, and eleven randomly created ones.

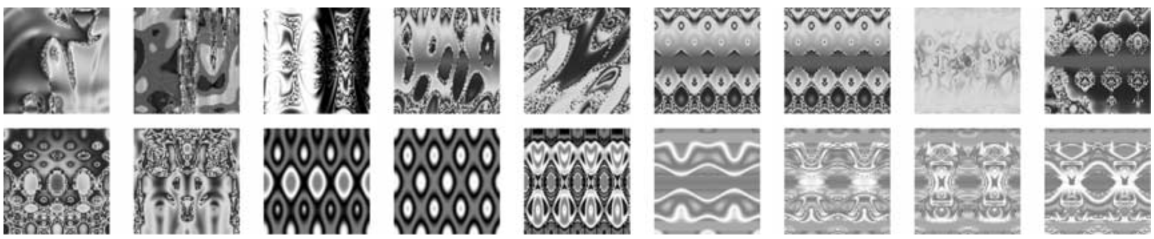

*Figure 15*. The best individuals of populations 2 to 20 (from left to right and top to bottom).

generated from compact genotypes. Consequently, the number of populations needed to produce interesting images is decreased. However, the price to pay may be too high, since it may lead to the generation of stereotyped images.

Any EC system has its points of attraction. In other words, some images are easier to generate than others; some of them can even be considered recurrent. The points of attraction depend, obviously, on the set of used primitives.

High level primitives tend to introduce points of attraction of average, or even high, quality (i.e. images of high aesthetic value). In a seemingly paradoxical way, this can be undesirable. The system has a natural tendency to converge to these points. Additionally, the user also tends to prefer these images due to their high aesthetic value. Therefore, the convergence to these points of attraction is almost unavoidable; hence the production of stereotyped images.

When the points of attraction of the system are images of low aesthetic value, which is usually the situation when low level primitives are used, the user evaluation of the images is enough to overcome the tendency of the system to these points, provided that this tendency is not too high and that the primitives can be combined in ways that allow the generation of interesting images.

It's important to notice that sets of high level primitives that don't have these shortcomings may exist. However, according to our empirical experience, their construction is extremely difficult.

## 4. Ongoing Research

Our current research efforts can be divided into two different areas: development of seeding procedures and automatic evaluation. In this section, we will make a description of our ongoing research concerning seeding and automation of evaluation, and present the experimental results achieved so far.

### *4.1. Seeding*

As we have already stated, seeding is an important part of NEvAr. Until recently, the user was responsible for the seeding process, i.e. she/he can choose individuals from the image database and transfer them to the initial population. In the beginning, this procedure was adequate. However, the growth of the image database made it an arduous task. Hence, the idea of automating the seeding procedure.

Our current approach is CBR inspired. The idea can be described as follows: The user chooses an image, and the seeding procedure selects similar individuals, belonging to the image database, to initialise the GP *experiment*.

In order the implement such an approach we must be able to compare images, in other words, we need to develop a similarity metric. Unfortunately, comparing images is not an easy task. Our first idea was to use the root mean square error (*rmse*) among two images as similarity metric (when the error is zero the similarity is maximal). The similarity between two images, *a* and *b*, was given by the following formula:

$$rmse\ sim_{a,b} = \frac{100}{1 + \sqrt{rmse_{a,b}}} \tag{1}$$

This measure is usually used to evaluate the error involved in the compression of images. However, our experiments show that the *rmse* is not appropriate for our task. In fact, it is easy to create two images, which are indistinguishable to the eye, and which have maximum dissimilarity according to this measure (e.g. consider two images composed by alternate vertical black and white stripes of one pixel width, the first image starts with a black stripe the second with a white one, these images will be similar to the eye and the *rmse* will be maximum).

The failure of this approach made us realise that the goal was not to find "mathematically" similar images, but images that resemble each other and that possess similar characteristics.

It is a well-known fact that some compression methods work better with some types of images than others. The *jpeg* format, for instance, is more appropriate for the compression of natural images than for computer generated images; fractal image compression takes advantage of the self-similarities present on the images and will, therefore, perform better when these similarities are big. Additionally, the quality of the compression is usually connected with the complexity of the image (i.e. with the predictability of its pixels) and can therefore be used as an estimate of image complexity.

Our previous experience with image compression methods led us to believe that we could use the quality of the compression to develop a similarity metric. For the scope of this paper we will define compression quality as:

$$\frac{\text{Compression ratio}}{rmse}, \tag{2}$$

and compression complexity as the inverse.

We use two different compression methods: *jpeg* and fractal based. The fractal image compression algorithm makes a quad-tree partitioning of the image [20]. By changing the maximum depth of the tree, we can specify, indirectly, the limits for the error involved in the compression. During compression, the colour information is discarded, the images are converted to greyscale and then compressed.

Let's define: "Image Complexity", $IC$, as the compression complexity resulting from the use of the *jpeg* method; "Processing Complexity", $PC$, as the compression complexity resulting from the application of the fractal based approach. We use two different maximum tree depths, $N$ and $N-1$. Therefore we have two different "Processing Complexity" estimates, $PC_1$ and $PC_2$.

In order to compare two images, $a$ and $b$, we start by calculating $IC$, $PC_1$ and $PC_2$, for each of them. The similarity between images $a$ and $b$ is given by the following formula:

$$sim_{a,b} = \frac{1}{1+\sqrt{|IC_a - IC_b| + |PC_{1a} - PC_{1b}| + |PC_{2a} - PC_{2b}|}} \tag{3}$$

In Fig. 16 we present a subset of the images belonging to the database. In Table 1, we present the $IC$, $PC_1$ and $PC_2$, measures for each of them as well as the similarity of these individuals with images 9 and 14 of the population.

*Table 1.* The $IC$, $PC_1$ and $PC_2$, measures for each of the images presented in Fig. 16, and the similarity among these images and images 14 and 9 of the same figure.

| Image | CI | CP1 | CP2 | Similarity to 14 | Similarity to 9 |
|---|---|---|---|---|---|
| 0 | 5.053 | 19.228 | 5.957 | 10.397 | 17.500 |
| 1 | 4.455 | 10.503 | 4.646 | 9.790 | 22.703 |
| 2 | 2.926 | 5.518 | 2.403 | 9.365 | 37.261 |
| 3 | 4.085 | 11.256 | 5.957 | 9.879 | 21.529 |
| 4 | 6.401 | 21.357 | 7.057 | 10.697 | 16.189 |
| 5 | 5.965 | 21.663 | 6.504 | 10.650 | 16.365 |
| 6 | 4.694 | 13.395 | 4.988 | 9.976 | 20.486 |
| 7 | 5.744 | 19.373 | 6.795 | 10.503 | 16.981 |
| 8 | 12.125 | 91.074 | 25.331 | 16.948 | 8.349 |
| 9 | 2.399 | 3.413 | 2.200 | 9.239 | 100.000 |
| 10 | 4.593 | 12.839 | 5.883 | 9.989 | 20.359 |
| 11 | 5.113 | 14.434 | 6.244 | 10.129 | 19.170 |
| 12 | 8.736 | 42.895 | 13.636 | 13.765 | 11.673 |
| 13 | 7.978 | 45.669 | 13.523 | 14.062 | 11.506 |
| 14 | 11.518 | 71.164 | 21.835 | 100.000 | 9.239 |
| 15 | 6.891 | 34.791 | 10.861 | 12.181 | 13.032 |

By ordering the individuals according to their similarity to image 14, we obtain the following list: {14, 8, 13, 12, 15, 4, 5, 7, 0, 11, 10, 6, 3, 1, 2, 9}. This ordering seems to be correct, the major deficiency being that individual 7 is considered less similar than individuals 4 and 5. When comparing to image 9, we get the ordered list: {9, 2, 1, 3, 6, 10, 11, 0, 7, 5, 4, 15, 12, 13, 14, 8}, which also appears to be approximately correct; image

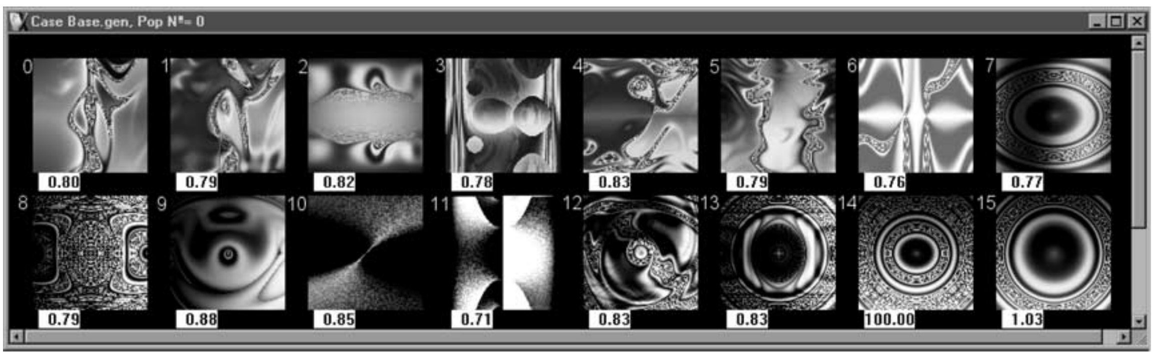


*Figure 16.* In the image above we present a subset of the images belonging to the database. The numbers bellow the image are presented as a curiosity, and indicate the *rmse* similarity to image 14. According to this metric the closest image is image 15, which is good, and the second closest is image 9, which is bad.

9 is characterised its fluid and organic forms, and so are individuals 2, 3, 6 and 0, and, although in a lesser degree, individuals 10 and 11.

Although the experimental results are still preliminary, the seeding procedure based on the above described similarity metric seems to produce good results. We use the similarity as fitness for the seeding process, and roulette wheel selection for choosing which images will become part of the initial population. We do not allow the repetition of images.

One of the main advantages of the similarity metric is that the complexity measures are static. Therefore, we store the $IC$, $PC_1$ and $PC_2$ values of the images belonging to the database. When we want to compare these images to a new one, we only need to calculate the complexity measures for that image, and apply the similarity formula. When we used *rmse similarity*, we were forced to compare the images of the database to the new one pixel by pixel, which is, of course, a computationally expensive task.

The experimental results achieved so far indicate that the comparison of images based on their characteristics, namely on their complexity, is adequate. This, of course, suggests that taking into consideration other types of features of the images (e.g. edges, colour, outline, etc.) may also prove useful.

It is also important to notice that the compression methods described can be applied to any image. Therefore we can compare the database images with images that were not generated with NEvAr. We still haven't explored this possibility, nevertheless we believe that it can produce interesting results.

### 4.2. Automatic Evaluation

In this section, we will describe our current approach to the automation of the evaluation stage. We are still in a very early stage of development, therefore the approaches and formulas presented can, and probably will, be subjected to changes as the research progresses. We will start by describing the filtering methods that we currently use. Afterwards, we will present an approach to automatic fitness assignment.

***4.2.1. Filters.*** The goal of the filtering layer is to discard individuals that are unquestionably bad. We use two types of filters: one works at the genotype level and the other at the phenotype level.

The generation of images that are either too simple or too complex, e.g. completely blank or *noise* (i.e. completely random), is frequent during the first populations of an *experiment* (see Fig. 7). As the population number increases, these images become less frequent, but still occur (see Fig. 11).

At its current state, the phenotype filter tries to tackle this problem. To do so, we calculate the image complexity of the individuals belonging to the population (i.e. the $IC$ measure), and discard images with $IC$ values outside a given interval. The user can specify lower and upper limits for this interval and, therefore, adjust the filtering level. This method is quite efficient during the first populations. In Fig. 17, we present a typical initial population and indicate which images would be discarded if the filter were turned on.

The main drawback of this approach is that it is time consuming, since the individuals must be rendered in order to calculate their $IC$ value. To cope with this disadvantage we are trying to develop a filter that works at the genome level.

The development of a genotype filter is a complex task. Currently, our filtering method is extremely limited. Basically, we verify if the variables $x$ and $y$ are both present (if none of them is present, the pixels of the image will all have the same value; if only one

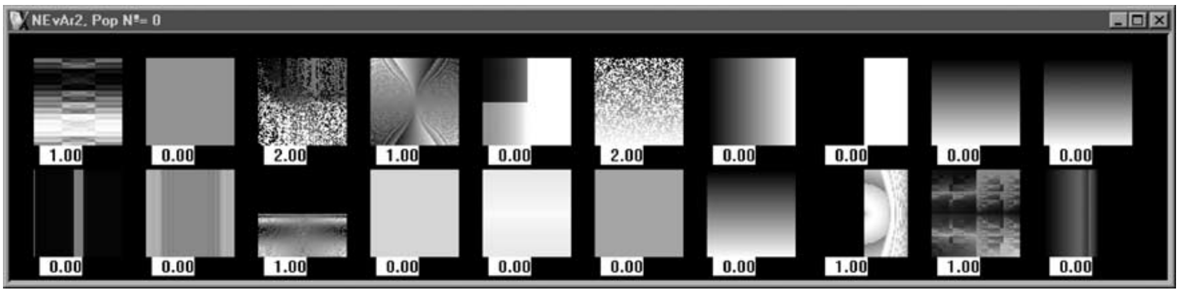


*Figure 17*. The numbers bellow the images indicate if they were discarded or not: zero indicates insufficient complexity (in this case $IC < 1$); two indicates excessive complexity (i.e. $IC > 10$); a value of one indicates that the image complexity is within the specified interval.

is present, the image will be composed by vertical or horizontal lines) and check if the root of the tree is an appropriate function (e.g. a noise generation operator at the root will result, unavoidably, in *noise*).

We are unsure about the future of the genotype filter, at least as a filter. One of the hypothesis is using Machine Learning techniques to try to elicit useful combinations of expressions, and then check for the existence of such combinations. The inclusion of *intron* removal and code optimisation techniques may prove useful in this task. However, even if we consider that useful sub-trees can be identified, their presence doesn't imply high image quality and neither does their absence imply low quality. Nevertheless, the elicitation of these sub-trees can be useful to other tasks, for instance, these trees my become part of the function set of NEvAr, i.e. they can become primitives of the system, thus avoiding the need to rediscover them.

***4.2.2. Automatic Fitness Assignment.*** Our initial idea for automating the evaluation assignment was to train a neural network (NN), and use it to assign fitness. However, disappointing results from an early work using NN [26] refrained our enthusiasm concerning this approach.

The success of the previously described seeding mechanism renewed our interest in automatic evaluation. Considering that you have a good way to compare images, which apparently we have; and that this comparison is based on the characteristics of the images and not on the images themselves, which is also true; then you can compare the characteristics of the images of the population with the characteristics of "good" images. Thus, it makes sense to use the similarity metric as a basis for fitness assignment.

Consequently, we devised a formula to assign fitness based in the $IC$ and $PC$ estimates earlier described. This formula is related with our personal beliefs about aesthetics. Our point of view is that the aesthetic value of an image is connected with the sensorial and intellectual pleasure resulting from its perception. It is also our belief that we tend to prefer images that are, simultaneously, visually complex and that can be processed (by our brains) easily. We will resort to an analogy, in an attempt to clarify our previous statement: a fractal image is usually complex, and highly detailed; yet it can be compactly described by a simple mathematical formula. In the same way, there are images which are visually complex and that can be represented compactly by our brain. Returning to the fractal example, the self-similarity can make fractal images easier to process, which, from our point of view, gives an explanation to why we usually consider this type of images interesting.

We won't try to justify our beliefs about aesthetics, basically because we lack sufficient experimental evidence to support them. We will, however, present the formula that we currently use to automate fitness and the experimental results achieved so far.

In the construction of our formula, we assume that fractal image compression is closer to the way humans process images than *jpeg* compression. Therefore, we will use $IC$ as an estimate of visual complexity and $PC_1$, $PC_2$ as estimates of processing complexity. The act of seeing is not instantaneous, it takes a (sometimes-long) interval of time. Hence, it is necessary to take into consideration the way our perception of the image changes through time. Our fractal encoding method makes a quad-tree partition of the image. In $PC_1$, the tree can have one more level than in $PC_2$. The image is, therefore represented with more detail. We will consider $PC_1$ and $PC_2$ as estimates of the processing complexity in different points in time ($t_1$ and $t_0$, respectively). From our point of view, a moderate increase in the amount of detail represented should be accompanied by an also moderate increase in the representation size, thus $PC(t_1)$ and $PC(t_0)$ should be as close as possible. By combining our ideas into a formula, we obtain:

$$\frac{IC^a}{(PC(t_0) * PC(t_1))^b * \left(\frac{PC(t_1) - PC(t_0)}{PC(t_1)}\right)^c} \tag{4}$$

the exponents $a$, $b$, and $c$ are used to change the weight of each of the factors. The division by $PC(t_1)$ is necessary to "normalise" the subtraction result.

We used this formula to assign fitness, the parameters $a$, $b$ and $c$ were set to 1, 0.4 and 0.2 respectively. Additionally, we imposed upper bounds for $IC$, and lower bounds for $PC(t_1)$, $PC(t_0)$, and for their subtraction. These lower bounds vary, from experiment to experiment, but were kept constant throughout each particular one. During the experiments, the filtering layer was inactive. The initial populations were randomly generated. We used two different population sizes, 20 and 40, but this didn't seem to have any influence in the results. We used roulette wheel selection, a crossover rate of 90% and a mutation rate of 10%. The evolution strategy is non elitist.

In Fig. 18, we present the best individuals, according to the automatic fitness procedure, from several independent runs.

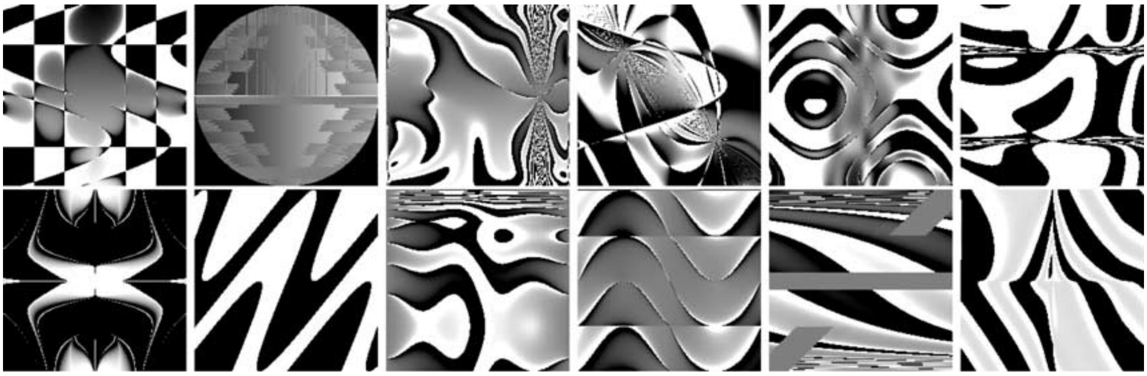

*Figure 18*. Best individuals from several independent runs.

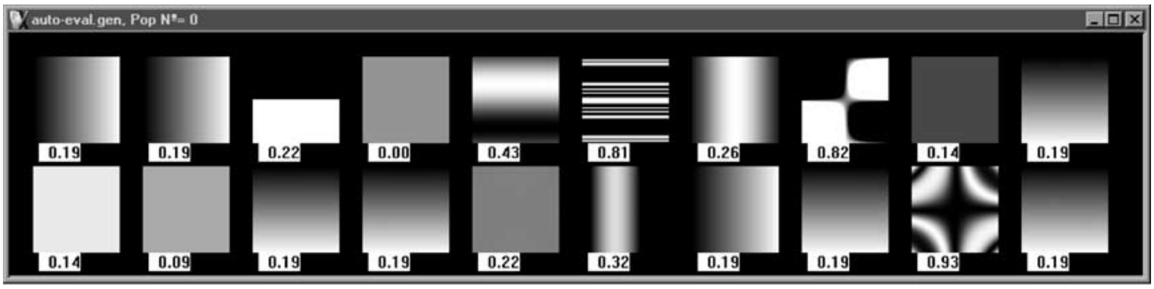


*Figure 19*. Population 0 of an experiment guided by the earlier described fitness function. The numbers bellow the images are the fitness values assigned to them.

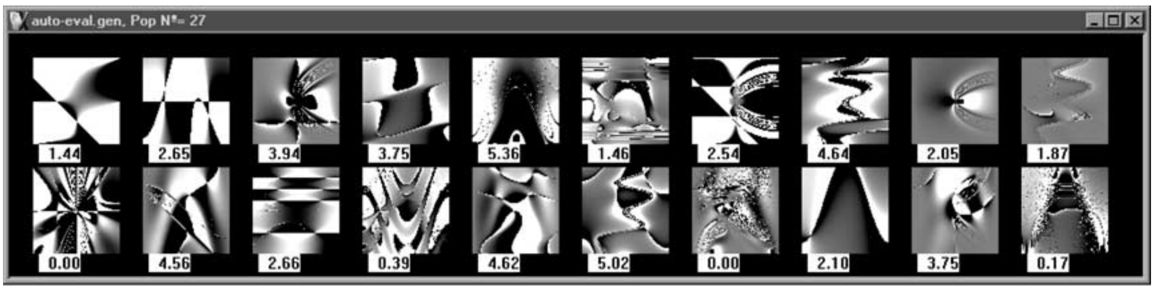


*Figure 20*. Population 27 of the experiment of Fig. 19. The numbers bellow the images are the fitness values assigned to them.

In Fig. 19, we present the first population of a particular experiment and the fitness values assigned to the individuals. Figure 20 shows population 27 of the same experiment. The best individuals of populations 0 to 29 are presented in Fig. 21. The best overall individual (of the first 30 populations) was found in population 27.

We consider the experimental results achieved through the use of formula 4 to be extremely promising. In fact, they widely exceeded our expectations, specially if we take into consideration that, when fitness is assigned randomly, the system converges to images that are either blank or noise. This happens even when the first population is not randomly generated. The major drawback of our approach is that, currently, it only allows the evolution of greyscale images.

Before finishing this section, and in order to prevent misinterpretations we want to stress the following: we don't intend to say that our perception system works

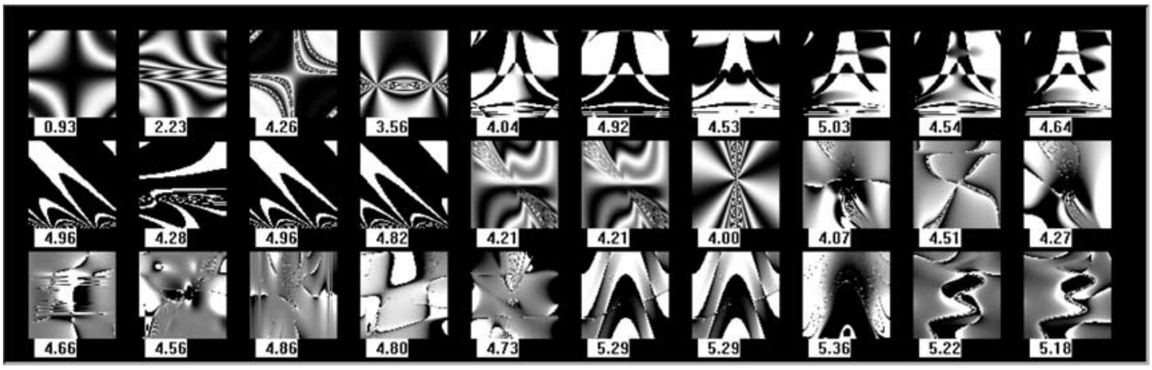


*Figure 21.* The best individuals of populations 0 to 29 (from left to right and top to bottom). The best individual was found in population 27.

by fractal compression, neither that the visual complexity of an image can be measured by the quality of *jpeg* compression. Instead, we suggest the use of the *IC* and *PC* as rough estimates for visual complexity and processing complexity. We also don't intend to say that our formula can fully evaluate the aesthetic value of images, in fact we believe that art cannot be created nor assessed in disregard of the cultural context. In other words, we do believe that cultural issues affect our aesthetic judgement. However, we also defend that the aesthetic value is, to some extent, linked with the complexity of the image and with the mental work necessary to its perception.

## 5. Conclusions

We consider that NEvAr is, from the artistic point of view, a tool with great potential. The use of NEvAr implies a change to the artistic and creative process. The artist is no longer responsible for the generation of the idea. Instead, the idea emerges from an evolutionary process, in which artist and tool interact. In spite of these changes, the artworks produced still obey to the aesthetic principles of the user. Therefore, the artist can express her/himself through the use of the tool and review her/himself in the works created.

Our experimental results (Section 3) indicate that it is possible to create interesting images without resorting to high level primitives, showing the inaccuracy of the idea that the generation capabilities of an Evolutionary Art Tool depend on the use of this type of primitives.

Our focus on the reuse of previously generated individuals led us to the study of seeding procedures. In Section 4.1, we presented our current approach to automatic seeding. The use of a CBR based approach and the development of a similarity metric, which compares the characteristics of the images and not the images themselves, appears to be very promising as the preliminary results show (4.1).

We also presented our current approach to automatic fitness assignment. Our research in this area is still on a very early stage, but the experimental results achieved so far (4.2.2) exceeded our expectations, and seem to indicate that our approach is useful.

When we think in Evolutionary Art Tools and in automatic fitness assignment, the idea of performing aesthetic judgements always comes to mind. There are, however, other possibilities to be explored. One can, for instance, try to devise a way of recognising some type of image (e.g. faces, cars, flowers) and use it to guide the evolutionary algorithm. In the future, we intend to explore this kind of possibilities, since we think that they can provide interesting and useful results.

## Notes

1. There also is a "special" type of terminal node that returns the pixel values of an image loaded from disk. This type of node is used to evolve "special effects" which can be applied to any image. Although this feature wasn't thoroughly explored we will present some experimental results in Section 3.1.
2. This method is similar, at least apparently, to the one used in [14]
3. In fact, it is possible to show that NEvAr can generate any image. This can be trivially demonstrated resorting to the standard GP *if* operator, which belongs to the function set of our system. Through the use of this operator the image can be partitioned in increasingly smaller blocks. The image corresponding to each block is the result of a different symbolic expression. We can, therefore, divide the image until we have a different symbolic expression for each of its pixels. By making each of the expressions equal to the 3d-vector representing the value of the corresponding pixel, the image is generated. It is also possible to show that any image can be generated without relying on partitioning, but this isn't so easy.

## References


1. J. Dewey, *How We Think*, D.C. Heath: Boston, 1919.
2. E. De Bono, *El pensamiento lateral. Manual de la creatividad*, Paidós: Barcelona, Spain, 1986 (in Spanish).
3. J.P. Guilford, *The Nature of Human Intelligence*, McGraw-Hill: New York, 1967.
4. G. Wallas, *The Art of Thought*, Harcourt Brace: New York, 1926.
5. D. Goldberg, "The design of innovation: Lessons from genetic algorithms, lessons for the real world," Department of General Engineering, University of Illinois at Urbana-Champain, IlliGAL Report N° 98004, February 1998.
6. J.H. Holland, *Adaptation in Natural and Artificial Systems*, MIT Press: Cambridge, MA, 1992.
7. J. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*, MIT Press: Cambridge, MA, 1992.
8. D. Horowitz, "Generating rythms with genetic algorithms," in *International Computer Music Conference*, Aarhus, Denmark, International Computer Music Association: San Francisco, CA, 1994.
9. D. Ralley, "Genetic algorithm as a tool for melodic development," in *International Computer Music Conference*, Banff, Canada, 1995, International Computer Music Association: San Francisco, CA, pp. 351–352.
10. J. Biles, "A genetic algorithm for generating jazz solos," in *International Computer Music Conference*, Aarhus, Denmark, International Computer Music Association: San Francisco, CA, 1994.
11. B.L. Jacob, "Composing with genetic algorithms," in *International Computer Music Conference*, Banff, Canada, 1995, International Computer Music Association: San Francisco, CA, pp. 452–455.
12. G. Wiggins, G. Papadoupoulos, S. Phon-Amnuaisuk, and A. Tuson, "Evolutionary methods for musical composition," *International Journal of Computing Anticipatory Systems*, Chaos: Liège, Belgium, 1999.
13. R. Dawkins, *The Blind Watchmaker*, W.W. Norton & Company: New York, 1987.
14. K. Sims, "Artificial evolution for computer graphics," *ACM Computer Graphics*, vol. 25, Addison-Wesley: Boston, MA, pp. 319–328, 1991.
15. S. Rooke, "The evolutionary art of steven rooke," http://www.azstarnet.com/~srooke/, 1996.
16. J. Ventrella, "Animated artificial life," in *Virtual Worlds—Synthetic Universes, Digital Life, and Complexity*, edited by J.C. Heudin, Perseus Books: Cambridge, MA, pp. 67–94, 1999.
17. E. Baker, "Evolving line drawings," Harvard University Center for Research in Computing Technology, Technical Report TR-21-93, 1993.
18. J. Graf and W. Banzhaf, "Interactive evolution for simulated natural evolution," in *Artificial Evolution*, edited by J.-M. Alliot, E. Lutton, E. Ronald, M. Schoenauer, and D. Snyers, vol. 1063, Springer Verlag: Berlin, pp. 259–272, 1996.
19. P. Bentley (Ed.), *Evolutionary Design by Computers*, Morgan Kaufman: San Francisco, CA, 1999.
20. P.J. Angeline, "Evolving fractal movies," in *Genetic Programming 1996: Proceedings of the First Annual Conference*, 1996, edited by J.R. Koza, D.E. Goldberg, D.B. Fogel, and R.L. Riolo, MIT Press and Stanford University, CA, USA, pp. 503–511.
21. A. Horner and D. Goldberg, "Genetic algorithms and computer-assisted composition," in *Genetic Algorithms and Their Applications: Proceedings of the Fourth International Conference on Genetic Algorithms*, 1991, Morgan-Kaufman: San Francisco, CA, pp. 427–441.
22. R.A. McIntyre, "Bach in a box: The evolution of four-part Barroque harmony using genetic algorithms," in *IEEE Conference on Evolutionary Computation*, Florida, USA, 1994, IEEE Press: New York, pp. 852–857.
23. L. Spector and A. Alpern, "Criticism, culture, and the automatic generation of artworks," in *Proceedings of Twelfth National Conference on Artificial Intelligence*, 1994, AAAI Press/MIT Press, Seattle, Washington, USA, pp. 3–8.
24. L. Spector and A. Alpern, "Induction and recapitulation of deep musical structure," in *Proceedings of International Joint Conference on Artificial Intelligence, IJCAI'95 Workshop on Music and AI*, Montreal, Canada, 1995, Springer-Verlag: Berlin.
25. P. Hodgson, "Understanding computing, cognition, and creativity," in *AISB Symposium on Musical Creativity*, edited by G. Wiggins, Edinburgh, UK, 1999.
26. S. Baluja, D. Pomerlau, and J. Todd, "Towards automated artificial evolution for computer-generated images," *Connection Science*, vol. 6, Taylor and Francis: London, pp. 325–354, 1994.
27. W. Latham and S. Todd, *Evolutionary Art and Computers*, Academic Press: Winchester, UK, 1992.
28. Y. Fisher, *Fractal Image Compression—Theory and Application*, Springer-Verlag: Berlin, 1994.