COSC 5P75: Directed Reading - EvoART

Michael Gircys

September 4, 2015

Contents

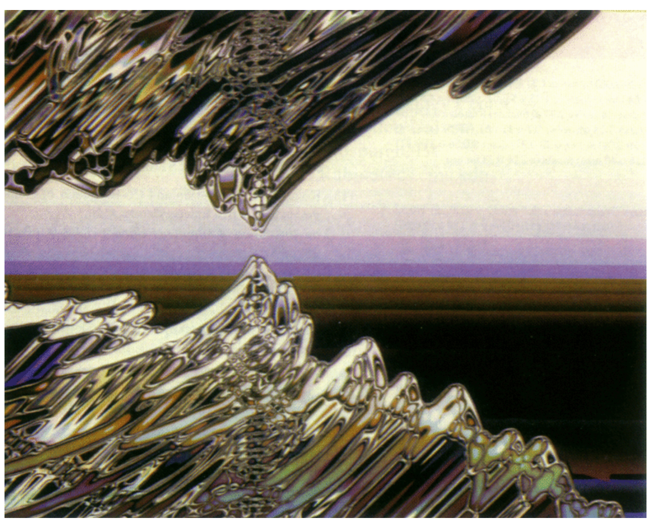
List of Figures

1 Introduction

The purpose of this report is provide a non-exhaustive review of publications relevant to the synthesis of textures from evolutionary methods. Specifically, works which present fundamental or novel approaches towards the advancement of artificially evolved artistic and aesthetic images are considered, and are surveyed and briefly summarized to provide a snapshot of the field. We examine previously considered representations, languages, and fitness metrics which have shown promising results in previous experimentation.

The concepts of natural and artificial selection are fundamental to the vision of our current world. Such a simple but powerful phenomenon is directly or indirectly responsible for every aspect of our modern life. Individuals who perform and adapt well to their environment are better able to produce scions with similar, superior traits. This concept, if generalized, provides a powerful tool for searching through candidate solutions for the superior solution to a structured problem. From Dawkins' paper on evolutionary design with the Blind Watchmaker [54], to the first definitive adaptation to art with Karl Sims [3], and beyond, the evolutionary process has been tested with success in producing superior solutions to less-traditional technical problems.

Figure 1: Karl Sims - Untitled [3]



Aesthetic evolution challenges candidate solutions of the system - artistic pieces - to outperform other generated solutions in measures of various aesthetic attributes. In an interactive system, the beauty, interestingness, or other measures of appeal of each piece are evaluated by a user, permitting a subjective measure to guide the evolutionary process. Various ways to automate this process - and to permit aesthetic evolution - without strong user guidance have been attempted with varying degrees of success through learning systems and more defined, rank-able measures of aesthetic appeal.

Many questions regarding the artistic meaning of artificially evolved images remain, questioning the extent to which modern definitions of art can apply to the various processes explored in this paper [10]. Nevertheless, there is little doubt that one can find many aspects of creativity and gain inspiration for future works from the results of evolutionary processes, natural or otherwise. This paper will attempt to briefly outline some of the existing work in the field that was deemed to be notable by the author.

We first review the concepts of genetic algorithms, genetic programming, and procedural textures in Section 2 of this report. In Section 3, possible genetic algorithm representations, languages, and operators are considered, with some distinction between those used for procedural textures, vector graphics, or alternate approaches. We discuss possible schemes for fitness evaluation in Section 4, beginning with interaction-dependant evaluations and fully automated approaches, to learning system hybrids and other methods. In Section 5, a subset of other noteworthy systems - which could not strictly fit into the report framework - are reviewed, before offering a conclusion in Section 6.

2 Review

The natural process of evolution has lead to generations of individuals who are better able to survive and adapt to the situations which they have encountered. By borrowing this metaphor, and repeatedly selecting fit individuals for candidate solutions, genetic algorithms are able to gradually refine the generated solutions.

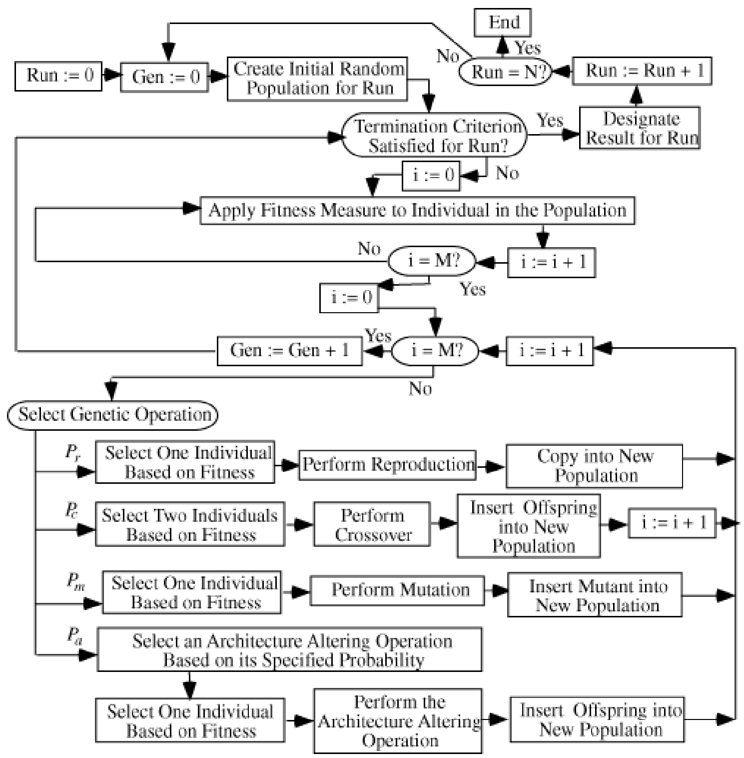
Genetic algorithms (GA) provide a way to investigate a search-space with some unknown structure. When employed correctly, genetic algorithms have proven to be capable of solving optimization, classification, regression, a-life, and other search problems. The specific interest researched for this paper pertains to applications in evolutionary design and art.

2.1 Genetic Algorithms

Applying the process of natural selection as found in nature to a group of possible solutions in a search space requires a number of considerations. A termination condition and fitness function must first be determined. As we wish to give the genetics of more ideal individuals preference for progressing to the next generation, we must determine a function which can be used to compare two individuals on a measure of "idealness". It is also necessary to know when we should stop searching for solutions. An exactly perfect solution may be achievable, and we should not continue to search for other solutions if we should find it. However, a perfect solution may not be possible, and better results might be obtained from performing multiple runs with fewer generations.

It is critical that the representation, and the interpretation of this representation, be sufficient for the problem we are trying to solve. The representation

Figure 2: Overview - Genetic Algorithms [24]



must be sufficient to encode a valid solution to the problem, and the representation must also permit the solution to be verified without error. This is particularly applicable to the GP algorithms below.

2.1.1 Fitness

Fitness functions are used to determine the correctness of a solution, and must be chosen carefully. It may often be required to test a solution across a large number of data points or scenarios to determine overall suitability of the solution. While scoring the solution across all data sets may provide more accurate evaluation, performance is sacrificed. It may be more effective to attempt multiple, quicker runs with less accurate fitness evaluation. Additionally, some data points should be held from use in scoring until the end of the run, at which time a final assessment is performed using these reserved data points. It is possible that solutions may find local optima for the set of evaluation points. By providing additional test cases after possible solutions have converged, the flexibility and adaptability of the solution can be evaluated.

For consideration of multiple distinct fitness measures, a number of multi-objective fitness functions can be considered, each with their own advantages and disadvantages. Note-worthy multi-objective techniques include Pareto ranking, fixed weight sum, and rank sum techniques [52].

2.1.2 Fitness-Proportional Selection of Parents

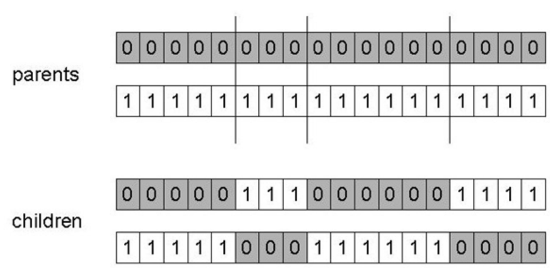
While many strategies exists for selecting parent individuals for a reproduction operation, there are perhaps two particularly common methods. The general premise requires selecting the most fit candidate amongst a population or sub-population, and individuals with better fitness should have a greater likelihood of passing genetic information to the next generation of candidates [2].

Tournament selection takes a configurable number of random candidates from among the population and selects the fittest of the lot. If a candidate dominates all others by a large margin, it will show up in the next generation just as much as if it had dominated all candidates by a small margin. Roulette Wheel selection uses each candidate's fitness to proportionally assign a likelihood that the candidate will get selected. Using this selection method, fitnesses can be scaled to adjust the selection pressure among candidates.

2.1.3 Crossover

In the initial, canonical versions of genetic algorithms by Goldberg [2], the number of crossover points is configured beforehand. Crossovers are typically limited to 1 or 2 points, though many-point crossovers are possible [1]. The position of these crossover points (as they align to an index in the representation) can be fixed, or randomly determined for each pair of parents. Genetic data is partitioned into regions based on the selected crossover points, and child solutions are created by copying and recombining alternating partitions of the parent solutions as seen in Figure 3. Each gene is filled by one of the parent genes of the same position. Two valid children should be possible by recombination of the pair of parent nodes.

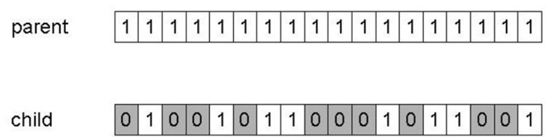
Figure 3: Genetic Algorithms - N Point Crossover [8]



2.1.4 Mutation

Mutation can be a useful operation to introduce and maintain genetic diversity. By selecting regions of a parent solution, and replacing it with modified or randomly generated information, it is possible to help prevent convergence on a local optimum. Figure 4 illustrates a high amount of mutation introduced within a solution. An alternate common method of GA mutation is to select a pair of equally sized regions within the same parent solution and swap genetic content.

Figure 4: Genetic Algorithms - Mutation [8]



2.2 Genetic Programming

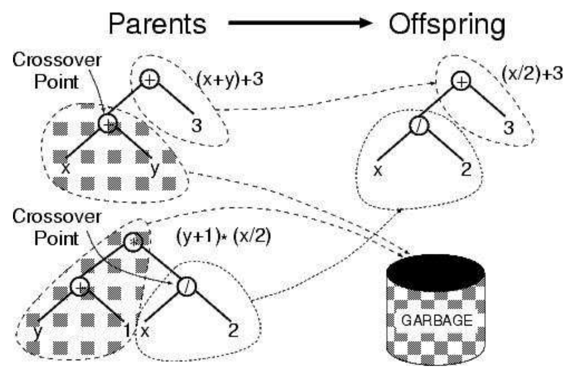
Genetic programming (GP) replaces the flat data representation of canonical GA for a variable-length, tree-based representation [57]. Each node represents a function, where the children of a node are recursively solved and the values are provided as its parameters. As the root node of a solution should represent a valid functional-paradigm program, solutions can be encoded as LISP S-expressions.

GP builds on canonical GA and in much the same way requires a suitable termination condition and fitness function. Additionally, there are a number of new considerations that a programmatic, node-based representation requires. The GP language - the possible terminal and non-terminal nodes - must be determined. The language must be sufficient to represent valid solutions to the problem. Omitting a function from the possible allowed nodes may make arriving at a correct solution much more difficult, or entirely impossible. A concern more serious with GP over GA is the assurance that a candidate will execute without error. A given node recursively evaluates its child nodes to determine the values of its parameters, but it may not be assured that the range of the child node is the same as the domain of the expected parameter. Divisions by zero, and square roots of negative numbers must have defined behaviour for the problem. This will likely mean creating safe versions of the offending functions to ensure non-error behaviour across the entire domain.

2.2.1 Crossover

Crossover is a strength of GP, and is comparatively simple in contrast to GA. When two parent solutions have been chosen, a random non-root node is selected from each parent. These nodes are swapped across the two solutions, and the results saved for the next generation. If there are failed constraints for valid children on the picked nodes, the node selection process can either retry, or pick the closest valid alternative. Figure 5 shows crossover for GP, though two children can be created from the process, and discarding one is not generally necessary.

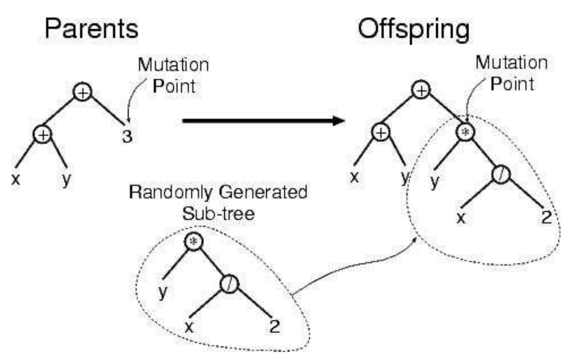
Figure 5: Genetic Programming - Crossover [27]



2.2.2 Mutation

Mutation of a tree is accomplished by selecting a random node within the tree and replacing it with a randomly generated alternate sub-tree, as seen in Figure 6. One technique occasionally employed is to generate a new sub-tree with depth or weight similar to the sub-tree which was replaced.

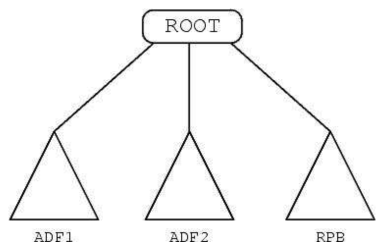
Figure 6: Genetic Programming - Mutation [27]



2.2.3 Automatically Defined Functions

Structure altering operations can be used to employ Automatically Defined Functions (ADFs) and other automatically-defined features such as lists, iterations, and recursions [56]. Abstractions for more complex functionality can be created by developing labelled sub-trees as children to an individuals root node (as in Figure 7), which can then by referenced by the main evaluated tree branch. Operations can be used in reproduction to create ADFs from a sub-tree of the main branch of an individual, or to remove the ADF by replacing it with it's definition in-line. The ADFs can evolve through mutation and crossover operations alongside the main branch of the tree.

Figure 7: Genetic Programming - ADFs [27]



2.2.4 Grammar-Guided GP

When the structure of more ideal solutions is known, or when candidate solutions require specific structure for validity, a grammar can be used to guide the creation and modification of the representation tree [34]. When selecting subtrees of candidates during crossover, care is taken to ensure that nodes are valid after the operation completes. During crossover, the generated replacement tree is built using grammar production rules to ensure compatibility. Internally, grammars can be used to ensure that the output of each evaluated node constitutes valid input for the parent node by employing strong typing. Figure 8 displays a potential grammar for generation of syntactically correct formulae strings, perhaps to optimize formulae size.

Figure 8: Genetic Programming - Grammars [27]

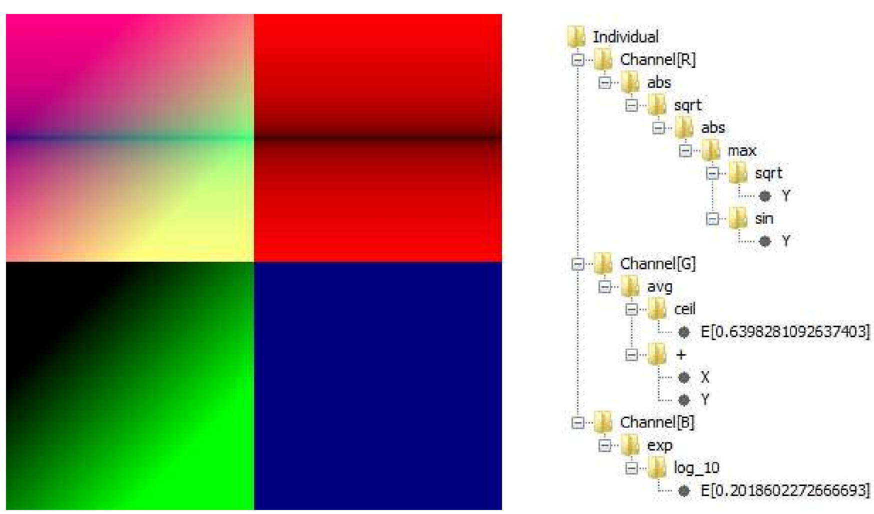
$$\begin{aligned}
\mathit{tree} &::= E \times \sin(E \times t) \qquad (6.3) \\
E &::= \texttt{var} \;\mid\; (E \;\texttt{op}\; E) \\
\texttt{op} &::= + \;\mid\; - \;\mid\; \times \;\mid\; \div \\
\texttt{var} &::= x \;\mid\; y \;\mid\; z
\end{aligned}$$

2.3 Procedural Textures

A procedural texture uses a formula or algorithm to determine colour value for a given position. The colour for a position in 2D, 3D, or other spaces can be independently determined and expressed by a common formula with respect to the coordinate values for that position. A strength of this approach is the abstraction of specific details, where instead of specifying every individual aspect of a rendered scene, image attributes emerge from a more concise formula [55]. The approach is also desired for its offering of parametric control. Formulae can be created which assign meaningful measures to user-controllable variables, permitting re-use and quick tuning of the function to a user's preference.

Sets of simple 2D texture functions can be created by using X and Y coordinate variables, and a minimal set of arithmetic operators. In Figure 11, the top row of monochrome images, from left to right, is generated from expressions `X`, `Y`, and `(abs X)` respectively. Operators can be nested to induce greater image complexity, where periodic functions tend to provide image repetition, and conditional operations can create contrasting intensity changes. Figure 9 displays how each channel in RGB space can be separately evolved and combined to create an interesting image. In rendering and evaluating image solutions, some care may need to be taken to provide sufficient and unbiased data type conversion. A common approach is to normalize or clamp position data and the final result values to a discrete range. A consideration of the behaviour shown from the included functions may be prudent to lessen the bias of the function input domain with respect to the render window. Rendering within [0,1] ranges for X and Y may not display all the same variation that can be seen within a [-1,1] range (ie. a centred spiral may only display a certain quadrant - appearing to be set of curves).

Figure 9: IEAS Example - Separate Channels and Complete Image Rendering

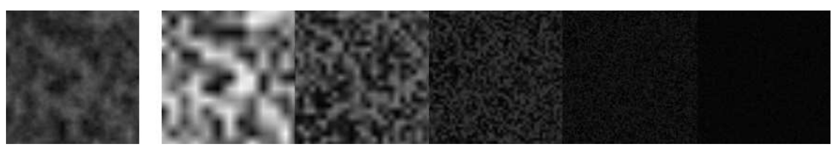


2.3.1 Noise Generators

Noise generators are an exceedingly important tool for the creation of procedural textures. Not all desired texture traits originate from ordered, fully deterministic methods; it can be desirable to display attributes with a feel of randomness. Noise functions introduce a stochastic element to the rendering function by precomputing and making available a mapping of (pseudo-)random values across each pixel.

A truly random noise implementation is often too chatoic; the contrast between adjacent values is often too great, giving a "grainy" feel to the image. This random noise suffers from a lack of structure, but by using a smaller sampling of random points to create a lattice, and using creative ways of interpolating between these points, a more ordered result can be obtained. This provides a source of gradient texture data which is highly stochastic, but yet maintains a flowing, structured appearance - gradient noise.

Figure 10: Fractal Noise Generation [18]



Fractal noise is arguably the most important element of modern procedural texture generation [55]. Building on a gradient noise, such as from the renowned Simplex or Perlin noises generators [12], we can combine renderings of the noise at various resolutions or *harmonics*. By summing these harmonics with weight proportional to their harmonic index (see Figure 10), we can generate a fractal noise that has both a soft flow and high amount of finer detail [18]. These noise functions can be further tuned and composed together through arithmetic operations to alter the characteristics of noise, such as those seen in Figure 13.

3 Representation & Languages

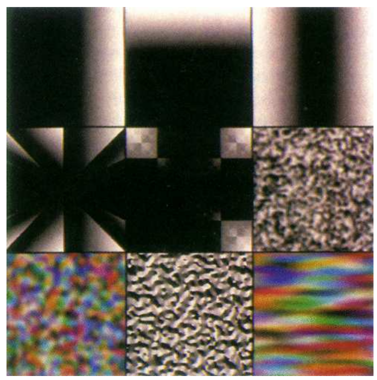
Finding a suitable representation for individuals is essential for GA, and the requirements for evolutionary art must be considered. It can often be seen that each language and representation returns a distinctly associable set of results, where the language is often said to be one of the greatest factors in determining the style of the resultant images.

Using a bitmap as an individual representation is often too fine and granular. If possible, we may wish to evolve a higher-level context with which we use to render our canvas. Commonly, canonical GA or GP is used as an intermediate, genotype representation. These individuals could then be converted into a phenotype of a bitmap raster image.

3.1 Symbolic Expressions

A procedural texture representation is a mathematical, symbolic expression which returns colour data. Typically, data for a single pixel is returned from evaluating a GP individual, which is iterated over each coordinate of the raster image. Such a GP representation may have three trees evolved on each individual, corresponding to a tree for each channel in the colour space. Variation arises from the change of location data which is provided as part of the GP language.

Figure 11: Karl Sims - Small Programs [3]



The founding work in generalized evolutionary art explored by Karl Sims in 1991 [3] uses a genetic programming representation to symbolically encode and evolve a 2D pixel colour evaluating function. The function set for the work contained standard mathematical operations, vector transforms, noise generators, and a number of image and colour processing operations. Operators could return either scalar or vector values. Many of the functions were implemented with a "warp" variant, which specify offsets to the coordinate values if used internally within the function (Figure 12). While comparatively minimal in comparison to some modern works, the language was impressively expressive with even minimal programs (Figure 11)). Expanding on his results, Sims was able to experiment with 3D textures through the addition of a Z terminal value to provide location data for all of (X,Y,Z) space. Similarly, animations were toyed with by evaluating subsequent frames within a (X,Y,Time) system.

In the 2000 article for Gentropy by Wiens and Ross [9], a similar approach can be found. While the focus of experimentation lies in the assessment of automated fitness, the variation in GP language can be noted. In comparison to Sims' language, one can find additional variation in the texture functions provided. Where Sims provided a single noise function with warp and colour variants, Wiens and Ross offer multiple noise functions each with distinct characteristics (Figure 13).

Warp and tiling functions are still present, permitting the specification of

Figure 12: Karl Sims - Warped Noise [3]

```
(sin (+ (- (grad-direction (blur (if (hsv-to-rgb (warped-
      color-noise #(0.57 0.73 0.92) (/ 1.85 (warped-color-
      noise x y 0.02 3.08)) 0.11 2.4)) #(0.54 0.73 0.59) #(1.06
      0.82 0.06)) 3.1) 1.46 5.9) (hsv-to-rgb (warped-color-
      noise y (/ 4.5 (warped-color-noise y (/ x y) 2.4 2.4))
      0.02 2.4))) x))
```

Figure 13: Gentropy - Noise Samples [9]

Noise	Turb	Turb Flow	Cloud	Marble

Figure 14: Gentropy - Textures [9]

IMAGE	TEXTURE FORMULA
	`rgb(Y*X, noise(Y, -0.71), min(Y, 0.25))`
	`rgb(mod(turbflow(Y,X,X), sin(X)), lum(marble(0.94, -0.78, (-0.46,0.50,-0.63))), turb(chn(COLGRAD), cos(-0.24)))`
	`rgb((wchn(Y*X,COLGRAD)+Y*X)/turbflow(noise(noise(Y, X-Y), cloud(turbflow(X-Y, mod(X,Y), Y/(X-Y)), Y/X, diff(sin(X), cos(noise(Y,Y))), noise(Y,Y))), lum((-0.10,0.34,0.98)), Y/wchn(X,COLGRAD)), noise(noise(Y,Y), cloud(diff(sin(X), cos(noise(Y,Y))), diff(0.94,0.76), mod(X,Y), noise(cos((X-Y)/X), cloud(Y/X, Y/X, sin(X), noise(Y,Y))))), cos(Y*X))`

offsets or absolute values for x and y coordinates to child functions. Tiling operators can be used to create repeating rectangular, circular, or kaleidoscopic patterns. Wiens also included primitive iterative control functions. The function *forv* was used to iterate its subtree across the current colour channel, with the *chn* and *wchan* terminals available to provide the current channel index for calculation.

Hewgill and Ross [16] expand on the works of Sims in their experimentation with 3D procedural texture synthesis. An existing 3D mesh is also accepted as a parameter of the problem, which can be used to derive additional parameters about each sampled point in 3D space. It was suggested, for example, that it may be desirable to have textures with variance closer to discontinuities or surface creases. To assist with this, terminal parameter nodes were included which provided measures for surface normals, interpolated (Phong-style) mesh normals, and surface gradient measures. Figure 15 displays initial results obtained with basic functions, inclusion of normal parameters, and further inclusion of gradient parameters.

Figure 15: Procedural 3D Textures - Language Evolution [16]

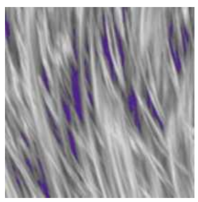
ArtiE-Fract is a system created by Lutton [22] for artists and designers based on iterated functions within a GA system. Lutton's program did not add iterated function systems (IFS) to the GP language. Rather, the image generated through GP in the program is used as an IFS attractor image, which is passed in to a configurable IFS. Three models of IFS (affine, mixed, and polar) can be combined and employed to display shapes which show more or less fractal symmetry. An IFS morphing can be seen in Figure 16.

Figure 16: ArtiE-Fract IFS Morphing [22]

In returning to a more direct GP language, Reynolds provides a more varyingly typed texture synthesis toolbox. The output range of the language functions includes real numbers, 2d position vectors, 3d colour vectors, and other textures [39]. The functions provided by the systems can be classified between texture generators, which provide base texture objects, and texture operators, which can apply additional transforms to the created textures. Accepting no other texture as input, the texture generators include uniform colourings, gradations, waveform gratings, and noises. Texture operators may include simple scale, translations, stretch, and rotation operators, and also twists, blurs, edge detections, colourize, hue/saturation adjustments and slices, and thresholds. While a complete and up-to-date listing could not be obtained, ongoing lab journals from the author show a large and developed tool-kit of operators at the disposal of the system.

Evolution of individuals is primarily through traditional crossover, and minor alterations to ephemeral constants. The created system has also held up well with filter creation, as in the authors works with alternative camouflage synthesis [40].

Figure 17: Reynolds - Gray with an Accent Colour - Example [39]



```
Max (Max (UniformColor (Pixel (0.308089, 0.127216, 0.564523)),
        VortexSpot (1.23485, 5.30871, Vec2 (1.99217, 0.137068),
                Furbulence (0.152681, Vec2 (-1.74168, 0.119476)))),
    VortexSpot (1.23485, 5.30871, Vec2 (2.91655, 0.119476),
            VortexSpot (1.23485, 5.30871, Vec2 (1.99217, 0.138486),
                    Max (UniformColor (Pixel (0.308089, 0.127216, 0.564523)),
                        Furbulence (0.35606, Vec2 (2.91655, 0.119476))))))
```

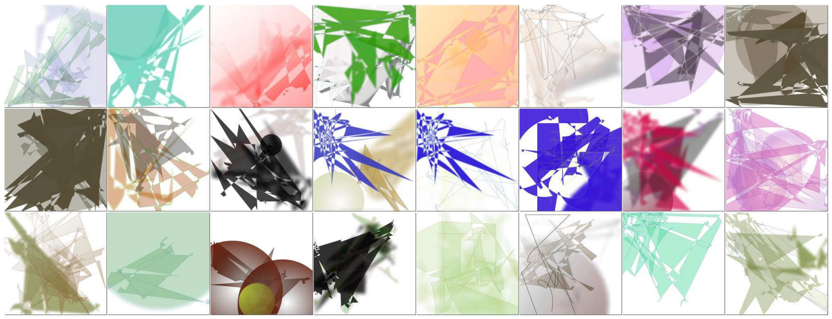
3.2 Vector Graphics

Vector graphics differ from procedural textures in that they are composed of higher level geometric primitives, instead of relying on shapes to emerge from lower level signals. While a procedural texture may need carefully ordered conditional nodes to precisely display all edges of a polygon, a vector graphic may see this appear from a single mutation. Vector graphics primitives can include polygons, ellipses, curves, images, gradients, and other items. As the representation for vector graphics permits the ease of displaying primitive predefined shapes, it is easy to consider renderings using vector-based brush strokes.

Heijer and Eiben analyse the feasibility of scalable vector graphics (SVG) for use as an evolutionary art representation in their 2011 paper [37]. While successful, a number of notes and concerns had been raised. SVG is a standard representation for vector graphics, and brings with it a number of advantages for interoperability between designer and evolutionary art system. However, the existing SVG standard is XML based, and has strict schema which cannot be violated - type safety is not as easily assured as with symbolic representation. The solution that Heijer employed required a customized mutation and crossover operator for each primitive to ensure that all required - and only valid - attributes were inherited for the offspring individual. Crossover was for simplicity left as a 1-point crossover, merging styling information from one parent, with shape information from the other. With the system encoding to XML (or some intermediate format), the representation can take a variable length, increasing or decreasing the number of shapes through reproductive operations.

Izadi considers using a vector graphics representation for evolving a filter using triangular primitives as brush-strokes [38]. Using a grammar-guided GP representation (Figure 19), groups of brush-strokes are chained together, with triangular brush-stroke primitives being generated by functions accepting ephemeral

Figure 18: Heijer & Eiben - SVG Renderings [37]



values for position and orientation data. Each brush stroke function accepts values for position, size & orientation, and colour. The final image is compiled as each node in the tree is evaluated - using a side effect of rendering on the top of a canvas. To handle the case of a brush stroke hiding a previously rendered stroke, a number of stroke placement strategies are evaluated. The pixel colour of overlapping strokes can be blended, the new stroke can be applied to only untouched pixels, or in the case of collision, the most recent stroke can be ignored.

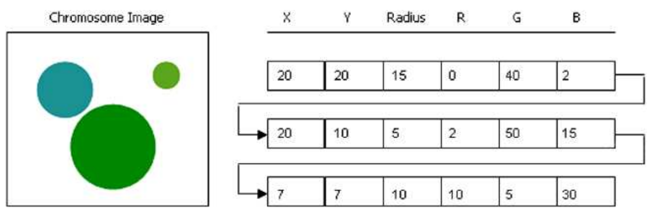
Figure 19: Izadi - Vector Graphic Grammar [38]

```
NODE = PROGRAM3 | PROGRAM4 | TRIANGLESTROKE
PROGRAM3 = Program3(NODE,NODE,NODE)
PROGRAM4 = Program4(NODE,NODE,NODE,NODE)
TRIANGLESTROKE = Triangle(T,T,T,T,T,T,T,T)
T = floating point number
```

Bergen and Ross expand on the idea of evolving filters using vector brush strokes, but returns to a fixed representation in the JNetic system [42]. The JNetic system aspired to create non-photo-realistic textures from a source image by repeated application of coloured primitives on a canvas. Options for primitives included circle, rectangle, line, N-polygon, paint-stroke, or grid squares. While the use of multiple distinct shape primitives is supported, a single shape primitive is optimal for the fixed-length representation. Crossover can be greatly simplified, as information at a given index will control similar properties across all individuals. A defined property of the primitive, such as X and Y coordinates, colour channels, and shape data, can be transferred completely to the offspring candidate - partial and possibly invalid or non-meaningful data may not arise in child individuals.

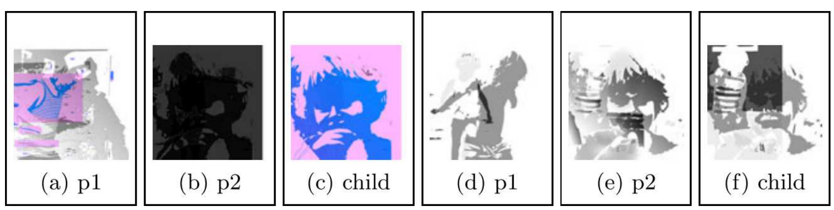
After their previous success in evolving abstract SVG images, Heijer and Eiben [43] consider strategies for evolving photo-realistic, vectorized images. A

Figure 20: JNetic - Chromosome Representation [42]



number of new mutation operations have been introduced which are separated into macro-level and micro-level mutation groups. Macro-level mutation effects the entire individual, where micro-level is isolated to a subsection of an individual (based on scope tags included in the language). Macro scale mutation includes sampling other images into the individual, or adjusting the style-sheet which controls all rendering properties for the individual. Micro scale mutation includes creating a mirror image of part of the individual, replacing curves with lines, or adjust scalar values by up to 5 percent. Crossover operations are also more context-aware. During a crossover, the background is taken from one parent, the style-sheet is taken from the other parent, and one of the shape groups is taken from a parent individual. Examples of parent property selection can be seen in Figure 21.

Figure 21: Evolving Pop Art - Property Inheritance [43]



3.3 Hybrid & Other Systems

Plants have long been used for aesthetic modelling with computational aesthetics, and Lindenmayer's L-Systems have proven to be a greatly useful and artistic grammar system. The evolution of L-Systems had previously been proposed by Koza in the first publication of *Genetic Programming*. While deterministic L-Systems present a simple context-free grammar that can be evolved, context-sensitive IL-Systems can provide additional challenges. Jacob [5] provides an abstracted method for the coding of L-Systems. Figure 22 shows an example coding for deterministic L-Systems, but can be easily appended with context-sensitive rules.

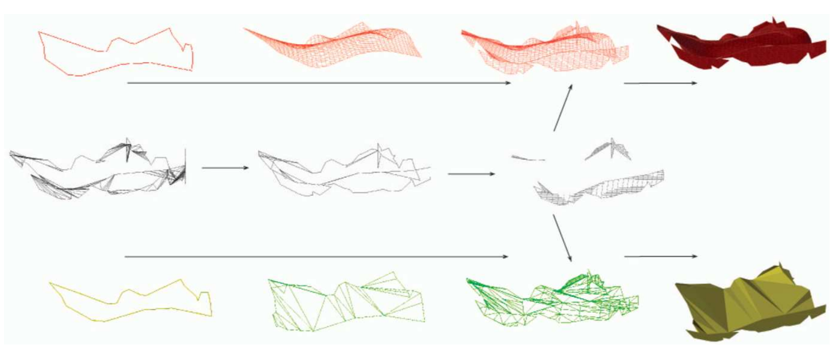
L-Systems were further explored in a 3D context by Hemberg *et al.* [21],

Figure 22: L-System Encoding [5]

```
LSystem[_AXIOM,_LRULES],                                          (1)

AXIOM[sprout[4]],                                                 (1)

LRULES[
  LRule[LEFT[], PRED[ sprout[4] ], RIGHT[],
        SUCC[
          SEQ[
            SEQ[f],SEQ[stalk[2]], STACK[PD[60],leaf[0]], ...,
            __SEQ
            SEQ[f],SEQ[stalk[1]], bloom[0]]]],
  LRule[LEFT[], PRED[ sprout["t_/;t<4"] ],
        RIGHT[], SUCC[sprout[t+1]]],
  ...
  LRule[LEFT[], PRED[ bloom[6] ], RIGHT[], SUCC[ bloom[1]]],
  ___LRule
],                                                                (1)

LRule[LEFT[],_PRED,RIGHT[],_SUCC],                                (1)

PRED[sprout[aIndex]],                                             (1)

SUCC[_SEQ | _STACK],                                              (1)

SEQ[BlankSequence[_sprout | _stalk | _leaf | _bloom | _f |
                  _YL | _YR | _PU | _PD | _RL | _RR | _SEQ]],      (1)
SEQ[BlankSequence[_sprout | _stalk | _leaf | _bloom | _f |
                  _YL | _YR | _PU | _PD | _RL | _RR ]],            (4)
...

leaf[leafIndex], bloom[bloomIndex]                          (2),  (2)
```

and Bergen and Ross [44]. Hemberg [21] made use of grammatical evolution systems to explore aesthetics in various architectural applications. The developed system, Genr8, uses an expanded version of the *Map L-systems* algorithm to convert the basic L-system topologies into faced surfaces. The conversion process from outline to topology can present multiple possible interpretations, as seen in Figure 23.

Figure 23: Genr8 - Outline Interpretation [21]



Bergen employs the coding scheme outlined by Jacob in an evaluation of 3D

L-System aesthetics [44]. To permit meaningful assessment of 3D geometry, the L-system is first used to fill a set size of voxel space, permitting quick assessment of simple measures, where a final rendering can be smoothened.

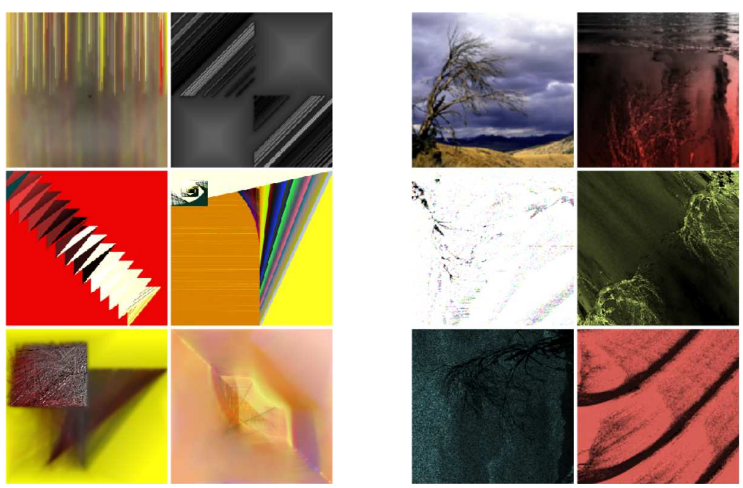
Machado *et al.* present an alternative system for evolution of context-free grammars [33]. Inspired by the concept of shape grammars, each individual genotype is a representation of a Context-Free Design Grammar. These design grammars can be constructed from, and used to construct, a graph representation. Crossover can then be performed through the exchange of sub-graphs, with care taken to preserve and restore the outgoing and incoming edges.

The Painterly Fool system developed by Colton *et al.* retains a flexible representation within its system [25]. Solutions within the Painterly Fool system are represented with GP, yet the precise structure of these trees are user-defined through a representation file, which assigns type data to various terminal and non-terminal nodes. Further, constraints on the tree structure may be specified within a user-defined constraints file, where operations that betray these constraints are reverted and retried. A compiler file may also be specified for conversion from traditional S-expressions into C-like syntax.

The Painterly Fool system may use these user-specified representations in various internal problem specifications, including simple symbolic texture synthesis, symbolic filter synthesis, or in a particle-based system. The particle-based artwork method used in the Painterly Fool uses multiple trees to specify position and colour of many particles, permitting time as a variable, and plotting over incremented time-steps (as with Sims). Each of these different methods had examples compiled to a common Processing [14] language representation.

Figure 24 shows evolved images from swarm problems (left), and source images beside their evolved textures (right).

Figure 24: Painterly Fool - Swarm and Filter Problems [25]



4 Evaluation & Fitness

We consider the first category of evolutionary schemes to be interactive evolution, which is dependant on a user to manually asses the fitness of each evolved candidate solution. The second category of evolutionary schemes is automated fitness. While a user may be required to initially make some adjustments to the fitness function, this scheme would have each evolved item automatically assigned a score, and would permit continued evolution of the solutions without interruption. There are many ways in which using a combination of the schemes can keep the user engaged while providing new and interesting results which are filtered to the users preferences. Learning systems may be used to refine and delegate classification decisions to an automatic virtual agent.

4.1 Interactive Evolution

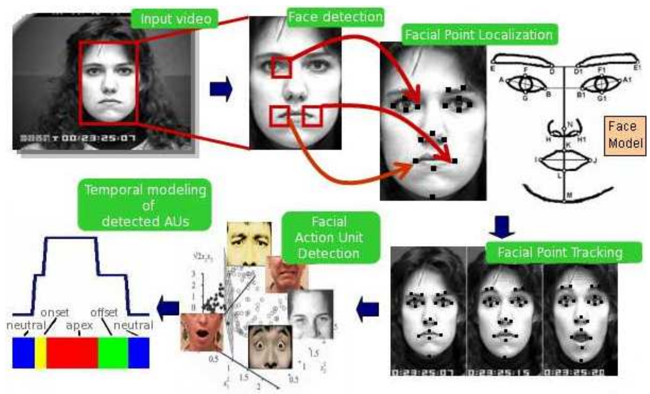
From the original work of Sims, to many of the newer and experimental evolutionary system, user interactivity has been a frequent requirement for the evaluation of aesthetic values. It is difficult to find exact definitions for aesthetic criteria, not aided by the possible variance of cultural aesthetic principles. While some common patterns may occasionally be presented with greater frequency, definitions and criteria for beauty can vary greatly among individuals even within largely monotonic cultures. By permitting a user to specify the appeal of each individual - evaluation from artificial aesthetic selection, - results become tailored to the preferences of the user without requiring any formal definitions of aesthetic values.

One interesting experiment in interactive evolution by Colton requires little explicit interaction from the perspective of the user [26]. While aesthetic theories can offer some insights into common attributes of "emotional" pieces of art, precise evaluation is an ongoing challenge. Colton uses source video of users displaying facial expressions associated with an emotion to drive evolution of an NPR filter for facial portraits. By evaluating extracted features of facial expressions known to correlate strongly with certain emotions - with assessment in facial morphology across time, as in Figure 25, - a fair estimation of the emotional impact of each source video frame can be assessed and used in scoring. Emotional scores across 6 classified base emotions were used to adjust GP parameters to drive the styles of filtered portraits, with the styling of each stroke adjusted to emotion expressed at times in the source video.

4.1.1 Combating User Fatigue

One of the key downfalls of interactive evolutionary systems is the degradation of interest from the user - user fatigue [19][35][51]. Repeated evaluation of images by the user can become mentally tiring, and generally necessitates systems relying on interactive aesthetic evolution to use a reduced amount of individuals per generation, with fewer generations.

Figure 25: Colton - Assessment of User for Emotional Cues 25

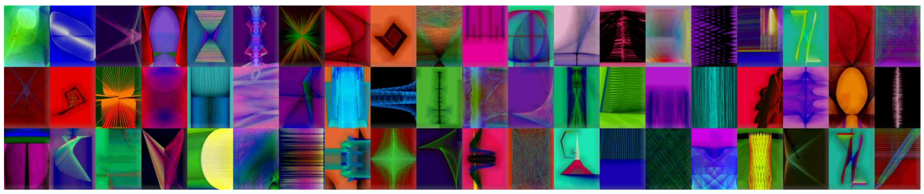


Machado *et al.* attempt to resolve some of the issues of user fatigue through their system with partially interactive evolutionary artists [19]. One mechanism employed is *Automatic Seeding*, where highly fit individuals are stored in common database. Stored individuals can be searched for similarity and later introduced within a run, or used to create initial populations. Preliminary filtering was performed to ensure that the image complexity of each phenotype was sufficient to warrant evaluation from the user - simplistic images could be discarded. The preliminary filtering was later performed on image complexity estimates from the genotype, ensuring that both coordinate terminals were used, and that a sufficient root node was used (noise functions as root nodes were found to be too lacking in desired image structure). Machado also offers a pair of tools, discussed later, for the system to learn aesthetic judgements from the users and assist in automatic evaluation.

Colton, Cook, and Raad explore possible user interface improvements for interactive evolutionary art systems, based on their experiences in adapting their j-ELVIRA particle-based desktop system to tablet devices [35]. Often, it was found that the computational and rendering speed offered by tablet devices was significantly below the full desktop systems designed for, and that the multi-minute delay required to render a small population of individuals could cause many users to lose interest. To reduce the large delay in generating the initial population, the first generation is selected from a collection of 1000 pre-rendered solutions, included with the system to portray a diverse set from which to evolve (examples in Figure 26). To further reduce the amount of non-interactive delay time from the system, mutation and crossover operations have been adjusted, and are performed immediately between images selected by a user. By reducing processing time of mutated or recombined images to seconds, instead of minutes for the entire population, users can more easily maintain their focus of evolving images. The manual selection of parents also showed that many recombined children did not appear as what the user initially thought would be produced.

Custom crossover operators were explored, and a suitable operator was found that tended to produce child images which inherited higher-level attributes more evenly between parents (splitting background colour, initialisation functions, and update functions).

Figure 26: Colton - Examples in i-ELVIRA [35]



4.1.2 Collaborative Aesthetic Fitness

An alternative to a single user evaluating every image is to permit many users to simultaneously assist in evaluating the set of images. The process is known as collaborative interactive evolution. Benefits can clearly be found in this *crowd sourced* method, as individuals can continue to evolve regardless of the continued presence of any single user. However, multi-user evaluation tends to lack the fidelity of single-user evaluation, as images must adapt to the varying aesthetic preferences of many examiners.

Stanley *et al.* create the collaborative interactive evolution program PicBreeder based on neuro-evolution of augmenting topologies (NEAT) and compositional pattern producing network (CPPN) systems [41][49]. By permitting users to publish creations that are of particular interest, users are able to both evolve a single image to their own particular liking, and also collaborate by evolving other published works which catch their eye. As copies of the published items of interest are retained, multiple users are able evolve distinct images to their liking from the same base individual without battling one another for stylistic dominance. By tracking the changes between published images and their source individual, evolutionary contributions for parents can be tracked, and users can be credited for their individual alterations (such as seen in Figure 27). Added social motivators may be used to further retain the interest of users.

Expanding on the PicBreeder system, Zhang *et al.* have provided an online tool - DrawCompileEvolve - for the evolution of user drawings created with primitive shapes [51]. An image drawn by the user, comprised of basic vector shape primitives with colour data, is converted into a CPPN encoding. Using the methods carried over from the PicBreeder system, a user can evolve images on his own, or collaboratively, that are more closely related to the users specified input image. Symmetries, repetitions, colours, and morphologies can be easily evolved from provided input images and popularized published works of others. Figure 28 illustrates a possible history of collaborative mutations over 130 generations.

Figure 27: PicBreeder - Lineage Display [41]

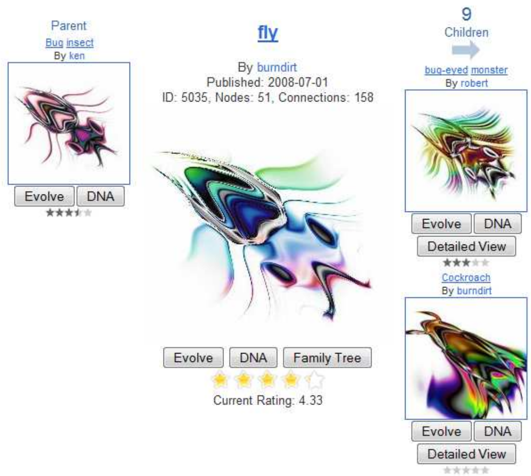
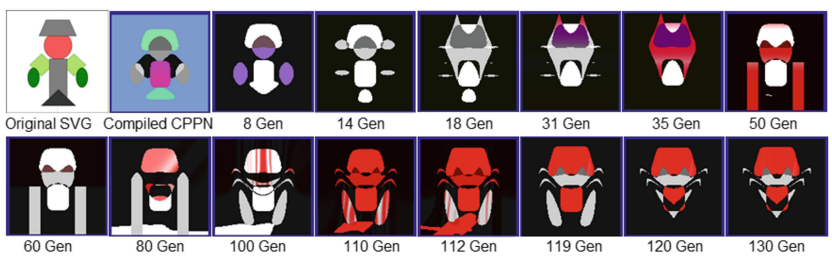


Figure 28: DrawCompileEvolve - Evolution Over 130 Generations [51]



4.2 Automated Evolution

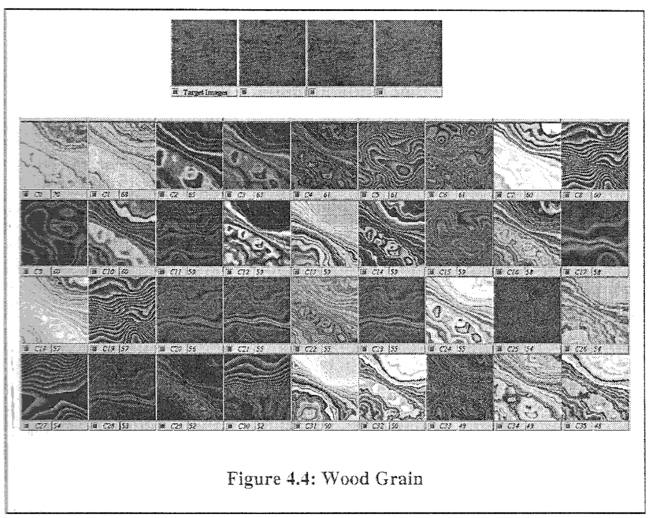
While evolving art using interactive user feedback can produce novel and appealing images, one interpretation of an ideal system would involve the automatic generation of beautiful, interesting images with little user interaction outside of an initial configuration. By synthesising artistic forms, and having novel and aesthetic results that emerge from simple principles, a system may be deemed to be truly creative.

4.2.1 Fitness for Novelty and Similarity

Novelty is one of the more desirable aspect available from an evolutionary search. The ability to explore a wide search space of potential images leads to a wide variety of styles that can be stumbled upon by the user. However, having a filter too loose can give an unconstrained search, which makes it difficult to improve upon a small selection of images. Images can be guided in colour, shape, and style by favouring evolved individuals who closely compare to known example solutions, but a naive approach may not converge with desired traits, or can lead to early convergence if a trivial solution can be reached. A number of approaches have been explored between comparing measures to target images, and measures which compare images relative to other evolved individuals in a population.

Ibraham is often credited with pioneering a completely automated texture evolution system through his development of GenShade [6]. GenShade evolves scenes using Pixar's Renderman shader system, and can permit fitness evaluation from user input, or through comparison to target textures with lumination, colour, and wavelet analysis measures. The evolved images are generated to phenotypes, where features can be extracted in YIQ colour space. Illumination is compared through distance in the Y channel, and chromacity is compared through I and Q distances. An overall distance between all three channels is also available. In the case where automatic texture generation is provided multiple target images, a comparison of the most common wavelets can be used to better match common aspects of the image group (Figures 29, 30). The GenShade automatic evolution mode is a first in texture and filter generation, and provided an impressive array of comparison and fitness metrics for the time.

Figure 29: GenShade - Wood Grain [6]



Saunders and Gero develop the *Digital Clockwork Muse* system [11] based on principles observed by Martindale [58]. The *Law of Novelty* presented by Martindale had concluded that novelty is a powerful force exerted on the development of artistic works. In an attempt to incorporate this law into the digital evolution of images, the *Digital Clockwork Muse* system uses an estimate of novelty based on classification error across previously seen individuals.

Figure 30: GenShade - Sims' Image [6]

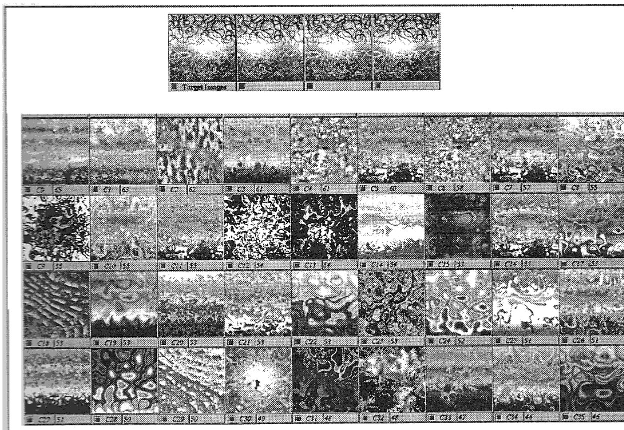


Figure 4.6: Sims's Image

(Target image reproduced with permission from Sims [27])

A number of digital agents are explored across a number of experiments. Each of multiple agents within the system consists of a neural network operating as a self-organizing map. By reducing candidates to points of interest with an edge detection filter, an agent can attempt to classify a candidate to a map of previously seen individuals. By comparing the vector of image values to the neuron weights of the agents self organizing map, an error distance can be computed, and used to determine relative novelty. Mildly interesting candidates can be retained for continued evolution, where those that present greater novelty are first sent to additional agents for peer review and training. A number of further experiments were performed within the system, suggesting the relation between novelty and higher fractal dimension, and that images with too great a variance from other candidates were not immediately scored highly across all agents. Based on configuration, it was found that the grouping of scores amongst agents lead could be lead towards cliques of like-minded agents.

Specific to vector images, there may be additional, more broad measures that we can use in comparison between candidates, or candidate and source images [30]. As vector images are fundamentally composed of primitive objects, there may be simpler functions available that can be used in evaluation or comparison of these primitives. A simplistic square shape, for example, may be compared with metrics using their centre point, width, length, mean colour, etc.. These may be useful for more efficient evaluation for use in hill-climbing, or for frame interpolation in animations.

4.2.2 Aesthetic Fitness for EvoART

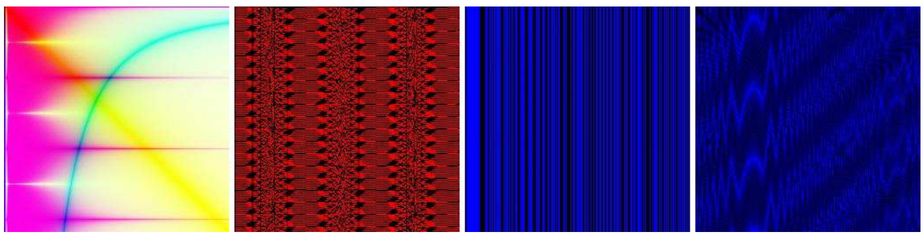
To guide the evolutionary process of evolving art towards appealing results, sets of aesthetic criteria and associated methods for obtaining measurements must be considered. The field of computational aesthetics encompasses measures for mathematically qualifying the appeal of shape, form, and colour.

Implementing Ralph's model of aesthetics, Ross *et al.* experiment with automated evolution using an absolute metric for fitness, scored independent from other candidates [23]. Having evaluated a large collection of fine art, it was observed that higher quality works tended to display a bell-curve distribution among its constituent colour gradients. The primary metrics explored are produced from the bell-curve colour-gradient analysis. How well a candidate's image gradient fit to a normal distribution is determined by the *deviation from normality* measure. The mean and standard deviation of the gradient are also available, measuring the range and changes seen in the gradients. An additional feature test included is a comparison of colour histograms between the evolved individual and a provided image. The colour histogram metric allows for evolution towards a given colour palette, but not necessarily the shape, of the supplied image. Images evolved with Ralph's model of aesthetics employed tended to produce visually interesting -though not always aesthetically pleasing- images that were harmonious with colour (Figure 31). In contrast, images produced without the primary metric tended to be either chaotic or boring.

Figure 31: Ralph's Model of Aesthetics with Colour Target [23]

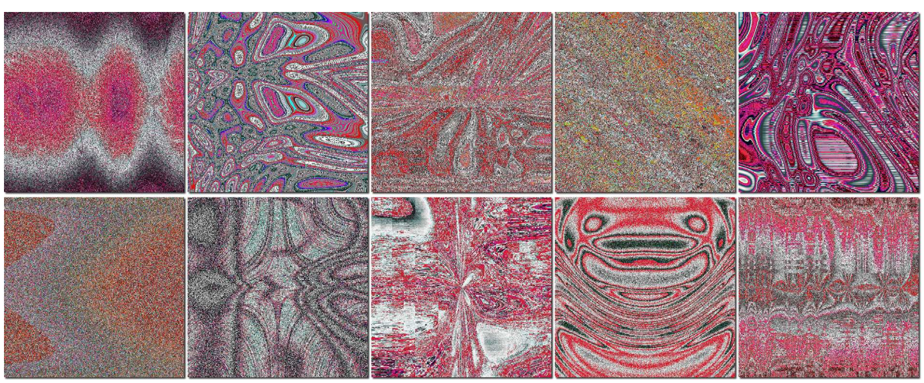
Based on a proposal of computational aesthetics put forth by Machado & Cardoso [7], Atkins, Browne, and Zhang explore alternative fitness measures for automated evolution of aesthetically pleasing images in full colour [28]. The proposal put forth by Machado & Cardoso is an evaluation of fitness based on a ratio of image complexity to information processing complexity. To expand on the previous work of Machado *et al.* which used jpeg compression as estimates for image complexity and fractal compression for processing complexity estimates, Atkins *et al.* consider measures of Shannon Entropy and run-length encodings for image and processing complexities respectively. In addition to the possible matchings of the above image and processing complexity measures, a number of RGB colour space normalization functions were also evaluated. It was found that the original estimate measures of jpeg compression for image complexity and fractal compression for processing complexity produced the most visually appealing results. Highly rated images using Shannon Entropy, Shannon Entropy and fractal compression, jpeg compression and run-length encoding, and jpeg compression and fractal compression are displayed from left to right in Figure 32 below.

Figure 32: Alternate Measures for Information and Processing Complexity [28]



After reviewing commonly employed schemes for automatically evolved image fitness, Heijer & Eiben evaluate and compare the different metrics to determine stylistic features common to each individual scheme [32] [31] [47]. Benford's Law suggests that any list of numbers sampled from real-life phenomena are distributed in a non-uniform way; the leading digits occur with frequency of about one third (the second digits occur 17%, third digits 13%, etc.). Evaluating fitness with this scheme entails taking a histogram of pixel intensities across the evolved image and comparing the fit of distribution to that expected from Benford's Law. Images evolved with this metric tended to have a rather grainy texture, and higher average chroma value than the other tested metrics (figure 33).

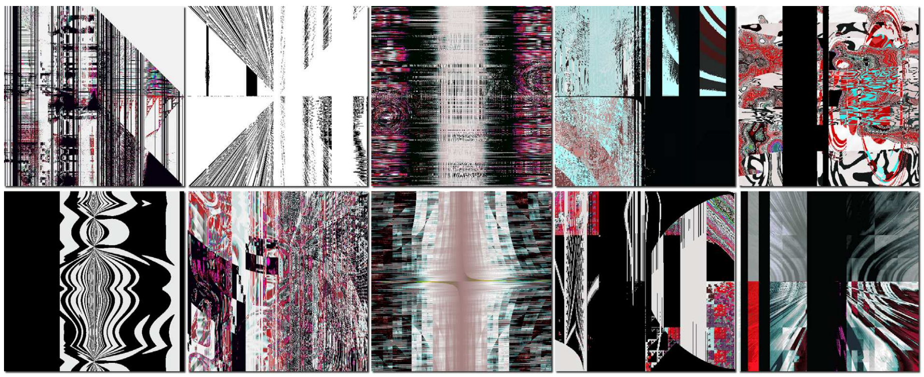
Figure 33: Style Characteristics of Benford's Law Fitness [31]



Global Contrast Factor is a measure of the mean contrast - difference in brightness - across the image. Each pixel determines contrast by comparing the difference of its intensity with neighbouring pixels. The average of these contrast measures is computed for the image at multiple rendered resolutions, and averaged further with a user-defined weight. Images evolved using Global

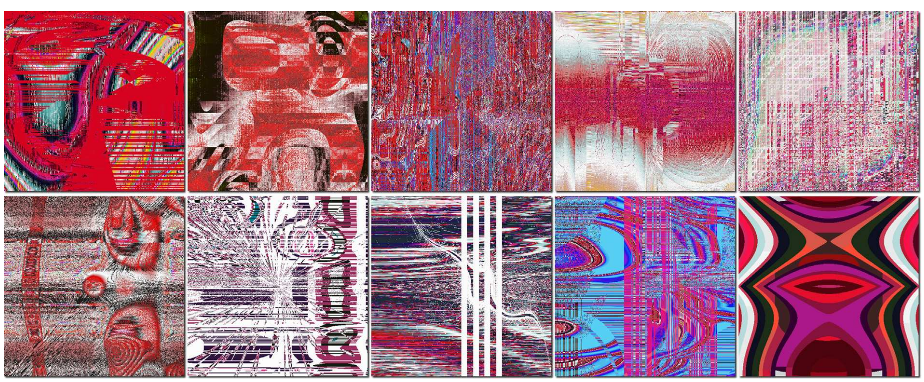
Contrast Factor fitness did display a greater amount of contrasting colours, and due to the evaluation of contrast at various resolutions, tended to be livelier. It was also found that there was a higher standard deviation for brightness, and lower mean chroma measure from among the metrics tested (Figure 34).

Figure 34: Style Characteristics of Global Contrast Fitness [31]



The Ralph's bell curve measure was tested, and was found to display abstract and distinct colour progression. Images tended to have high brightness, and "resembled textures that are used in computer graphics"[47] (Figure 35).

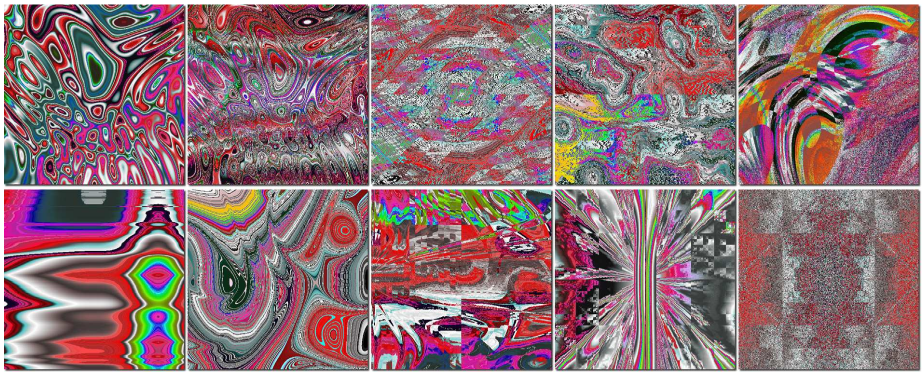
Figure 35: Style Characteristics of Ralph & Ross Fitness [31]



An aesthetic measure based on information theory, was calculated using Shannon Entropy measures on a pixel brightness histogram. High scoring would require intensities to follow a uniform distribution. Evolved images guided by Shannon Entropy were generally colourful, though showed a similar grainy texture as with Benford's Law (Figure36).

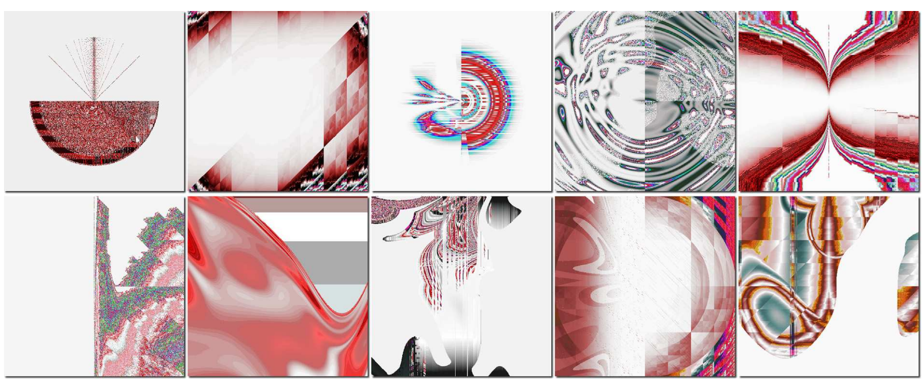
The Machado & Cardoso Complexity ratio was examined, using a slight variant with JPEG2000 compression in lieu of fractal compression. Images

Figure 36: Style Characteristics of Information Theory Fitness [31]



displayed larger white areas and thus had an elevated mean brightness (Figure 37). The structure of the evolved program genotypes was noted simpler than with other measures.

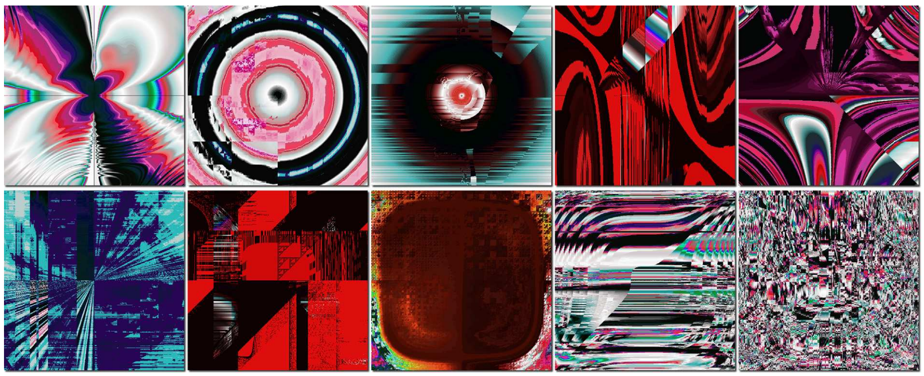
Figure 37: Style Characteristics of Machado & Cardoso Fitness [31]



A measurement of fractal dimension was estimated using a "box-counting" technique, and compared to a previously determined ideal value. Previous work had found peak preference for fractal dimension to be approximately 1.35, where higher values were seen as complex, and lower values boring. Evolved images with approximately ideal fractal dimension tended to favour darker and lower chroma colour values, though strong contrast was found, and with brighter colour than when evolved with the Global Contrast Factor measure (Figure 38).

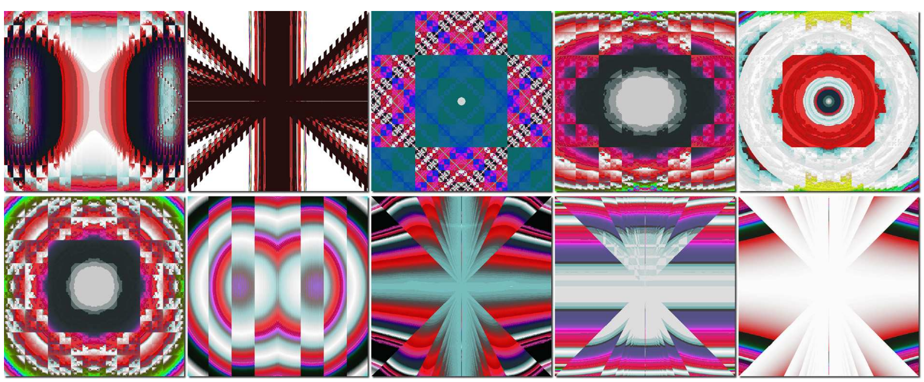
The last fitness scheme evaluated pertains to reflectional symmetry. For each pixel, average colour channel intensity was determined. The image is tested for horizontal, vertical, and diagonal symmetry by determining if the intensity dif-

Figure 38: Style Characteristics of Fractal Dimension Fitness [31]



ference between pixels (a given pixel and its reflection) exceeds a threshold (for the experiments by Heijer & Eibin, a threshold of 5% was used). To prevent simple, monotonous images, the information theory measure was used as a *liveliness* calculation, and used as a weight across each form of symmetry. The resultant images showed strong combinations of vertical, horizontal, and diagonal symmetry, and a variety of texture and colour can be seen (Figure 39).

Figure 39: Style Characteristics of Reflectional Symmetry Fitness [31]



Machado *et al.* continue a focused investigation in image feature selection and novelty within the application of automated aesthetic selection [53]. Adapting on the work of Gero and Saunders [11], two banks of images are loaded, *internal* and *external*, where internal images are those produced by the system without any explicit fitness and external images are existing pieces of art which display desired traits. By evaluating evolved individuals, and comparing the feature measures as more similar to the external or internal banks, it is expected that highly (automatically) rated individuals will display fewer of the default traits inherent to the systems synthesis engine.

Similar features as found in their previous work on partially automated systems are evaluated for image classification. Some of the initial metrics calculated include jpeg complexity, fractal and jpeg complexity per colour channel, average and deviation per colour channel, and fractal complexity. A number of additional filters are first applied to both improve efficiency and offer alternative viewpoints. Each image, prior to feature extraction, is converted to 128x128 24-bit bitmap. In addition to the features extracted from this standardized image (the entire image, each quadrant, and middle subsection), the same features are also extracted after each of Sobel and Canny edge detection, distance transform, quantization, and salience filters have been applied. The feature extractor ultimately provides 804 possible features to the feature selector and artificial neural network.

A subset of the possible features are chosen through use of the *CfsSubsetEval* method, which evaluates features for their information redundancy and class correlation. After the initial population has been generated, a predetermined amount of the top features are used for training and evaluation with the neural network.

Figure 40: Machado *et al.* - Classifier Training [53]

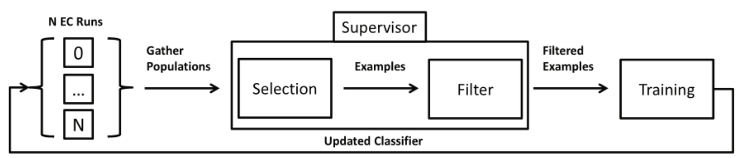
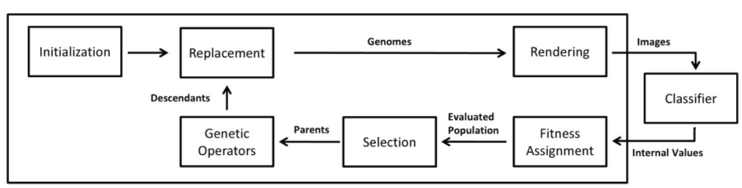


Figure 41: Machado *et al.* - Automatic Classification of Evolved Images [53]



4.2.3 Multi-objective Fitness

In computational models of aesthetics, it is seldom found that the appeal of images can be suitably determined with a single measure. Rather, the criteria desired can encompass a vast scope of complimenting and contrasting features. Ongoing research continues to explore how to adapt multi-objective fitness and optimization schemes for use in determining aesthetic fitness.

In his early work with multi-objective EvoART, Greenfield explores the use of a variant Pareto ranking algorithm with simple measures on image segmentations [15]. Referring to his previous experimentations with image segmentation, Greenfield renders 32x32 pixel thumbnails of each evolved individual, and performs an image segmentation for feature processing. The segments are indexed by area, and have provided metrics which, in addition to segment area, include boundary length and number of region adjacencies.

In measuring and evaluating an individual on any one of the segment metrics alone, individuals would tend to produce common, less interesting attributes. A heavy weight assigned to segment area would direct individuals to display stripes of equal width, where focussing on boundary length tended to produce individuals displaying concentric circles.

Using multiple functions composed of the available segment metrics, Greenfield was able to evolve interesting images with distinct segmentation. Reproduction was performed with a steady state method. Individuals were chosen for reproduction via a tournament selection, with use of an intermediate fitness function (comparing largest area to total boundary lengths). Figure 43 displays promising images which had converged with one set of fitness functions after 20 generations. It was noted that the fitness criteria used with the multi-object method was still delicate, as certain features tended to dominate the front. While a run may have a diverse Pareto front, undue pressures may still be found, as in Figure 42 (displayed after 400, and 700 generations).

Figure 42: Greenfield - Multi-Objective Dominance [15]

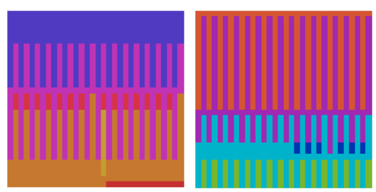
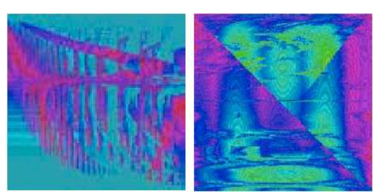


Figure 43: Greenfield - Multi-Objective Balance [15]



Ross and Zhu have also experimented with Pareto ranking of aesthetic fitness, adapting previous work in texture feature tests to use a pure Pareto ranking scheme in lieu of a weighted sum [17]. In addition to a simplistic Pareto ranking, A pair of tests were performed where divergence strategies were also employed. Building on the Ross and Wiens *Gentropy* system [13], a number of image feature tests were available for use as individual objectives. A direct texture image comparison was available, giving a sum of RGB colour space distances between an individual and target texture. For a more position-independent colour measure, histograms across colours were also produced, where each image was first quantized to provide histogram bins, and the resultant distances across the pairs of target and individual histogram bins were measured. In an attempt to provide an estimated measure of shape, pattern, and edge similarity, wavelet decompositions were compared across grayscale renditions of the target image and individual. Smoothness was also evaluated through the distance of the images after run through an edge-detection filter.

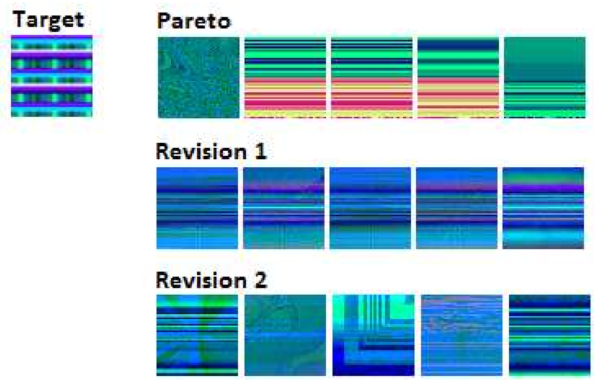
Where a pure Pareto ranking was used as a baseline, it was often seen that results would prematurely converge, as marginally improved individuals would quickly dominate the top rank of solutions. The first scoring adjustment tested was the use of nearest neighbour distance within a given rank. Individuals that were more diverse (across the image features tested) would have their scores scaled to the maximum score found in that rank, while less diverse individuals would be scaled towards the lower scores found in their rank. While tournament selection was employed (and thus having selection sensitive only to relative score differences), less diverse individuals would become more likely to be replaced by future, more diverse candidates, and would also consequently lessen the ability of a single similar set of candidates from dominating a rank. Further refining the Pareto scoring method, a second revision was used by normalizing each of the candidates' image feature scores to a sub-rank across other individuals within the Pareto rank group. This helped relieve concerns of one feature metric providing greater variability than others.

After experimentation, pure Pareto ranking displayed an expectedly large amount of convergence between solutions, where the revised schemes led to stronger colour match (see Figure 44). While solutions may not have been greatly improved, a greater amount of diversity was seen with the second scoring revision.

One common issue which remains with Pareto-style scoring is the presence of outliers which display great fitness in only 1 or a few of the objectives, and poor scores in the others. As multi-objective aesthetic fitness tends to require many measures to be strongly scored, the outliers from Pareto ranking are often undesirable. To mitigate this, Ross and Bergen investigate the use of a *sum of ranks* approach to scoring within the context of evolutionary art [29]. Using their JNetic texture system, images were evaluated using the mean, standard deviation, and deviation from normality from Ralph's model of aesthetics [23], as well as a colour histogram distance. Both standard Pareto, and sum of ranks scoring were used to perform multiple experimental runs. While runs using the sum of rank scoring did not present results that were as diverse, the overall quality of images was found to be much higher than Pareto ranking.

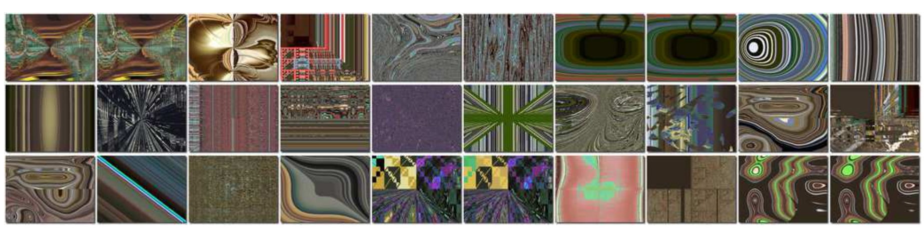
Previously, Heijer and Eiben explored automatic evolution of images using individual metrics for aesthetic fitness and determined styles that were common to each metric [31]. Expanding on their work, Heijer and Eiben assess the inher-

Figure 44: Ross & Zhu, Comparison of Pareto Divergence Strategies [17]



ent styles that images displayed when automatically evolved using combinations of the previously tested metrics [36]. For each run, two feature scores were chosen from among Benford Law, Global Contrast Factor, and Ross & Ralph Bell Curve. The Arabitat evolutionary system was utilized to evolve images, and compare them using combinations of the above image features with a Pareto ranking system. In comparison to the images evolved automatically with single aesthetic metrics (Figures 33, 34, 35), distinct styles can be seen where the images dominate strongly across both objectives, such as in Figures 45, 46, and 47.

Figure 45: Style Characteristics of Benford Law and Ross & Ralph [36]



Reynolds extends his evolutionary art tool-kit with a prototype for specifying simple fitness routines from user-provided criteria [39]. In an initial experiment, images were evolved with a set of multi-objective criteria for evolving a predominantly grey image with a small amount of accent colours. Some manual tuning was required, though suggestions were made for expanding the criteria to fit a wider range of possible user requirements.

Five required fitness criteria were identified for the experiment. A fraction of “good pixels” was required, measuring the number of pixels below the grey threshold, or above a colour threshold. The distance from the ratio of coloured pixels to the tuned target value, and a measure for how close the suitable colour

Figure 46: Style Characteristics of Global Contrast and Ross & Ralph [36]

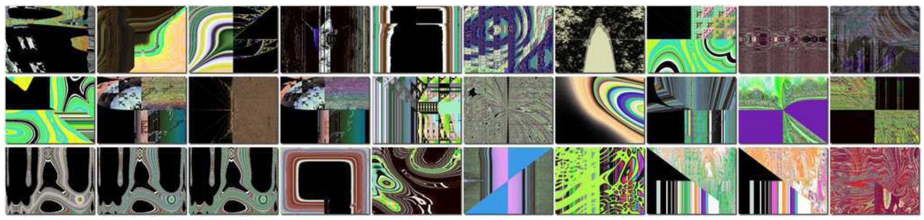
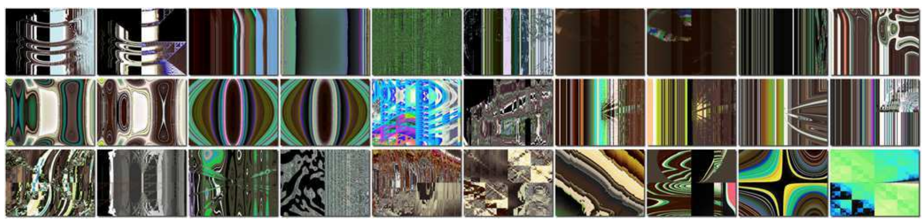


Figure 47: Style Characteristics of Benford Law and Global Contrast [36]



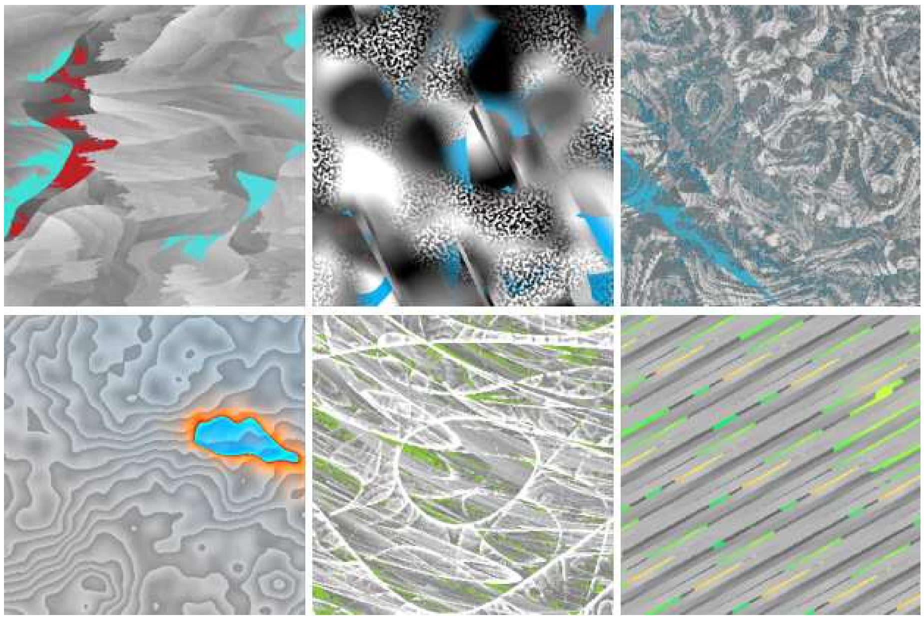
and grey pixels are to "midrange" values were also used. To ensure contrast, a number of samples were taken across the evolved image and scored for meeting a contrast threshold. The size of the bounding box containing all suitable coloured pixels was also made available to promote dispersed and non-uniform colours. Given the multiple features required for an ideal image, a traditional multi-objective approach is initially considered. However, the multiple features were merged into a single metric by normalizing the output of each feature and multiplying the scores together to create a product of fractions. Two useful properties were found with this approach: any low score limits the maximum obtainable score for the individual, and adjustments to any single objective score adjusts the overall score in the same direction.

Images were often found matching the high-level description of "gray with an accent colour" within 10 to 30 generations, though often with a feel of sameness to images evolved from other runs (Figure 48). While adjustments may be made to preserve novelty of the produced results, the fitness metric for the user-specified criteria was sufficient to consistently find suitable results, and the approach may be adapted for the specification of other high-level descriptors.

4.3 Hybrid & Other Systems

A number of attempts have been made, through use of learning system techniques, to create updating fitness functions based on the ongoing feedback of users. The ideal solution would be able to run without user interaction for a number of generations, providing distinct and novel results which conform to the users' artistic preferences inferred from previous evaluations. The flow of

Figure 48: Evolving Images from High-Level Descriptions: Gray with an Accent Colour [39]

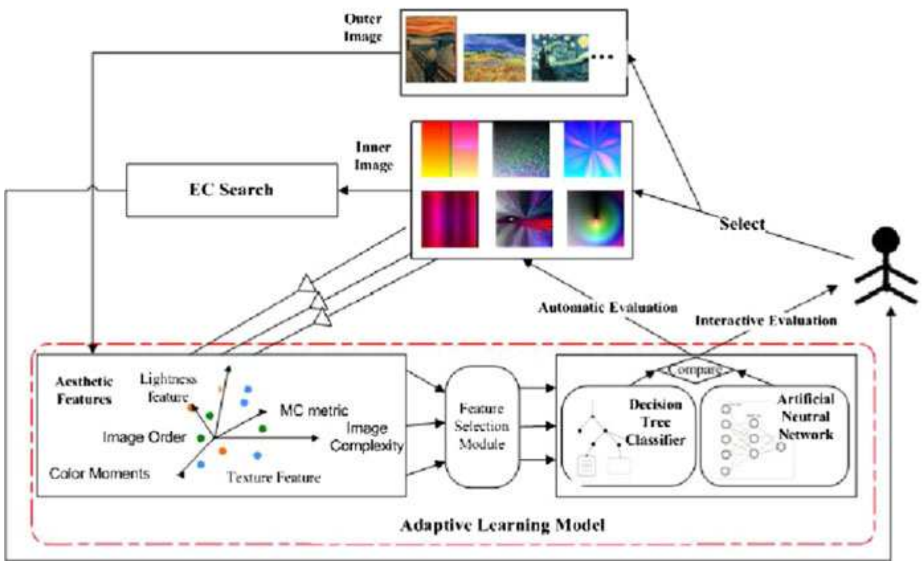


UI and improvement in learning system techniques has improved greatly since the prototypical work of Baluja *et al.* [4], using manual score assignment and pixel-mapped neural networks.

Machado *et al.* begin their investigation into aesthetic learning systems with their work in partially interactive evolutionary artists [19]. Branching from their work in automated evolutionary systems, the primary fitness measure used is ratio of information and processing complexities. The IEC system NEvAr is adjusted, allowing a user to either rank individuals as before, or to permit automatic ranking of all individuals in a given generation. Optimum expected values (for subsections of complexity measure) can be explicitly set by the user, or a single particular image can be specified as ideal (from which the optimal values can be derived). While the use of partially automatic options was often beneficial after practice, it was noted that initial attempts were disappointing, with a suspicion that valuing images for novelty ended up providing an unfocused search with the given fitness measure.

Not set back by the issues found in the partially interactive system, Machado *et al.* worked to develop artificial art critics (AACs). These artificial critics evaluate candidate images across metrics including jpeg complexity, fractal and jpeg complexity for each colour channel, average and deviation across each colour channel, and a pair of fractal complexity measures. 33 metrics are evaluated across the entire image, and each of the 33 metrics are also evaluated in the quadrants and middle subsection of the image. A full 198 measures are forwarded to an artificial neural network, which has been trained to identify artwork from one of several possible painters. While certain artists showed a higher classification success rate, artists were correctly identified by the artificial critics between 93% and 97% of the time.

Figure 49: Li - Learning Aesthetic Judgement Through User Feedback [45]



Improving on the concept of using learning systems to classify images from aesthetic judgements, Li *et al.* employ a pair of classifiers - permitting users to defer evaluations to a trained system, and alleviate user fatigue [45]. Various features were extracted for use in training the system and classifying images. With a focus on exploring multiple colour spaces, features included mean, standard deviation, and skewness for each of hue, saturation, and lightness measures, as well as information entropy for HSL, RGB, and Y_{709} spaces. Additional features included image and processing complexity, Benford's law correlation in lightness channels, and ranges of local binary patterns. Measures are extracted across the entire image, as well as in each quadrant, and a middle subsection (similar to Machado [19]). The features are reduced from a count of 150 to a handful of significant features, determined by ranking *InfoGainAttributeEval* evaluations.

Based on the selected subset of extracted features, both a C4.5 decision tree and multi-layer perceptron neural network can be trained to classify future images from an initial interactive evaluation of a generation. The decision tree and neural net are continually updated and trained when user evaluation is provided. The user can score evolved images with a fixed number of distinct ratings. The classifier is trained to better associate the user-given rating category with scores of the selected image feature measures. Users can choose to have the current generation automatically evaluated, in which case the learning system is not changed, or the user can provide explicit evaluations, which the learning system will use to provide revised judgement, as seen in Figure 49.

Absent of automatic classification and learning systems, a simpler approach used by Machado *et al.* is to present a straightforward interface for design and adjustment of fitness criteria [48]. They had explored artistic rendering of ant trail paintings, and present a number of potential fitness features based on generation process statistics to the user. A seemingly intuitive interface has been provided to procure the relative weights of the features as specified by the user. To assist in portraying the potential effects of fitness weight adjustment, the icons displayed next to each weight slider are actively updated to display the effects on a micro scale. While many systems can provide configuration options for fitness, a key difference is that these fitness configurations can be updated after each generation. While no subsystem is included to learn about the preferences of the user, a mix of both interactive and automated fitness is achieved. A user can actively pressure future generations to display tailored aesthetic preferences by updating the fitness weights, yet repeated generations can be produced without significant analysis from the user.

5 Other Approaches

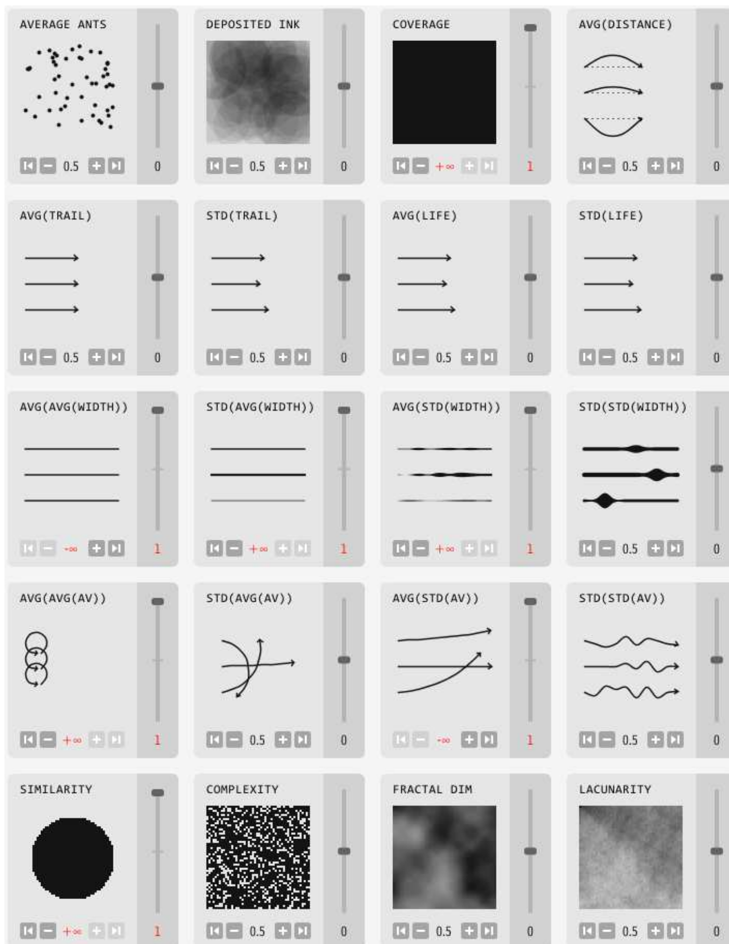
The ways in which the principles of evolution can be applied to evolutionary art is vast. Many modern evolutionary design and art systems warrant additional explanation of their processes. A few notable classifications of such systems are outlined below, where research on these display promise.

5.1 Particle Swarm

In Sims' original work [3], the prospect of using time as a variable was proposed for the creation of video. However, a number of artists have since toyed with the idea of adjusting primitives with translation and other operations across time steps, and maintaining the previously drawn segments to create create an appearance of motion. One particularly noteworthy rendering method is the display of swarm behaviour and paths over a short amount of time steps.

Colton *et al.*, in their j-ELVIRA interactive evolutionary art system, employed a GP representation to evolve the routines defining initialization and updates within a particle swarm [35]. For each individual, a background colour is defined, and 6 trees are evolved to represent the initial coordinate and colour channel (RGBA) values of each particle. An additional 6 trees are evolved to determine the change in these particle values across time-steps, accepting added variables for the current coordinate and colour values, as well as variables for particle and time-step indices. For each time-step, lines from the previous particle positions to the new positions are drawn, and the canvas is blurred.

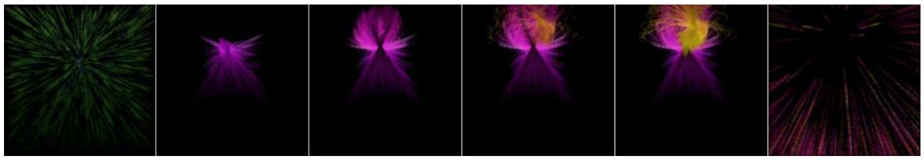
Figure 50: Machado - Ant Painting Fitness Weights [48]



Particle swarms are used to create animations of fireworks bursts by Trujillo *et al.* through an online collaborative evolutionary system [46]. Linear GP is used to encode sequences of particle flow and parameter adjustments for a given individual. Possible genomes consist of either rendering for some magnitude of time, or toggling behaviour flags, such as attraction/repulsion, acceleration/deceleration, tracing, and movement in cardinal directions. Figure 51 displays a sampled sequence of frames rendered in this method.

To permit for more complex scenes, a pair of crossover operators are provided. *2-point crossover* operates as expected, and will swap certain sub-lists of parents, where an *append crossover* will produce a pair of children by concatenating parents in both possible orders. A steady-state population is used, and individuals are ranked through collaborative user interaction, receiving a fitness

Figure 51: Fireworks - Animation Frame Samples [46]



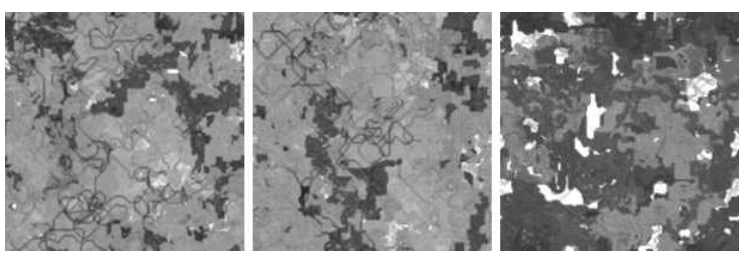
that is a ratio of *likes* to times viewed.

5.2 Ant Trail Painting

Another alternative form of evolved art includes the use of visualizations for ant colony optimization. Greenfield [20] explores the use of evolving behavioural processes in ant models, producing artistic works representative of ant scent trails. All individual ants in the generation were applied on a canvas, where each individual ant evolved attributes of colours to deposit, colours to follow, random movement probabilities, amount to veer off course, and probability to change direction when the followed colour threshold is met.

Experimentation was done across a number of fitness functions, where individuals were evaluated based on the number of squares visited, the number of times scent-following behaviour was exhibited, the ratio of the first two measures, the sum of the first two measures, and the product of the first two measures. The latter-most fitness was found to be most interesting, where more ideal solutions would tend to display higher amounts of shading and finer structure detail, as seen in Figure 52.

Figure 52: Greenfield - Ant Paintings [20]



Machado *et al.* [48] add to the work of Greenfield, using GA representation with a larger set of parameters encoded in the genotype. Encodings capture sets of multiple ants with variables for energy gain and decay scaling, deposit and search node size and transparency, initial energy and death threshold, perturbation of transparency and angular velocity of offspring, initial coordinates and orientation, and sensory sensitivity and direction. A simple 2-point crossover and Gaussian mutation operators were used with tournament selection.

After each individual has been run for a number of time steps, images features of complexity and similarity are evaluated from the resultant canvas. For use in evolution of an NPR filter, the similarity metric compares the individual's canvas to the base image through use of the root mean square error of jpeg compression. Additional fitness metrics were made available to users of the system, permitting them to specify the weight of various metrics including average number of ants, deposited ink measures, coverage, average distance travelled, similarity, and complexity, among others. The perceived feel that the filtered images gives can be greatly influenced by the user-assigned fitness weights, as seen in Figure 53.

Figure 53: Machado - Ant Painting Feature Preferences [48]

5.3 Compositional Pattern-Producing Networks

PicBreeder provides a milestone of research in collaborative interactive evolution. The system by Stanley *et al.* [41] permits users to both evolve images to their individual tastes, and to collaborate by evolving images previously published by other users. Continued evolution of an individual by a single user allows for increased fidelity in obtaining the treats desired by the single user. By publishing an image, and evolving other published images, the system permits for interesting exploration of solutions, using interactive aesthetic selection, while a given user yields their attention from the application. Upon their return, users are still able to further refine their individual, as copies of the base published individuals are retained.

To permit for easier composition of individuals while also keeping track of an individuals heritage, the use of compositional pattern-producing networks (CPPN) seemed desirable. CPPNs may be seen as a variation on artificial neural networks, where a more varied selection of activation functions are available. While there are a number of ways in which a CPPN may be initialized and evolved, PicBreeder employs neuroevolution of augmenting topology systems (NEAT), which adds nodes and links to a base network. Child individuals are combined through various operators to produce the parent image, as seen in Figures 54 and 55 [49].

Figure 54: CPPN - Composition [49]

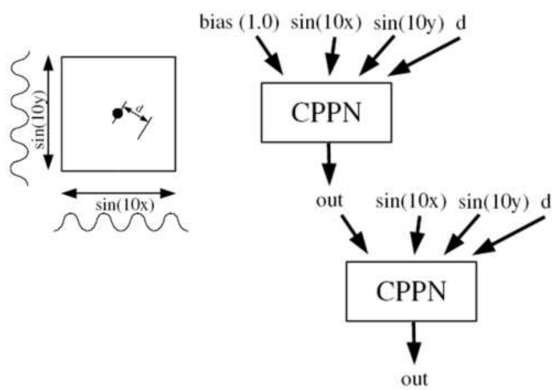
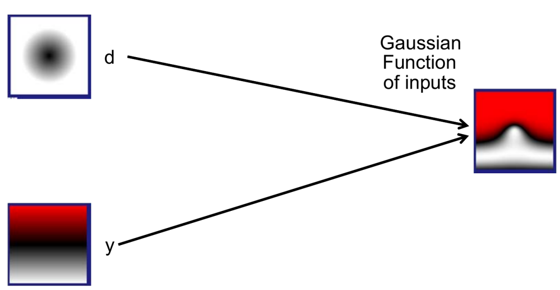


Figure 55: CPPN - Composition 2 [49]



Zhang *et al.* expand the capabilities and utility of the PicBreeder system within their implementation of DrawCompileEvolve [51]. This system uses a similar method to Stanley's for representation, and performing and tracking alterations of individuals. The effort required to produce moderately developed images, however, is significantly lessened by the ability of a user to draw vector art objects which are then simply converted into compositional pattern-producing network encodings. This provides a much easier path for users to evolve higher-level and higher-detail images.

6 Conclusion

Work in the area of computational aesthetics and evolutionary art has since inspired artists, both technical and not, to explore works generated with novel search techniques and evolutionary computation. Since Sims' prototypical work, the amount of interest and number of publications in the topic has shown great optimism.

While a focus was maintained on the creation of planar imagery, there are countless mediums where artistic and aesthetic creativity can be expressed. Within even the realm of 2D textures and filters, a hugely varying amount of representations, languages, and aesthetic criteria were explored - with many still left unexplored. Many works of art are being evolved through 3D models, 2D and 3D animations, and other acoustic mediums with large measures of success.

There is still great room for improvement, and many open questions remain in the field of computational aesthetics. However, there have been numerous models of aesthetics proposed, covering great strides in the development of suitable aesthetic fitness measures. Advances have been made in directing searches towards novelty, combinations of aesthetic estimates, and higher-level visual criteria. Work in multi-objective scoring has also been improved to permit quicker discovery of ideal images without heavy sacrifices to individual measures.

With many of the proposed measures and representations, there can often be a flexibility that is offered to solutions not anticipated by the designer. It is hoped that systems will continue to present appealing and unexpected results, and that future development will further foster what may be considered a creative emergence.

Figure 56: Machado - Hand [50]

References

[1] John H Holland. Adaptation in natural and artificial systems: An introductory analysis with applications to biology, control, and artificial intelligence. 1975.

[2] David E Goldberg. Genetic algorithms in search, optimization, and machine learning. *Addion wesley*, 1989, 1989.

[3] Karl Sims. Artificial evolution for computer graphics. *ACM Computer Graphics*, 25(4):319–328, July 1991. SIGGRAPH '91 Proceedings.

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

[5] Christian Jacob. Genetic L-system programming. In Yuval Davidor, Hans-Paul Schwefel, and Reinhard Männer, editors, *Parallel Problem Solving from Nature III*, volume 866 of *LNCS*, pages 334–343, Jerusalem, 9-14 October 1994. Springer-Verlag.

[6] Alaa Eldin M Ibrahim. Genshade: an evolutionary approach to automatic and interactive procedural texture generation. 1998.

[7] Penousal Machado and Amílcar Cardoso. Computing aesthetics. In *Advances in artificial intelligence*, pages 219–228. Springer, 1998.

[8] S Marsili Libelli and P Alba. Adaptive mutation in genetic algorithms. *Soft computing*, 4(2):76–80, 2000.

[9] Andrea L. Wiens and Brian J. Ross. Gentropy: Evolutionary 2D texture generation. In Darrell Whitley, editor, *Late Breaking Papers at the 2000 Genetic and Evolutionary Computation Conference*, pages 418–424, Las Vegas, Nevada, USA, 8 July 2000.

[10] Alan Dorin. Aesthetic fitness and artificial evolution for the selection of imagery from the mythical infinite library. In *Advances in artificial life*, pages 659–668. Springer, 2001.

[11] Rob Saunders and John S Gero. The digital clockwork muse: A computational model of aesthetic evolution. In *Proceedings of the AISB*, volume 1, pages 12–21, 2001.

[12] Ken Perlin. Improving noise. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 681–682. ACM, 2002.

[13] Andrea L. Wiens and Brian J. Ross. Gentropy: evolving 2D textures. *Computers and Graphics*, 26(1):75–88, February 2002.

[14] Processing Foundation. Processing. https://processing.org/, 2003. [Online; accessed 2015-09-01].

[15] Gary R Greenfield. Evolving aesthetic images using multiobjective optimization. In *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, volume 3, pages 1903–1909. IEEE, 2003.

[16] Adam Hewgill and Brian J. Ross. The evolution of 3D procedural textures. In Bart Rylander, editor, *Genetic and Evolutionary Computation Conference Late Breaking Papers*, pages 146–147, Chicago, USA, 12 July 2003.

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

[18] Lode Vandevenne. Texture generation using random noise. http://lodev.org/cgtutor/randomnoise.html, 2004. [Online; accessed 2015-09-03].

[19] Penousal Machado, Juan Romero, Amílcar Cardoso, and Antonino Santos. Partially interactive evolutionary artists. *New Generation Computing*, 23(2):143–155, 2005.

[20] Gary Greenfield. Genetic learning for biologically inspired aesthetic processes. *International Journal on Artificial Intelligence Tools*, 15(04):577–598, 2006.

[21] Martin Hemberg, Una-May OReilly, Achim Menges, Katrin Jonas, M Goncalves, and S Fuchs. Exploring generative growth and evolutionary computation for architectural design. *Art of Artificial Evolution, Springer, Heidelberg*, 2006.

[22] Evelyne Lutton. Evolution of fractal shapes for artists and designers. *International Journal on Artificial Intelligence Tools*, 15(04):651–672, 2006.

[23] Brian J. Ross, William Ralph, and Hai Zong. Evolutionary image synthesis using a model of aesthetics. In Gary G. Yen, Lipo Wang, Piero Bonissone, and Simon M. Lucas, editors, *Proceedings of the 2006 IEEE Congress on Evolutionary Computation*, pages 3832–3839, Vancouver, 6-21 July 2006. IEEE Press.

[24] Genetic Programming Inc. www.genetic-programming.com. www.genetic-programming.com, 2007. [Online; accessed 2015-09-01].

[25] Marc Hull and Simon Colton. Towards a general framework for program generation in creative domains. In *Proceedings of the 4th international joint workshop on computational creativity*, pages 137–144, 2007.

[26] Simon Colton, Michel F Valstar, and Maja Pantic. Emotionally aware automated portrait painting. In *Proceedings of the 3rd international conference on Digital Interactive Media in Entertainment and Arts*, pages 304–311. ACM, 2008.

[27] W. B. Langdon. A field guide to genetic programming, March 2008. Thank you to Seminar 08051.

[28] Daniel L Atkins, Roman Klapaukh, Will N Browne, and Mengjie Zhang. Evolution of aesthetically pleasing images without human-in-the-loop. In *IEEE Congress on Evolutionary Computation (CEC 2010)*, Barcelona, Spain, 18-23 July 2010. IEEE Press.

[29] Steven Bergen and Brian J. Ross. Evolutionary art using summed multi-objective ranks. In Rick Riolo, Trent McConaghy, and Ekaterina Vladislavleva, editors, *Genetic Programming Theory and Practice VIII*, volume 8 of *Genetic and Evolutionary Computation*, chapter 14, pages 227–244. Springer, Ann Arbor, USA, 20-22 May 2010.

[30] Simon Colton. Stroke matching for paint dances. In *Proceedings of the Sixth international conference on Computational Aesthetics in Graphics, Visualization and Imaging*, pages 67–74. Eurographics Association, 2010.

[31] E. den Heijer and A. E. Eiben. Comparing aesthetic measures for evolutionary art. In Cecilia Di Chio et al., editor, *EvoMUSART*, volume 6025 of *LNCS*, pages 311–320, Istanbul, 7-9 April 2010. Springer.

[32] Eelco den Heijer and A. E. Eiben. Using aesthetic measures to evolve art. In *IEEE Congress on Evolutionary Computation (CEC 2010)*, Barcelona, Spain, 18-23 July 2010. IEEE Press.

[33] Penousal Machado, Henrique Nunes, and Juan Romero. Graph-based evolution of visual languages. In *Applications of Evolutionary Computation*, pages 271–280. Springer, 2010.

[34] Robert I. McKay, Nguyen Xuan Hoai, Peter Alexander Whigham, Yin Shan, and Michael O'Neill. Grammar-based genetic programming: a survey. *Genetic Programming and Evolvable Machines*, 11(3/4):365–396, September 2010. Tenth Anniversary Issue: Progress in Genetic Programming and Evolvable Machines.

[35] Simon Colton, Michael Cook, and Azalea Raad. Ludic considerations of tablet-based evo-art. In *Applications of Evolutionary Computation*, pages 223–233. Springer, 2011.

[36] Eelco den Heijer and AE Eiben. Evolving art using multiple aesthetic measures. In *Applications of Evolutionary Computation*, pages 234–243. Springer, 2011.

[37] Eelco den Heijer and Agoston Endre Eiben. Evolving art with scalable vector graphics. In Natalio Krasnogor et al., editor, *GECCO '11: Proceedings of the 13th annual conference on Genetic and evolutionary computation*, pages 427–434, Dublin, Ireland, 12-16 July 2011. ACM.

[38] Ashkan Izadi, Vic Ciesielski, and Marsha Berry. Evolutionary non photo–realistic animations with triangular brushstrokes. In *AI 2010: Advances in Artificial Intelligence*, pages 283–292. Springer, 2011.

[39] Craig Reynolds. Evolving textures from high level descriptions: Gray with an accent color. In *Applications of Evolutionary Computation*, pages 384–393. Springer, 2011.

[40] Craig Reynolds. Interactive evolution of camouflage. *Artificial Life*, 17(2):123–136, Spring 2011.

[41] Jimmy Secretan, Nicholas Beato, David B D'Ambrosio, Adelein Rodriguez, Adam Campbell, Jeremiah T Folsom-Kovarik, and Kenneth O Stanley. Picbreeder: A case study in collaborative evolutionary exploration of design space. *Evolutionary Computation*, 19(3):373–403, 2011.

[42] Steven Bergen and Brian J Ross. Automatic and interactive evolution of vector graphics images with genetic algorithms. *The Visual Computer*, 28(1):35–45, 2012.

[43] Eelco den Heijer and AE Eiben. Evolving pop art using scalable vector graphics. In *Evolutionary and Biologically Inspired Music, Sound, Art and Design*, pages 48–59. Springer, 2012.

[44] Steve Bergen and Brian J. Ross. Aesthetic 3D model evolution. *Genetic Programming and Evolvable Machines*, 14(3):339–367, September 2013. Special issue on biologically inspired music, sound, art and design.

[45] Yang Li, Changjun Hu, Leandro L. Minku, and Haolei Zuo. Learning aesthetic judgements in evolutionary art systems. *Genetic Programming and Evolvable Machines*, 14(3):315–337, September 2013. Special issue on biologically inspired music, sound, art and design.

[46] Leonardo Trujillo, Mario Garcia-Valdez, Francisco Fernandez de Vega, and Juan-J. Merelo. Fireworks: Evolutionary art project based on evospace-interactive. In Luis Gerardo de la Fraga, editor, *2013 IEEE Conference on Evolutionary Computation*, volume 1, pages 2871–2878, Cancun, Mexico, June 20-23 2013.

[47] Eelco den Heijer and A. E. Eiben. Investigating aesthetic measures for unsupervised evolutionary art. *Swarm and Evolutionary Computation*, 16:52–68, 2014.

[48] Penousal Machado, Tiago Martins, Hugo Amaro, and Pedro H Abreu. An interface for fitness function design. In *Evolutionary and Biologically Inspired Music, Sound, Art and Design*, pages 13–25. Springer, 2014.

[49] Kenneth O Stanley. Generative and developmental systems tutorial. In *Proceedings of the 2014 conference companion on Genetic and evolutionary computation companion*, pages 765–794. ACM, 2014.

[50] Penousal Machado. Evolving assemblages. http://cdv.dei.uc.pt/evolving-assemblages/, 2015. [Online; accessed 2015-09-01].

[51] Jinhong Zhang, Rasmus Taarnby, Antonios Liapis, and Sebastian Risi. Drawcompileevolve: Sparking interactive evolutionary art with human creations. In *Evolutionary and Biologically Inspired Music, Sound, Art and Design*, pages 261–273. Springer, 2015.

[52] Carlos Coello Coello, Gary B Lamont, and David A Van Veldhuizen. *Evolutionary algorithms for solving multi-objective problems.* Springer Science & Business Media, 2007.

[53] João Correia, Penousal Machado, Juan Romero, and Adrián Carballal. *Feature selection and novelty in computational aesthetics.* Springer, 2013.

[54] Richard Dawkins. *The blind watchmaker: Why the evidence of evolution reveals a universe without design.* WW Norton & Company, 1986.

[55] David S Ebert. *Texturing & modeling: a procedural approach.* Morgan Kaufmann, 2003.

[56] John R. Koza. *Genetic Programming II: Automatic Discovery of Reusable Programs.* MIT Press, Cambridge Massachusetts, May 1994.

[57] John R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection.* MIT Press, Cambridge, MA, USA, 1992.

[58] Colin Martindale. *The clockwork muse: The predictability of artistic change.* Basic Books, 1990.