Genetic Art in perspective

Rachele Bellini and N. Alberto Borghese

Department of Computer Science
Università degli Studi di Milano

Abstract. Since the pioneer observations of Alan Turing, emotional and aesthetical capabilities have been considered as one of the fundamental element of a genuinely intelligent machine. Among the proposed approaches, genetic algorithms try to combine intuitively a generative impulse with a critical capacity that steers the production towards a valuable goal. The approach here presented is based on Karl Sim's approach in which a set of possible primitives is defined and it represent the genotype of the system. Such expressions are combined using genetic algorithms rules to obtain more complex functions that describe new images. At each step, images are evaluated by the user and this implicitly drives the evolution process. Results can be impressive, however a clear understanding of the determinants of our aesthetic evaluation is presently beyond reach.

1 Introduction

When Alan Turing in the forties of last century proposed the reasons for which a machine could never become intelligent, the lack of creativity and of aesthetic sense was one of the main arguments [1]. The introduction of genetic art in the eighties sounds like a counter-example of this, but, although results have been impressive the understanding of underlying mechanisms is still far beyond reach.

Genetic (or evolutionary) art is a type of digital art in which the artwork is created by a genetic algorithm [2, 3]. Goal is to achieve an aesthetically valuable digital artwork. The final result is usually mathematically described by an ensemble of functions, but the process of its creation involves the human judgment and taste.

We review here the pipeline of creating artistic images through genetic algorithms and highlight some critical points.

2 Algorithm Description

The algorithm starts from a set of images usually randomly generated from the ensemble of available functions. These images constitute what are named parents of the first generation [4, 5]. A new set of images, called son images, are generated from each parent image by transforming each pixel of each image through a function obtained combining simple functions with genetic operators.

The resulting images are judged by a human user who implicitly assigns a fitness value to each of them that represents the personal degree of preference according to the image aesthetic value or interest. Images are ordered with respect to their fitness and the best images are chosen as parents for the next generation.

The functions that define the child-images are composed starting from the functions that created the parent image: they inherit all of the sub-functions and the operators of the parent-image, except for one of them (a function or operator), extracted randomly, which is replaced from a new one (function or operator) also randomly extracted from the database. In such way, an evolution of the images can be envisaged. The process can be iterated ad libitum, evolving the images from one generation to the other, until a satisfactory image is reached or the user decides to interrupt the evolutionary process. Some examples of this evolutionary process are provided in the Results section.

2.1 Images definition

Each image is digitized into a N x N pixels. If color images are considered, the value of each pixel will be its RGB value, where each channel is coded over 8 bits for a total of 24 bits per pixel. If gray level images are considered, the value will be the pixel gray value discretized over 8 bits: from 0 to 255. The considered images are N x N, with N = 300 pixels. Each image is described by a 2-dimensional mathematical function, which is sampled in the center of the pixels of the image.

2.2 Sampling modalities

Several functions and sampling strategies can be used. In general, the selected functions are those trigonometric combined with power functions (cf. Table 1). For sake of simplicity, trigonometric functions are defined with unitary frequency. Here we show three different sampling approaches, each of them having different aesthetic features. A first approach consists in choosing a unitary sampling step, s = 1. In this case, the trigonometric components will have a period of 2 π and this approach leads to images characterized by a high-frequency texture, as shown in Figure 1.

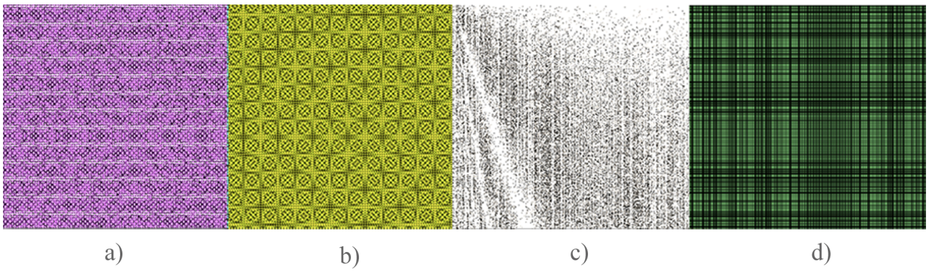

Fig. 1. An example of a set of four images generated with functions with sampling step size s = 1: a) sin(x*y)*cos(x*y)/cos(y)+sin(y)+cos(x)+sin(x); b) tan(cos(sqrt(x*y*x*y))); c) sin(x*y*x)+cos(y*x*y)/sin(x*x)/sin(y/x); d) sin(x*x)*cos(y*y).

A lower frequency content can be obtained if the functions are sampled with a less dense granularity. We take a sampling step, s = 1 / N. This approach leads to much more smoothed images, sometimes just gradients, aesthetically very different from the ones produced with the former approach (see Figure 2).

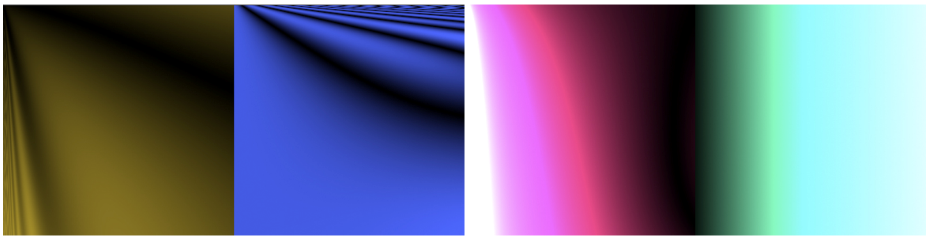

Fig. 2. An example of a set of four images generated sampling the same functions of Fig. 1, with a step size of 1 / N: a) cos(sqrt(x+y))*y+sqrt(cos(y)*sin(x))*sin(y/x)+y-sin(x); b) sin(x*y*x)+cos(y*x*y)*cos(x/y); c) sin(x*y*x)+cos(y*x*y)-sqrt(abs(x)+abs(y))/abs(x); d) abs(x*3).

In a half-way approach we evaluate the function with a period of the trigonometric component approximately equal to the image width, i.e. s = 2 π / N. We have chosen this step size because of its more pleasant aesthetic results. In this paper, all the following images are generated sampling the functions with a step size s = 0.02, that approximates this condition (Figure 3 shows an example of this).

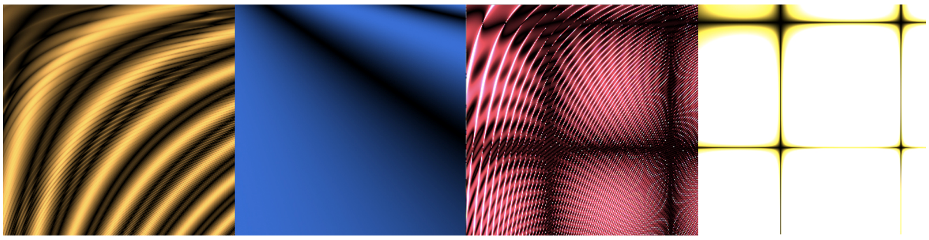

Fig. 3. An example of a set of four images generated evaluating the same functions of Fig. 1 with a sampling step, s = 0.02:
a) sin(abs(cos(x+y))+abs(cos(y*x*y)))+tan(cos(sqrt(x*y*x*y))); b) cos(x/y); c) sin(x)*cos(y)/sin(x*y*x)+cos(y*x*y)/abs(x); d) ((x+y)*y*x*sin(x)*cos(y)).

2.3 Intensity adaptation

The computed pixel values have to be adapted to the image range. No constraint is made about the values assumed by the generating functions: trigonometric functions can assume both positive and negative values and polynomial and absolute value functions can have a maximum value that may either not span the entire range of 255 values or exceed this range.

Moreover, at each iteration, only one function is selected to produce each image. Different scaling values, that we will name “color weight”, will be applied to the R, G and B channels separately to obtain a color image. When the color weight is equal on the three channels, the image will appear in gray levels (Fig. 4), otherwise a color image is produced, whose tone changes at each generation (Fig. 5). To obtain this, the color weight of each channel is randomly generated inside the interval between 0 and 255. The value of that channel for all the pixels is multiplied by the color weight of that channel. Values that exceed 255 are clipped to 255 and values below zero are clipped to 0. Different images will have different triplets of color values.

Besides creating color images, color weights have also the aim of spreading the function value in the useful range of each color channel.

Fig. 4. A set of gray-scale images generated weighting the three color channels equally.

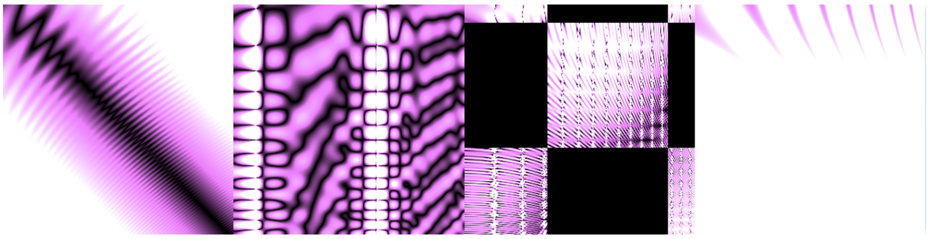

Fig. 5. A set of images generated with the same color-weights: the red and blue weights are set to 255, while the green one is set to 100.

2.4 Function generation and the genetic operators

The function applied to the pixels of the parent images is obtained as a combination of a variable number of sub-functions (from a minimum of one to a maximum of five sub-functions), chosen among those defined in Table 1. Such functions are mainly trigonometric and polynomial functions of x and y and were defined experimentally by trial and error [6].

```
sin(x*x + y*y)
sin(x*x)*cos(y*y)
sin(x/y)*cos(x/y)
cos(x/y)
sin(y/x)
abs(y)-x
x+abs(y)
cos(x*x/y)+sin(y*y/x)
sin(x)+sin(x)+cos(y)+cos(y)
cos(x)+cos(x)+sin(y)+sin(y)
sin(x)+cos(x)+sin(y)+cos(y)
cos(y)+sin(y)+cos(x)+sin(x)
tan(cos(sqrt(x*y*x*y)))
```

```
abs(x)
abs(y)
abs(x)*abs(y)
sin(x)*cos(y)
sin(x*y)*cos(x*y)
sin(x*x-y*y)
sin(x*x)
y-abs(x)
y-sin(x)
x-cos(y)
abs(x)+y
sin(x*x*x-y*y*y)
sin(y*y*y)+sin(x*x*x)
cos(y*y*y+x*x*x)
cos(y*y*y)+cos(x*x*x)
abs(y*3)
abs(x*3)
sin(x*x/y-y*y/x)
sqrt(abs(x)+abs(y))
sin(x*y*x)+cos(y*x*y)
sin(sqrt(abs(x)))-cos(sqrt(abs(y)))
sqrt(cos(x)+sqrt(x)*sin(y)+sqrt(y))
cos(x)*sin(x*y)
cos(y)*sin(x*y)
sin(x+y*x*y+x*x)
sin(y+x*y*x+y*y)
abs(x*y+x*x+y*y)
((x+y)*y*x*sin(x)*cos(y))
((x+y*x)+sin(x*y)+cos(y/x))
sin(x*y+x)+cos(y*x+y)
cos(x+y)*sin(x+y)/2
cos(sqrt(x+y))*y+sqrt(cos(y)*sin(x))
sin(sqrt(y+x))*x+sqrt(sin(x)*cos(y))
cos(x)*sin(x)+cos(y)*sin(y)
sin(abs(cos(x+y))+abs(cos(y*x*y)))
sin(cos(x)*abs(y)*abs(y))
cos(x)*sin(y)*cos(x*y)
```

Table 1. The sub-functions used to create the function applied to the pixels of a parent image at each generation.

Once the sub-functions are extracted, these are combined through a set of simple operators, restricted here to the basic four operations: multiplication, ‘*’, division, ‘/’, sum, ‘+’ and subtraction, ‘-‘, also extracted randomly. Therefore, the final function that is applied to each parent image is created joining the chosen sub-functions with the chosen operators.

A simple example of the obtained functions is shown in Fig. 7. Here the sub-functions randomly extracted are: `sin(x/y)`, `cos(x/y)` and `abs(y*3)` and the extracted operators are ‘*’ and ‘-‘. Therefore the final composed function is the following: `sin(x/y)*cos(x/y)-abs(y*3).`

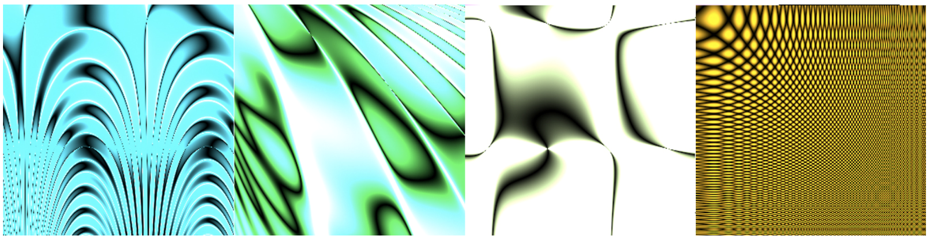

Fig. 6. Some examples of obtained functions. From left to right: a) sin(x)+sin(x)+cos(y)+cos(y)/sin(cos(x)*abs(y)*abs(y)) with the color weights [25, 218, 238]. b) cos(x)+cos(x)+sin(y)+sin(y)+cos(x/y)/cos(x*x/y)+sin(y*y/x) with the color weights [47, 252, 110]. c) y-sin(x)/cos(y)+sin(y)+cos(x)+sin(x)/cos(x)+cos(x)+sin(y)+sin(y) with the color weights [123, 144, 105]. d) cos(y*y*y)+cos(x*x*x) with the color weights [200, 151, 21].

3 Results

In nature there is no aesthetic rule that binds somatic aspect between parents and their children. Even if some basic characteristics are expected, children can be either extremely similar to their parents, or very different.

This phenomenon is reproduced with our genetic art algorithm: since any operator and any function can be replaced during the evolutionary process between a parent-image and a child-image, there is no control over the amount of aesthetic change. The amount of change can be sometimes even non-noticeable (Fig. 7), while in other situations the child-image can be aesthetically totally different from the parent one (Fig. 8).

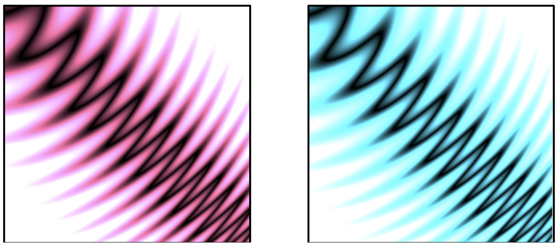

Fig. 7. An example of high-similarity between a parent image (left) and a child-image (right). In this case the function is exactly the same (`y-abs(x)-sin(x*x + y*y)`), while the color weight is different ([204, 82, 131] in the first image, [70, 222, 252] in the second).

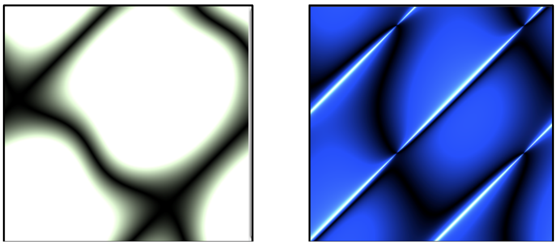

Fig. 8. An example of low-similarity between a parent image (left) and a child image (right), where just an operator is changed. In this case the parent function is: sin(x)+sin(x)+cos(y)+cos(y)+cos(x+y)*sin(x+y)/2 (color weight [104, 66, 104]), while the child one is: sin(x)+sin(x)+cos(y)+cos(y)/cos(x+y)*sin(x+y)/2 (color weight [4, 16, 163]) y-abs(x)/sin(x)+cos(x)+sin(y)+cos(y)+cos(y)*sin(x*y).

As it can be seen, some of the images present some saturated areas in the white and black region. In fact there is no explicit control on saturation when applying the transforming function to the pixels of an image. For instance, we have analyzed the amount of saturated pixels over a set of 40 images and found out that 1/3 of the images has at least 90% of saturated pixels, 1/3 has less than 10% and 1/3 has a percentage between 10% and 90%.

An additional example of the evolution over four generations is shown in the following lines. In this case, only one parent is considered at each generation as the son with the best fitness value. Each parent generates four children images that are evaluated and ranked. The initial random functions extracted for the four images, a-d in the first generation, are the followings:

```
1a. tan(cos(sqrt(x*y*x*y)))+sin(x*x)
1b. y-abs(x)
1c. abs(x*3)-sin(x*x*x-y*y*y)-sin(y+x*y*x+y*y)
1d. y-abs(x)+sin(x)+cos(x)+sin(y)+cos(y)+cos(y)*sin(x*y)
```

The image with the highest fitness was the fourth one (1d) in Fig. 9, which becomes the parent-image for the new generation. The three following sets of functions represent the three generations descending from the former parent (cf. Fig. 9):

```
2a. y-abs(x)/sin(x)+cos(x)+sin(y)+cos(y)+cos(y)*sin(x*y)
2b. y-abs(x)+sin(x)+cos(x)+sin(y)+cos(y)+cos(y)*sin(x*y)
2c. y-abs(x)*sin(x)+cos(x)+sin(y)+cos(y)+cos(y)*sin(x*y)
2d. y-abs(x)*cos(x)*sin(x)+cos(y)*sin(y)+cos(y)*sin(x*y)

3a. y-abs(x)*tan(cos(sqrt(x*y*x*y)))+cos(y)*sin(x*y)
3b. y-abs(x)-tan(cos(sqrt(x*y*x*y)))+cos(y)*sin(x*y)
3c. y-abs(x)-cos(y*y*y)+cos(x*x*x)+cos(y)*sin(x*y)
3d. y-abs(x)-cos(y)+sin(y)+cos(x)+sin(x)+cos(y)*sin(x*y)

4a. y-abs(x)/cos(y)+sin(y)+cos(x)+sin(x)+cos(y)*sin(x*y)
4b. y-abs(x)/sin(x*y*x)+cos(y*x*y)+cos(y)*sin(x*y)
4c. y-abs(x)/sin(x)+cos(x)+sin(y)+cos(y)+cos(y)*sin(x*y)
4d. y-abs(x)/x+abs(y)+cos(y)*sin(x*y)
```

4 Discussion and Conclusion

Overall, the generated images exhibit a high variability and richness also within the same generation: a simple change in one of the function components that realize the transformation function may produce a very similar or different result, depending on the overall function shape and on the component. Such property, observed experimentally, can be object of further investigation to determine which can be the determinant characteristics that make an image more attractive than another. The correlation between the fitness value of an image and its aesthetic value can be a tool that may allow investigating the processes underlying our aesthetic evaluation [7, 8].

This analysis could also allow improving the genetic algorithm introducing some elitist mutation rule for which, as far as the generations progress, some of the subcomponents, those with the most impact, are kept fixed and only the other subcomponents are mutated [5, 9].

From first experimental observations, the heavy use of trigonometric functions has been introduced as these produce gentle oscillating variations that are often interpreted as “motion” and add aesthetic value to the images. The frequency of the oscillations introduced is also important as these should be related to the amplitude of the image: too high or too small frequencies tend to produce less pleasant images.

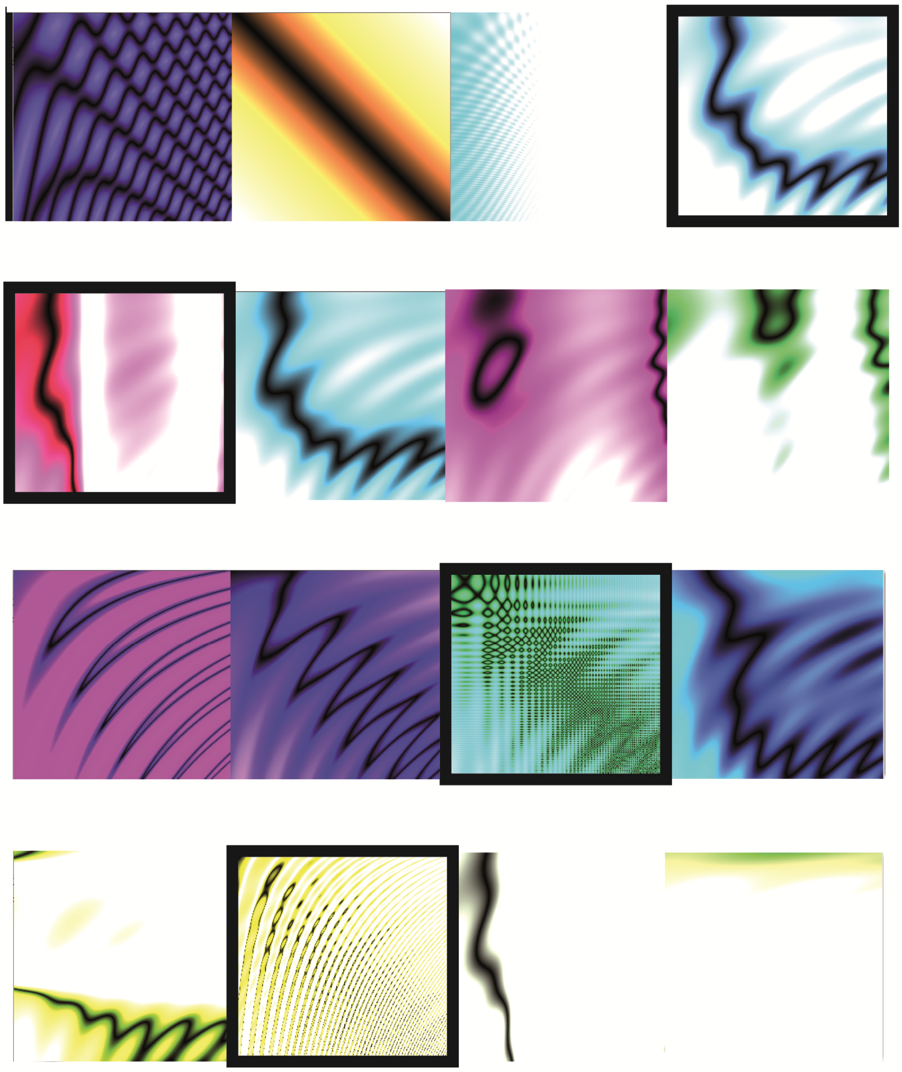

Fig. 9. The images of four different generations. The correspondent functions are presented above. The selected sub-image in each image is highlighted by a black frame. Notice how some parent-child relations are aesthetically very clear, while some others are not.

Genetic algorithms turn out to be a particularly effective tool to generate images with aesthetic value that do not contain structured scenes. The fitness value cannot be easily captured by analytical functions as usually done in the computer science field. This opens the challenge to identify the models, the features and the determinants for our aesthetic evaluation. Besides a comprehension of the mechanisms of aesthetic evaluation, this can add value in all the manufacts in which the external shape can be designed or colored arbitrarily.

References

1. Turing, A.: Computing Machinery and Intelligence. Mind LIX (236), pp. 433–460 (1950)
2. Crow, F., Demos, G., Hardy, J., McLaughlin, J., Sims, K.: 3D Image Synthesis on the Connection Machine. International Journal of High Speed Computing, pp.329-347 (1989)
3. Romero, J., Machado, P. (eds.) The Art of Artificial Evolution: A Handbook on Evolutionary Art and Musi. Springer (2007)
4. Koza, J.: Genetic Programming: On the Programming of Computers by Means of Natural Selection. MIT Press Cambridge (1992)
5. Schmitt, L. M.: Theory of Genetic Algorithms, Theoretical Computer Science 259, pp. 1–61 (2001)
6. http://softologyblog.wordpress.com/category/genetic-art/
7. Galanter, P.: Computational Aesthetic Evaluation: Past and Future. In: J. McCormack & M. d'Inverno (eds.), Computers and Creativity. Springer, Berlin (2012)
8. Lewis, M.: Evolutionary Visual Art and Design, in The art of artificial evolution: a handbook on evolutionary art and music. In: Romero, J., Machado, P. (eds.), pp. 3–37. Springer, Berlin (2008)
9. Cerveri, P., Pedotti, A., Borghese, N.A.: Combined evolution strategies for dynamic calibration of video based measurement systems. In: IEEE Trans. Evolutionary Computation, vol. 5, no. 3, pp. 271-282 (2001)