Evolving Images for Entertainment

Qinying Xu
School of CS and IT
RMIT University
Melbourne, Australia
qinying@cs.rmit.edu.au

Daryl D'Souza
School of CS and IT
RMIT University
Melbourne, Australia
djds@cs.rmit.edu.au

Vic Ciesielski
School of CS and IT
RMIT University
Melbourne, Australia
vc@cs.rmit.edu.au

ABSTRACT

Images are widely used in media contexts such as web design, games and video animation. The process of creating interesting images can be enjoyable if a useful tool is involved. In this paper we describe an interactive image generation tool called IMAGENE, based on genetic programming, which can be used to create novel, surprising, and sometimes stunning, images. A new method for presenting colour images is also proposed, which results in more colourful images being generated. The system permits the user to progressively evaluate and generate new images from previous sets of images. In doing so, the user is also able to control various settings and parameters. From a user perspective, the system has the following qualities which make generating images entertaining: appealing images arise at random, different styles of images can be created by choosing different function settings and families of images which have common characteristics between parents and children can be generated.

Categories and Subject Descriptors

J.5 [**Arts and Humanities**]: Fine Arts; H.5.1 [**Multimedia Information Systems**]: Animations

Keywords

Image generation tool, Genetic Programming, Evolutionary Search

1. INTRODUCTION

In this digital age, the computer is an indispensable media tool. One application is the generation of interesting images for web design, games and animation. These images may be natural ones, such as grass, fur or water surfaces, or synthetic, such as the artificial image backgrounds in video games. Synthetic images can be manually constructed by using computer-aided design tools, or automatically generated. The latter category of images are known as procedural images or procedural textures [5].

Since the computer has no concept of creativity, software support for image generation is typically limited to capturing an artist's ideas visually on the screen using a range of paint and graphics systems. There has been a need for artificial image creation systems. Evolutionary Art is an approach to creating novel, interesting images [10]. In one form of evolutionary art populations of procedural images are generated based on the principle of evolution: better images have a higher chance of surviving as each new population of images is generated. Often, these images are different from the images manually created by humans and the evolutionary process frequently results in the creation of surprising and interesting new images.

The human view of what makes a pleasing or interesting image is subjective and personal, therefore most evolutionary image systems are interactive and rely on a human user for aesthetic judgments and image ranking [6, 13].

Our original goal was to use our existing genetic programming system to generate images like the ones in the published literature, for example [7, 9, 11, 12, 13]. However, as time went on and we collaborated with a number of artists and designers the focus of the work became more on the process of generating images and on how we could make this process more productive and enjoyable.

We have built an interactive evolution system, IMAGENE, which can generate novel and interesting images and allows user control over a range of options and parameters. The system is based on NEvAr [9] and employs the evolutionary technique of genetic programming (GP) [8]. We discovered that the system provides entertainment value as the user becomes involved in the process.

The outline of this paper is as follows. Section 2 presents the context of our work, highlighting the key concepts that contribute to evolutionary image generation. We briefly discuss related work in Section 3, and then describe our system in Section 4. We report the experiences of different user cohorts that have used IMAGENE, in Section 5. Sample evolved images generated via IMAGENE resulting from user experiences are presented in section 6. Finally, in Section 7 we draw some conclusions and point to directions for future research.

2. BACKGROUND

One main direction of evolutionary art is the generation of procedural images or textures that are aesthetically pleasing to users. Several methods of generating procedural images have been proposed, typically using GP to evolve images in accordance with desired user aesthetics. Before we review related work in the area, we provide some background on procedural image generation and GP concepts.

2.1 Procedural Images

Procedural images or textures are produced by procedures or algorithms in which the images are generated on the basis of underlying mathematical formulae [5]. The input to each formula is a pair of coordinates or a pixel location in an

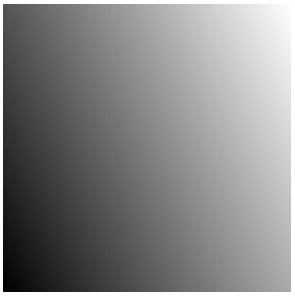


Figure 1: Grey level image rendered from the formula $x + 2 \times y$.

image space, and the output is a corresponding pixel grey level value. Figure 1 shows an example of a procedural image and its corresponding formula, in which the pixel grey level value is defined by a function of (x, y) coordinates. When x and y are close to zero, the value of the function is close to zero and the corresponding pixels will be black, as in the bottom left corner of Figure 1. As x and y increase, the corresponding pixels become lighter. In this example the pixel in position $(28, 28)$ translates to the grey value 84.

For colour images, an RGB (Red, Green, Blue) triple is necessary to display true-colour images. In current colour image evolutionary systems a 3d-vector is used [9, 16]. Each component of a vector corresponds to a different colour channel (Red, Green and Blue). Referring to the grey image formula in Figure 1, if we change the constant 2 to a vector equivalent $[1, 2, 3]$, a colour image formula $x + [1, 2, 3] \times y$ results. We can translate this formula to its vector equivalent $[x + 1 \times y, x + 2 \times y, x + 3 \times y]$. It is clear that each component of the vector is a function of (x, y), which corresponds to one colour channel.

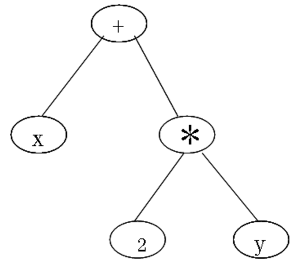
Procedural images have a number of advantages. They can simulate a variety of natural phenomena, including stone, cloud and landscaping effects. Their mathematical nature makes them extremely robust, and there is an infinite variety of formulae. However, it is difficult to comprehend a procedural image formula which combines complex mathematical functions (see Figure 5). It is even more difficult to create a formula that will generate an interesting image. A method of generating formulae that give interesting images is needed. GP is one such method.

2.2 Genetic Programming

In a tree-based GP system each individual is an evolved program, represented as a tree composed of terminals (leaves) and functions (interior nodes). The evolutionary process starts with a random population of individuals and evolves new individuals using operators such as crossover, mutation and evaluation repeatedly. The evaluation of an individual is carried out using a fitness function, an individual with higher fitness has a higher probability of being selected for the next generation.

2.2.1 GP Representation

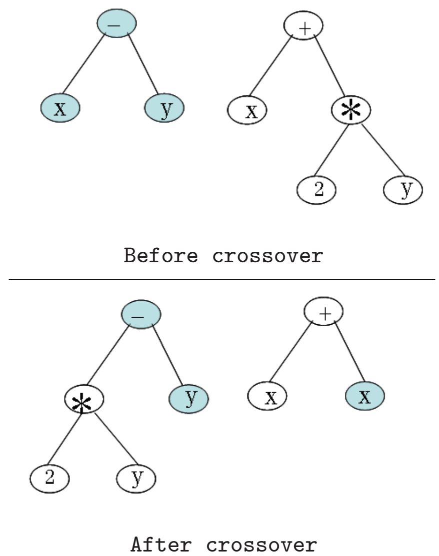
Each image formula is an individual and the number of individuals in each generation is called the population size. Two sets of tree nodes need to be provided to the evolutionary system, the function set and the terminal set. The function set is drawn from the set of common arithmetic operators, such as $+$, $-$, $\times$ and $/$. Pixel coordinate positions (x, y) and constants are used in the terminal set.



`Formula:` $x + 2 \times y$

`Figure 2: A GP tree and corresponding formula`

To illustrate how a GP tree maps to an image formula, Figure 2 shows a program tree which corresponds to the formula and image shown in shown in Figure 1. The first node $+$ has two arguments, x and $2 \times y$, the second argument is also a formula represented as a sub-tree. This whole tree represents the formula $f(x, y) = x + 2 \times y$.



`Figure 3: Example of crossover`

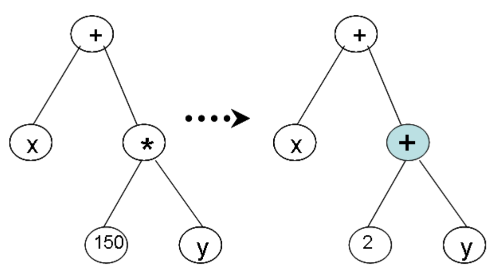


Figure 4: Example of Mutation

2.2.2 Genetic Operators

There are two main types of genetic operator: crossover and mutation. Methods of crossover and mutation differ, depending on the problem and its objectives. We describe the simplest but most widely used methods.

The usual crossover method is to exchange sub-trees between individuals. Figure 3 is an example of crossover. The node x in the left hand tree and the node $*$ in the right hand tree have been selected as crossover points. The children are generated by swapping the corresponding subtrees.

For the mutation operator, a node (function or terminal) is randomly selected and then replaced by a randomly generated tree. Figure 4 is an example of mutation, node $*$ in the left hand tree is replaced by the sub-tree $2 + Y$.

2.2.3 Interactive GP Algorithm

The GP evolutionary process is based on crossover, mutation and guided by a fitness function. In an interactive GP system, fitness evaluation is provided via user interaction with the system. For each generation, the user rates the fitness of the individuals in the current population before the system proceeds to generate the next population. Once the user is satisfied with a population of individuals, the evolutionary process is terminated. The interactive GP algorithm is as follows:

- STEP 1: Initialize the population of randomly generated GP trees. Ask the user to rate the fitness of the individuals in the initial population.
- STEP 2: Repeat until solution is found:
  - Repeat until new population generated:
    - With a given probability select two individuals based on their fitness and apply crossover to generate two offspring. Add them to new population.
    - With a given probability select one individual based on fitness and apply mutation. Add it to the new population.
  - Ask the user to rate the fitness of individuals in new population.

3. RELATED WORK

There are several existing systems using genetic programming to evolve images [4, 17]. The goal of these systems is to find better images during evolution with the better images being assigned higher fitness values. The fitness assignment guides the evolutionary process and plays a significant role in these systems. However, it is not an easy task to develop an appropriate fitness function in the field of image generation. In the recent past, two major directions for image evolutionary systems have emerged, based on different fitness evaluation methods: interactive image evaluation and automatic image evaluation.

As stated in section 2.2.3, interactive systems are based on user supervision [13]. Interactive image evolution takes users' inputs as the evaluation of fitness, as in the NEvAr system [9], wherein a user views and ranks generated images from the population interactively. A skilled user can guide the evolutionary process to produce desirable images [9].

```
RED Program:
(cos (+ (log (twice (+ (log (tan (cos (- (twice
(sqrt 64.10)) (sin (+197.57 X)))))) (* (Y
42.43)))) (sin (+ (sqrt (/ (sin (/ (sqrt (sin
106.44)) Y)) (* (twice (* (X (sin 71.83))) (* (sin
(- (log Pi) (/Y 129.63))) (/ (/ (* Pi Pi)189.50)
(triple Y)))))) (sqrt X)))))

GREEN Program:
(cos (log (* (sqrt (log (sin (- (sin (* (sqrt (/
X 173.57)) (sqrt Pi))) (sqrt (+ Pi (- (/Y X) (+
10.62 Pi)))))))) (cos (triple 27.30)))))

BLUE Program:
(cos (+ (log (twice (+ (log (tan (cos (- (twice
(sqrt 64.10)) (sin (+ 59.97 X)))))) (* (Y
42.43)))) (sin (+ (sqrt (/(sin (/ (sqrt (sin
106.44)) Y)) (* (twice (*X (sin 71.83))) (* (sin
(- (log Pi) (/Y 189.50))) (/ (/ (* Pi Pi)189.50)
(triple Y)))))) (sqrt X)))))
```

Figure 5: Colour image formulae in IMAGENE

More recently, automatic image evaluation, which removes the users from the manual process, has been investigated [2, 14, 18]. The Gentropy system [18] uses image features such as colour, luminosity, and wavelet coefficients in its fitness function. More recently, some aesthetic models [11, 15] which use different measurements to score images have been developed. These systems generate some appealing images without user involvement.

4. THE IMAGENE SYSTEM

Research into automatic image evaluation systems has achieved some level of success, however, as users' individual views are often personal and subjective, automatic evaluation is not an easy task. In comparison, a human can easily identify and assess an image. Moreover, the interaction between a user and a good image evolutionary system can be an entertaining process, as the user sees the random images in the initial generation develop into more and more interesting ones as the evolution proceeds.

Interactive image generation systems require users to view and rank generated textures continuously. This is impractical when the population size is large or many generations are needed. IMAGENE uses a small number of individuals and a few evolutionary steps. For grey level images eight individuals are used to give eight grey images which are shown on one screen. In contract to other systems such as NEvAr which uses a vector to represent the RGB components of an image, IMAGENE uses three individuals to represent one colour image.

4.1 A New Method for Colour Image Generation

As stated in section 2.1, most GP colour image systems use a 3d-vector to represent an RGB value. We use a new method to create colour images. Three individuals are used to represent one colour image, each individual giving a different colour channel (Red, Green and Blue). An example of three colour image formulae is given in Figure 5. The corresponding tree for the green program is shown in Figure 6.

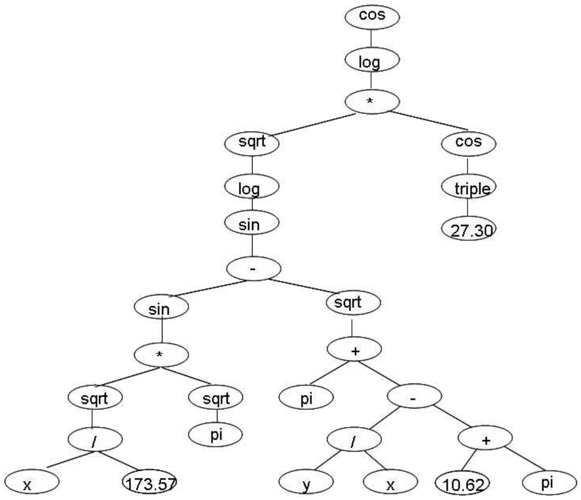


Figure 6: GP tree of the Green Program in Figure 5

Figure 10a is the corresponding image. In this example, all the constants such as 27.30 and 173.57 were randomly generated during evolution.

This method results in colourful images which contain many intricate patterns. While our method needs three times the population size of other methods, we show the same number of images to the user in each generation, so there is no extra work for users.

4.2 Image Rendering

The output of an image formula is likely to be outside the range of grey level or colour pixel values accepted by standard image display programs. For instance, for an input of $(100, 125)$ the output of the formula $x + 2 \times y$ (Figure 1) is 350. This is beyond the allowable range for a display program that accepts values up to 255. We use the following method to convert the outputs of formulae to the range [0,255]:

- Store the output for each pixel position in a 2D array.
- Find the largest value and smallest value in this array, then calculate the difference.
- Divide each value in the array by the difference and then multiply by 255.
- Use these new values as the pixel's grey or RGB value.

Our system is able to map a formula to symmetric and non-symmetric renderings of images, that is, one formula can depict two images. In non-symmetric renderings, the origin $(0, 0)$ is the top left corner of the image, whereas for symmetric renderings the origin is the centre of image. For symmetric renderings the absolute value of x and y is input to the image formula. By providing two alternatives for each formula the system is able to present images in two distinct ways at each generation, thereby allowing the user to find a greater variety of images, and more quickly.

4.3 User Involvement

A trained user can guide our system in an extremely efficient way. The user need not have background or knowledge in computer programming or art, they will know how to control the evaluation criteria after a few runs. In some runs it is no longer possible to find interesting images. This usually means that the population has lost diversity and user should restart IMAGENE.

The user has control over three aspects of the system: Evaluation, creation of the initial population and parameter setting.

4.3.1 Evaluation

At each generation, the user will view and rank newly generated images. The user is presented with a window of 8 images. The user is required to enter up to 8 numbers which give the user's ranking of the images. The more highly ranked images will have a higher chance of being selected by the genetic operators in creating the next generation. It is not necessary to rank all the images. Users may rank only the images of interest to them. IMAGENE will automatically give a lower ranking to the remaining images. A experienced user knows how to select potential individuals as parents, a selection which will result in desirable offspring images. The newly generated child images will usually have some features in common with parent images. We denote such parents and children images as a family of images. One such family is shown in figure 11.

4.3.2 Creation of initial population

By default IMAGENE creates the initial population of images by randomly generating GP trees (see section 2.2). That is, the tree shapes, functions and terminals that characterize each tree are randomly selected by the system. Thus, the initial population of images will typically appear as a set of unrelated or distinct images. While this initial population is unlikely to be viewed favourably, users can quickly generate more interesting sets of images by allocating high ranks to images in the initial population that are deemed to have interesting features, such as repeating patterns, and appealing colours or shapes. In the next generation, users are likely to find some interesting images, with characteristics shared with their chosen parents.

If a user is dissatisfied with the randomly generated initial population of images, they may recreate a new initial population manually. A database of images is maintained by the system with the ranked images from different generations or system runs. A user may select favorite images from the database to be present in the initial population. The system can automatically read the formulae for these images and translate them into corresponding GP trees.

4.3.3 Parameter Setting

Different functions and terminals will result in different styles of image being generated. For example, trigonometric functions create repeating patterns, while linear functions result in smooth changes in pixel intensity. IMAGENE allows the user to select different sets of functions and terminals. The user can also experiment with different settings of GP parameters such as crossover and mutation rates and maximum permitted tree depth.

5. USER EXPERIENCES

IMAGENE has been used by approximately 100 users. The users have ranged from school children who used it for 1-2 hours to experienced artists and designers who have used it for many hours over many months. User experiences have produced a variety of images, some of which are shown in figure 6.

IMAGENE was used by a group of 15-16 year old high school students who were attending a science summer school. They had been selected for the school by academic achievement and interest in science. They received a short presentation on the basic concepts of formulae, image rendering and computational evolution. They then used the system to generate their own images. Soon some were excitedly calling to friends across the room to come and see images they liked. Some students were unlucky and didn't get any interesting images for a number of generations and were starting to lose interest. Many students wanted printouts of images they had generated to take home. Evaluation feedback after the summer school indicated that most students enjoyed this experience.

IMAGENE was used by a multi-media design student for a major project. This student spent many hours with the program with the goal of generating interesting images. She generated many stunning images, some of which are reproduced in this paper. Many of the interesting images tended to come in families, each member of a family being a recognizable variation of another member of the family. She did, however, report a sense of frustration with system for two reasons: (1) Many of the images generated were not very interesting. A considerable amount of time was spent ranking images which were not at all interesting or were very similar to images previously ranked. (2) There was not quite enough user control of the process. The only possible input was an order of preference for the displayed images, whereas she wanted to explore such possibilities as "I wonder what would happen if images 4 and 8 were parents?".

Experienced artists and designers who have used IMAGENE have been delighted with some of the images that have been generated and have been very excited about using the evolved images as a starting point for further creative work, with tools such as photoshop. Some images created in this way have won first prize at an international evolved art competition [3]. However, they have reported a sense of frustration in that considerable effort is needed to rank many uninteresting images before a stunning one was generated and were frustrated by lack of a way to have creative input into the evolutionary process.

6. EVOLVED IMAGES

We present a series of some evolved images from IMAGENE, in this section. Figures 7 and 8 show some images from an initial population generated randomly by IMAGENE. It is clear that most of them, collectively, are not appealing or interesting. However, within individuals it is possible to identify some interesting features, such as the depiction of a rainbow-shaped pattern in Figure 7b, and a background repeating pattern in Figure 8c.

Interaction with IMAGENE becomes exciting once a user finds appealing images, such as those shown in Figures 9 and 10. Beautiful images may be located in many places, but the feeling of satisfaction assumes a higher plane when a beautiful image is created by oneself. Most users find it is entertaining to drive the evolutionary process, which starts from a very random population of images, then suddenly delivers an interesting image in a subsequent generation.

The colour images are usually more attractive than grey images. We compared our colour images with the images created by the method of using a 3D-vector described in section 2.1. Although we have not conducted a rigorous comparison, our colour images tend to be visually more colourful and varied.

The evolution of families of images is another exciting outcome of IMAGENE. In Figure 11, image (c) is the child of image (a) and (b) and image (d) is the child of (b) and (c). The members of a family have some similar features yet are different. They could be deployed in various media contexts, such as animation design and game background.

Examples of non-symmetric and symmetric image generation are presented in Figures 12 and 13, respectively. While the two sets of images look quite different, corresponding images in the sequence have been produced from the same image formula. It is difficult to imagine that the images labeled (d), for example, are produced by the same formula. This ability to generate alternative images with the same formula assists the user to find diverse and interesting images quickly.

6.1 Number of potential images

It was interesting to note that IMAGENE did not disengage or irritate users by generating similar images over and over again. While some image patterns re-occurred in several runs, most runs generated different patterns. In this section we analyze the reasons for this.

As illustrated in Figure 6 the nodes in the generated trees are either unary or binary. The number of possible formulae depends on two factors: (1) the number of different unlabelled tree shapes that are possible for a given depth, and (2) the number of different ways of assigning labels from the function and terminal sets to the interior nodes and leaves.

There are no published results on the number of tree shapes for the kinds of trees that appear in IMAGENE. However there are results for binary trees. The number of different tree shapes at a given depth, d, is a "double exponential" sequence [1]. For tree depths $d = 1, 2, 3, 4, 5, 6...9$ the number of tree shapes is $1, 3, 21, 651, 4.5E5, 2.1E11...3.7E60$. For the kinds of trees that appear in IMAGENE these numbers would be even higher since binary tree shapes are subset of the possible shapes with unary and binary nodes.

For any given tree shape there is a large number different ways of assigning labels to the nodes. Thus, the number of possible formulae at a given depth is much greater than the number of shapes. For a depth of 9 the number of formulae is much greater than $3.7E60$. In IMAGENE we use a maximum depth of 12. With such gigantic search space it is not surprising that new runs give novel images.

7. CONCLUSIONS

This paper has described an interactive image evolutionary system known as IMAGENE which can generate interesting, novel, and sometimes stunning images under user supervision. As an interactive application, the user plays a key role in using IMAGENE. The user assigns the fitness to each generation of images and can choose whether the initial population of images is created automatically or manually. As can be expected from a system that generates images with operators that involve randomness, many images are not very interesting. However interesting and pleasing images usually emerge after a few generations of the evolutionary process.

We have achieved our original goal of evolving interesting images. We have also succeeded in making the process an enjoyable and entertaining one for many users. Over 100 users of different ages and backgrounds have used the system and have stated that they enjoyed using it. Factors contributing to entertainment value include: (1) The system keeps generating novel images in different runs, (2) Not all images are novel and interesting. It takes some effort and perseverance to get really good images. (3) Families of images, in which the children have similar characteristics to their parents are generated, (4) The user can change the kinds of images that are generated by experimenting with different options and parameter values.

It has been particularly pleasing that the artists and designers were inspired to use the evolved images for further creative work, either by selecting and recombining evolved images into new compositions, or adding various effects with programs like Photoshop.

While we have yet to perform a rigorous comparison, it does appear that our approach of using 3 individuals to render one image gives more novel images with richer colours and patterns than techniques which use one individual with a 3D vector representation of colour.

Some aspects of the system need further work. We would like to improve the current user interface which is somewhat inelegant and clumsy in parts. We would like to explore new terminals and functions and the different image styles which could result. We would like to explore different methods for rendering the images from formulae, in addition to the symmetric/non symmetric methods. We would like to find ways in which the artist can have more creative input into the evolutionary process. We would also like to carry out a more rigorous evaluation of the system. One intriguing direction for further work would be to use machine learning techniques to try to learn a user's aesthetic preferences and then to use these preferences in guiding the evolutionary process.

8. ACKNOWLEDGMENTS

We thank Marsha Berry, Naomi Herzog and Karen Trist, our art and design collaborators, for their contributions to this work. We also thank Xiang Li for coding the original IMAGENE grey level and colour systems, Irwan Hendra and Emil Mikulic for technical assistance and Katherine Chau for the use of her evolved images.

9. REFERENCES

[1] A. V. Aho and N. J. Sloane. Some doubly exponential sequences. *Fibonacci Quarterly*, 11:429–437, 1970.

[2] S. Baluja, D. Pomerleau, and T. Jochem. Towards automated artificial evolution for computer-generated images. *Connection Science*, 6(2–3):325–354, 1994.

[3] V. Ciesielski, M. Berry, N. Herzog, X. Li, and A. Song. Winning image. *IEEE Congress on Evolutionary Computation Evolved Art Competition*, 2005. http://www.cs.rmit.edu.au/~vc/evolved-art/gallery.

[4] A. Dorin. Aesthetic fitness and artificial evolution for the selection of imagery from the mythical infinite library. *Lecture Notes in Computer Science*, 2159:659–668, 2001.

[5] D. S. Ebert, F. K. Musgrave, D. Peachey, K. Perlin, and S. Worley. *Texturing and Modelling: A procedural approach*. Morgan Kaufmann, 2003.

[6] J. Graf and W. Banzhaf. Interactive evolution of images. In *Evolutionary Programming*, pages 53–65, 1995.

[7] D. Hart. Toward greater artistic control for interactive evolution of images and animation. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Sketches*, page 2, 2006.

[8] J. R. Koza. *Genetic programming: On the programming of computers by means of natural selection*. Cambridge, Mass. MIT Press, London, England, 1992.

[9] P. Machado and A. Cardoso. All the truth about NEvAr. *Applied Intelligence*, 16(2):101–118, 2002.

[10] S. Rooke. Eons of genetically evolved algorithmic images. pages 339–365, 2002.

[11] B. J. Ross, W. Ralph, and H. Zong. Evolutionary image synthesis using a model of aesthetics. In G. G. Yen, L. Wang, P. Bonissone, and S. M. Lucas, editors, *Proceedings of the 2006 IEEE Congress on Evolutionary Computation*, pages 3832–3839, 2006.

[12] B. J. Ross and H. Zhu. Procedural texture evolution using multiobjective optimization. *New Generation Computing*, 22(3):271–293, 2004.

[13] K. Sims. Interactive evolution of equations for procedural model. volume 9, pages 466–476, 1993.

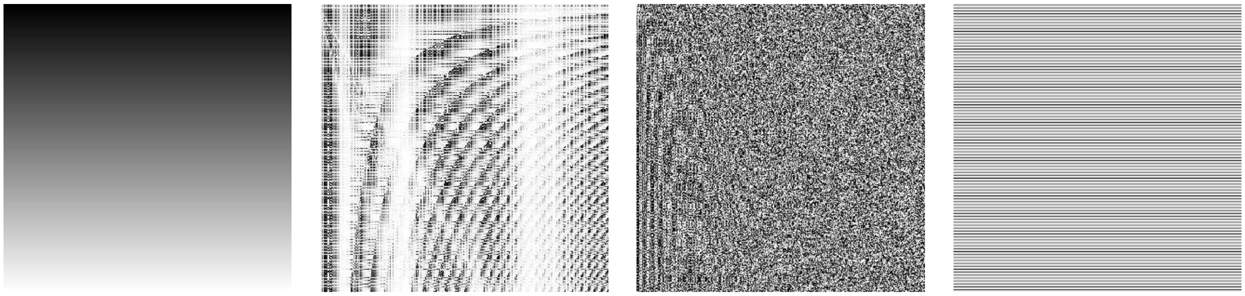
[14] E. J. Stollnitz, T. D. DeRose, and D. H. Salesin. *Wavelets for Computer Graphics: Theory and Applications*. Morgan Kaufmann, San Francisco, 1996.

[15] N. Svangaard and P. Nordin. Automated aesthetic selection of evolutionary art by distance based classification of genomes and phenomes using the universal similarity metric. In *Applications of Evolutionary Computing*, volume 3005, pages 447–456, 2004.

[16] L.-Y. Wei and M. Levoy. Fast texture synthesis using tree-structured vector quantization. In *SIGGRAPH '00: Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 476–488, 2000.

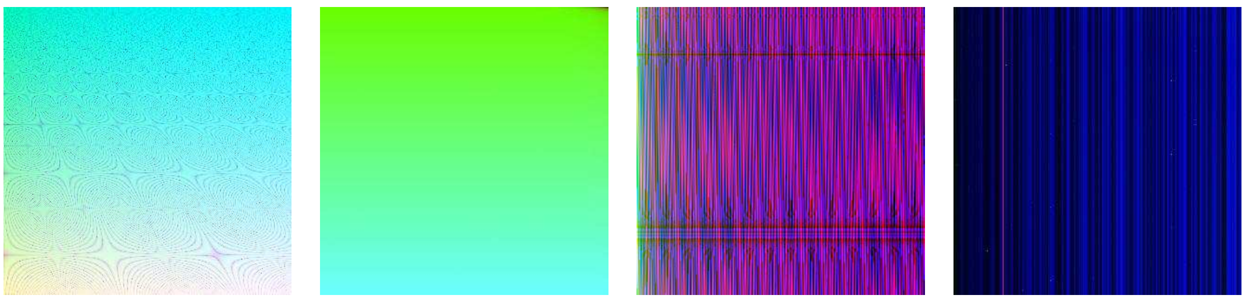
[17] M. Whitelaw. Breeding aesthetic objects: art and artificial evolution. pages 129–145, 2002.

[18] A. Wiens and B. Ross. Gentropy: Evolutionary 2d texture generation. *Computers and Graphics Journal*, 26(1):75–88, 2002.



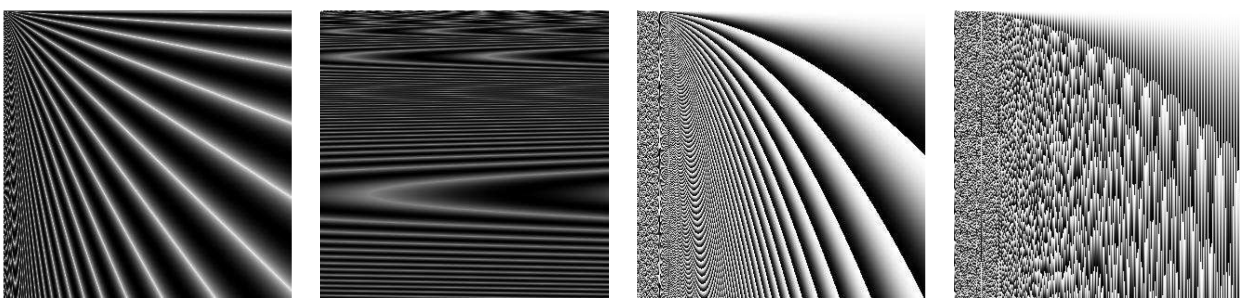
(a) (b) (c) (d)

Figure 7: Grey scale images in an initial generation



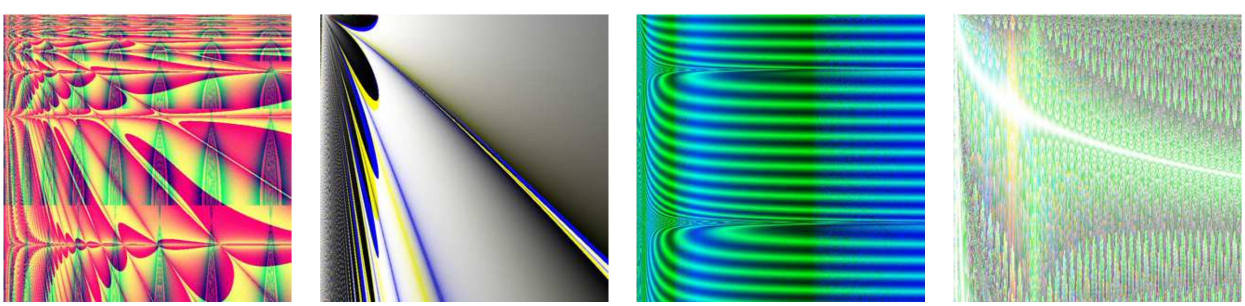
(a) (b) (c) (d)

Figure 8: Colour images in an initial generation



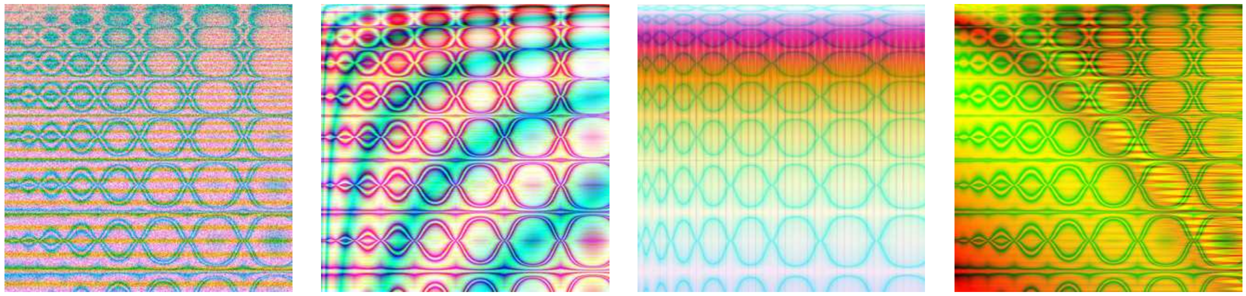
(a) (b) (c) (d)

Figure 9: Some evolved interesting grey level images



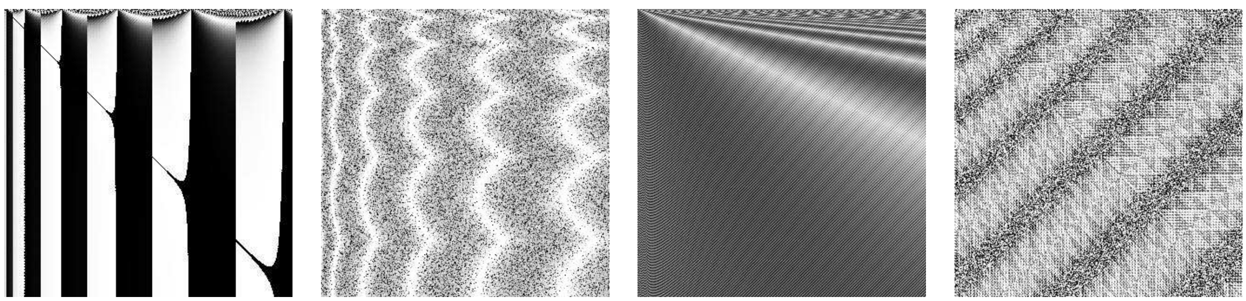
(a) (b) (c) (d)

Figure 10: Some evolved interesting colour images



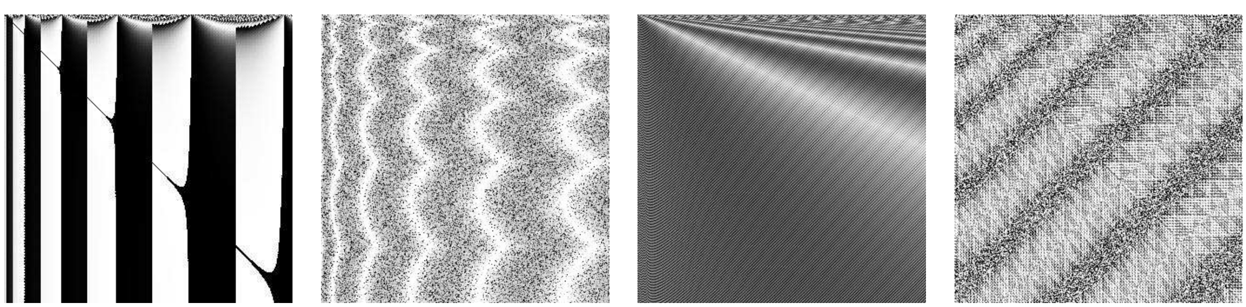
(a) (b) (c) (d)

Figure 11: An image family



(a) (b) (c) (d)

Figure 12: Examples of non-symmetric renderings



(a) (b) (c) (d)

Figure 13: Examples of symmetric renderings