EVOLVING CREATIVE PORTRAIT PAINTER PROGRAMS USING DARWINIAN TECHNIQUES WITH AN AUTOMATIC FITNESS FUNCTION

Steve DiPaola
School of Interactive Arts and Technology,
Simon Fraser University
10153 King George Highway,
Surrey, British Columbia, V3T 2W1, CANADA
steve@dipaola.org
http://www.dipaola.org

We experiment with computer creativity by employing and modifying techniques from evolutionary computation to create a related family of abstract portrait painter programs. In evolutionary art, most systems evolve paintings by allowing the artist to selectively breed the artwork 'by hand' from a selection of the currently evolved population. Our system differs in that it uses an automatic 'creative fitness function' which allows the evolutionary process to run without stopping for 'creative human intervention'. A recent type of Genetic Programming (GP) is used called Cartesian GP, which has several features that allow our system to favour creative solutions over optimized solutions.

INTRODUCTION

A new field that has emerged over the last 10 years in computer 'artificial intelligence' systems is creative evolutionary systems. Creative evolutionary systems use techniques from evolutionary computation, a class of computer software systems that employ software techniques derived from Darwinian evolution to find an optimized solution within a large search space, the most popular of which are genetic algorithms (GA) and genetic programming (GP).

Standard evolutionary systems use a computationally intensive, automated programming methodology inspired by biological evolution to find computer programs that best perform a user-defined task. They do this by repeatedly testing the current population of programs by a given 'fitness function' test and then marrying (i.e. applying genetic crossover) those programs that do best, thereby passing on those best genes to new offspring. The 'genotype' of an individual program are those genes or computer codes that make up a recipe. The product of this recipe is the 'phenotype' which is the individual organism or in GP's case the final computer program.

Creative evolutionary systems can be used to evolve aesthetically pleasing structures in art, music and design. Within computer visual art, these systems are often referred to as evolutionary art systems.

According to Bentley from his seminal book on the subject [1], a creative evolutionary system is designed to 1) aid our own creative process and, 2) generate results to problems that traditionally required creative people to find the solutions. Bentley goes on to state that in achieving these goals, a creative evolutionary system may also appear to act 'creatively' - although this is still a source of debate. Unlike general evolutionary computation systems, creative evolutionary systems have been criticized because most of these systems use the presence of a human (often playing the role of the creative decision maker or fitness function) to guide the direction of the evolutionary search. Our portrait painter system specifically uses an automatic fitness function (albeit specific to a portrait painting where a portrait sitter resemblance is encouraged), thereby attempting to work through the human fitness function dilemma and directly explore how computer algorithms can be creative.

Our work is based on Ashmore and Miller's work [2], which uses a relatively new form of Genetic Programming [3] called Cartesian Genetic Programming (CGP) first developed by Miller [4]. CGP uses typical GP Darwinian evolutionary techniques (crossover, mutation, and survival), but has several features that allow the GP system to favour creative solutions over optimized solutions including: accommodating for genetic drift where different genotypes (i.e. genetic codes or recipes) map to the same phenotype (i.e. the working computer program or individual), visual mapping modules and knowledge of a painterly colour space. Portrait painting was chosen for this project as it limits the creative space of all art paintings, weighs towards resemblance, and has a known portrait sitter/painter relationship well suited to explore computer creativity. This work with its specific goal of evolving portrait painter programs to create a portrait 'sparked' by the famous portrait of Darwin (the resemblance fitness function), speaks to the evolutionary processes as well as creativity, as seen by our early results in which the evolving programs use recurring, emergent and merged creative strategies to become good abstract portraitists. This technique has uses in computer creativity, art making as well as educational applications for hands-on understand of evolutionary and creative processes.

RELATED WORK

Speaking broadly, creative evolutionary systems that combine with the aesthetic decisions of a human to judge fitness started well before computers. Standard historical selective breeding practises, where a human selects the parents for each generation from a given evolved set of choices, is the basis for centuries of 'creatively' modified trees, roses, corn, dogs, cats, cows and so on. Current evolutionary art systems borrow from this time tested approach. It was evolutionary biologist Richard Dawkins who first showed with his “Biomorphs” program that accompanied his 1986 book “The Blind Watchmaker” that a computer can be combined with the aesthetic preference of a user to generate interesting results. Dawkins work inspired artists such as William Latham and Stephen Todd as well as Karl Sims.

Karl Sims' work went on to inspire many of the modern evolutionary artists today. In his 2D work [5], Sims used a very rich instruction set, containing image processing functions as well as mathematical functions based on LISP expression trees. As with most evolutionary art systems to follow, Sims system evolved a number of images (16 in his case) and allowed the viewers to pick their favourites, thereby allowing the most 'aesthetically pleasing' images to survive and mutate to the next generation. Other well known artists used similar techniques: Steve Rooke [6], also working in LISP, is very well known for his artwork which added evolvable fractals to the function set and Penousal Machado [7], a researcher at the Artificial Intelligence Laboratory at University of Coimbra, in contrast to Sims' complex function set, used a very simple function set which is believed to open up the possible search space.

These systems, as with most creative evolutionary systems, use a human (often the artist or viewer under interactive control) to make the aesthetic decisions after each population. In contrast, using an automatic fitness function where the computer judges aesthetic or creativity fitness is a more open-ended research problem. Recent work by Bentley [1] in the design space, Thompson as well as Miller in the electronic circuit area and the father of GP, John Koza, in building a creative invention machine, have begun to explore automatic fitness strategies. In art related areas, the problem is still quite hard. How do you write a logically oriented fitness function that has a sense of the aesthetic or the creative? Given this, systems that use creative fitness functions in art are still quite naïve. Ashmore and Miller [2] have attempted to use an automatic fitness function with Cartesian GP that evolve imagery for greater image complexity or circular objects in the image (using a Hough Transform operation) to start a population, then they allow the user to take over. They also attempted an automatic function for evolving towards a source image. We have based our system upon their work - expanding their 'evolving towards a source image' with a more sophisticated similarity function as well as revising their system for a portrait painter process.

CREATIVE EVOLUTIONARY COMPUTATION ISSUES

Evolutionary computation uses the techniques of natural or Darwinian evolution, to find solutions to a given problem from a very large search space. Typically finding the best solution in a search space is called 'optimization' - as in finding the most optimal solution. Here is a simplified stepwise approach to show the evolutionary algorithm process:

PreStep 1: Create a set of functions (the tool set all individuals are created from).
PreStep 2: Create a fitness function (a test which scores fitness to a given result).
PreStep 3: Initialize a population of individuals with random functions.
- Step 1: Score all individuals in the current population against the fitness test.
- Step 2: Those with the best scores are mated together with crossover techniques.
- Step 3: Some low level random mutations are performed on some mated individuals.
- Step 4: These new offspring fill up a new current population.
- Step 5: Return to Step 1 until an individual scores an acceptably high fitness score.

Note that evolutionary algorithms do not explicitly 'program in' information about the problem; the systems are blind in this respect. Instead, the systems maintain populations of solutions, allow better solutions to mate and have offspring, thereby passing on those successful characteristics with some random variation from successive parent to offspring, and so on. This procedure causes evolution to occur. It is the external fitness function and to some extent the initial function set that has some open knowledge of the solution direction, not the individuals or the evolutionary process.

Creative systems need to favour exploration over optimization, finding innovative or novel solutions over a preconceived notion of a specific optimal solution. The best creative evolutionary systems only provide tools to build new solutions, allowing the evolutionary process to discover novelty and innovation by itself. One way creative evolutionary systems differ from more traditional evolutionary systems, is the relaxation or removal of constraints. While strong constraining or parameterization of a function set allow evolutionary systems to produce optimized and fast results, this simply limits the available search space and hence the ability for the system to come up with solutions that are 'outside the box'. Instead of a parameterization of the solution, creative systems use a set of low-level components. Solutions are then constructed by using these components, allowing greater exploration. With this exploration comes the hazards of a very large search space (creating longer runs) and worse, the possibility of being caught in local maximas where the evolutionary systems no longer progress. We will discuss later, using Cartesian GP techniques, how we deal with the 'local maxima' problem.

Another way to push evolutionary systems towards being creative is to allow a creative person - a human - to play the role of the fitness function. Most creative evolutionary systems use this approach and, as mentioned previously, are somewhat criticized by the evolutionary computation community for it. While this computer / human collaboration has created some very successful results and is a valid technique, it does have several disadvantages, including:

- Speed: the system stops at every run and waits for a human to judge the results;
- Consistency: humans tend to judge for the situation at the moment and often can not see the big picture; and
- Coverage: it is impossible to give a human all the possibilities to judge from, so most systems of this type limit the population (i.e. 8-16 individuals).

Another disadvantage, is a human 'creative decision maker' approach puts off the issue of computer creativity. While computer creativity is an open ended problem and most early solutions will be quite naïve, working on computer creativity begins to give insights into how human creativity works, which has been one motivation for our research work.

CARTESIAN GENETIC PROGRAMMING

Cartesian genetic programming (CGP) is a form of genetic programming where the program is represented by a directed graph of indexed nodes [4]. Each node has a number of inputs and a function that gives an output based on the inputs. The genotype is a list of integers that determine the connectivity and functionality of the nodes, which can be mutated and crossed over to create new directed graphs.

The genotype may contain nodes that are not connected to the output nodes so are not expressed in the phenotype, this is called node redundancy. As well as node redundancy there is also functional redundancy and input redundancy. This redundancy provides CGP with greater neutrality [4.8] when compared with standard GP. Neutrality is the presence of a genotype/phenotype mapping which allows different genotypes (recipes) to map to the same phenotype (individuals). When a plateau or local maxima is reached genetic drift may occur across the plateau. Genetic drift is the changing of unexpressed genes, or nodes, in the genotype that may lead to a later improvement in fitness when they are expressed. If genetic drift occurs then a later offspring may have the ability to create a fitter individual, enabling escape from local maxima. See Figure 1.

Using CGP with genetic drift has a double benefit for our portrait painter program system. Evolutionary systems like ours are most creative when they use a simple function set of low-level components (as opposed to a complex parameterized set). This has the disadvantage of being more susceptible to being caught in local maximas because of the large search space. Genetic drift allows our system the potential to escape from local maxima but in addition, drift also allows our abstract portrait system to be more creative or novel in its evolutionary search as it is forced to drift to other plateaus. In our implementation, drift is programmed in the follow way: if the best individual of a new population is the same as the last population for more than three iterations, other genotypes that map to this same phenotype are chosen over the current non-progressing genotype.

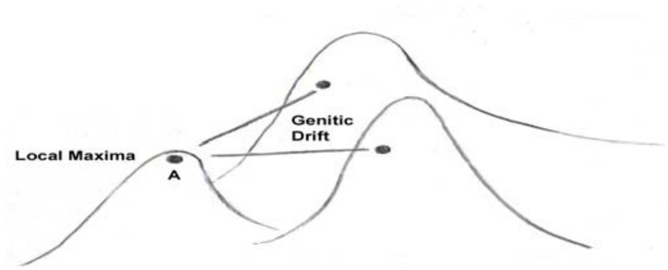


Figure 1. CGP allows different genotypes to map to the same phenotype, so if 'A' is caught in a local maxima, 'genetic drift' can map to other genes outside that maxima.

THE FUNCTION SET

Ashmore and Miller's function set, which we use for our system uses CGP graphs that have two inputs: the x and y coordinates of a pixel in the image, and three outputs: the three colour channels ([red, green and blue] or [hue, saturation and value]) for that pixel. The program represented by the chromosome maps each coordinate, based on its value, to a specific colour. Hence, changing the functions and connectivity of the nodes will change the colour values of each pixel, and so change the image. The genotype is stored as an integer array.

The function set has 15 functions, labelled 0 to 14 below, where input1 and input2 are the x and y position of the output image, respectively. Some functions have a parameter variable that can be affected by random mutation. Functions are specifically low level in nature which aids in a large 'creative' search space. For the portrait painter system we specially used hue, saturation and value (HSV) colour space because it is more painterly than the typical computer RGB space. HSV space allows us to create a fitness function that can begin to favour more painterly rules (i.e. moving through tonal or value space). The exact function set is (written in Java here):

```
0: input1 | input2;
1: parameter & input1;
2: (input1 / (1.0 + input2 + parameter));
3: (input1 * input2) % 255;
4: (input1 + input2) % 255;
5: if(input1>input2) input1 - input2; else input2 - input1;
6: 255-input1;
7: abs(cos(input1)*255);
8: abs(tan(((input1%45)*p)/180.0)*255));
9: abs(tan(input1)*255)%255);
10: sqrt( (input1-parameter)2 + (input2-parameter) 2); (thresholded at 255)
11: input1%(parameter+1)+(255-parameter);
12: (input1 + input2)/2;
13: if (input1>input2) 255*((input2+1)/(input1+1));else 255*((input1+1)/(input2+1));
14: abs(sqrt(input12-parameter2+ input22-parameter2)%255);
```

Note how most functions above simply use the x,y position (input1,input2) of the final image to contributed to what the colour of that position will be. This allows correlated painterly effects as you move through the image. Functions 0 through 6 use simple logical or arithmetic manipulations of the positions (low level functions create a larger 'creative' search space), whereas 7 through 14 use trigonometric or logical functions that are more related to geometric shapes and colour graduations. Many functions are clipped to 255 since the computer colour space is between the integers 0 to 255 per colour component.

GOALS AND IMPLEMENTATION

The goals for our first prototype project of this process, where we use Darwinian evolution techniques to evolve portrait painter programs to paint portraits of Darwin, are many and interdisciplinary. On the artistic level, this prototype project was a conceptual art piece that creates and evolves a related family of abstract portrait painters. Each portrait is created via one evolved computer program. These programs are created and evolve by Darwinian evolutionary techniques. The environment in which they prosper and have offspring is a resemblance fitness function test to the portrait painting of Darwin by John Collier. The 'most fit' (i.e., individuals that resemble the portrait better than their neighbours) of a population are 'married' together to create 'more successful' offspring. The genes (or function set) are specifically low-level and component based which gives the system a stronger ability to produce innovative and novel results.

Can you bring the ghost (creativity) out of the machine using the ghost of Darwin -- his namesake techniques and portrait? Our first pass prototype ran continuously on one high-end PC for 50 days. The portraits results can be viewed in brief, in the results section and at http://www.dipaola.org/evolve/darwin. We have culled together our favourite portraits in terms of aesthetic value as well as examples that help to show the process. As discussed, with standard evolutionary computation, the end result or 'optimization' is what is important. Our process is more about the journey or 'creativity', while the overall population improves at resembling Darwin's portrait, that is less the point; the Darwin goal is simply the creative spark. Since the genes of each portrait can be saved, it is possible to re-combine (marry) and re-evolve any of the art works in new variants as seen in Figure 2. While this effort works as a new media art piece, we believe the work speaks to the evolutionary processes as well as creativity, and therefore has uses in education. Our system can be reworked to allow science and art museum visitors to better understand the evolutionary process as well as the creative process through hands-on interactive control. With different functions and fitness test, different strategies can be explored; for instance, inviting Aquarium visitors to create and evolve ocean based virtual life forms with different locomotion or survival strategies.

Our system was adapted from Ashmore and Miller's evolutionary art system [2]. We rewrote significant parts of their successful Java based art producing system, with the specific goals of creating an automatic fitness function based in painterly resemblance and an art process that better mimics the portrait painter process.

The automatic fitness function uses a 'portrait to sitter' resemblance. That is, the closer an evolved portrait resembles the source sitter image, the better it performs on the fitness test. Resemblance or image similarity can be measured in many ways. Our fitness function must give a specific and correlated score at any resolution level to be effective; judging painterly similarity of any portrait image in deciding which individuals are more fit even in very early runs. This is why creative fitness functions are still very difficult. They must judge even arbitrary results with full accuracy.

Since the advent of photography, portrait painting has not just been about accurately reproducing the sitter but also about using modern painterly goals (i.e., form, colour, light, feeling …) to achieve a creative representation of the sitter. We have created a fitness function that mainly rewards accurate representation (similarity to source picture at 80%), but also rewards for painterliness (the rules of good painting at 20%). Our goal is to keep researching creative portrait painting techniques and build a more sophisticated model, both to the fitness function and to the function set (the tools).

The system differs from most creative evolutionary systems in that the individual with the highest fitness score is not our end goal. Portrait programs in the beginning of the run will look less like the sitter but from an aesthetic point of view might be highly desirable. As the fitness score increases, portraits will look more like the sitter. While this gives us a somewhat known spread from very primitive (abstract) all the way through realistic portraits, any of these portraits might be deemed a good abstract portrait. So in effect our system has two ongoing progressing processes: 1) those 'most fit' portraits that pass on their portrait resemblance strategies, making for more and more realistic portraits, and 2) 'strange uncles': related to the 'most fit', but while not great at sitter similarity, portraits that are artistically compelling. Figure 3 shows only those images that were most fit for their population in order, whereas Figure 4 (and the website) show both the fit images and those images we picked out as artistically compelling. The process than is both automatic in that it does not need to stop for human decision making, but uses a human as critic and editor, able to save interesting portraits during the process. The goal is not to remake the portrait of Darwin, but to explore a family tree of related portraits which all share and pass-on painting strategies that are created through a blind evolutionary process. One of our goals is, as we better understand our iterative research process, to continually use a less painterly source image, substituting a realistic face which is more truthful to a portrait sitter, and begin to put painterliness and creative decision making more within the fitness function and function set, thereby slowly moving 'creativity' from mimicry (be like the Darwin painting) to knowledge (evolve painterliness based decision making).

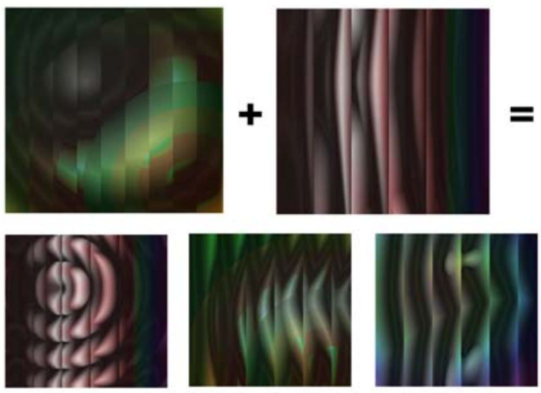


Figure 2. Two portrait programs taking randomly from the Darwin run that were mated together to produce new paintings. Note how strategies have merged in the offspring.

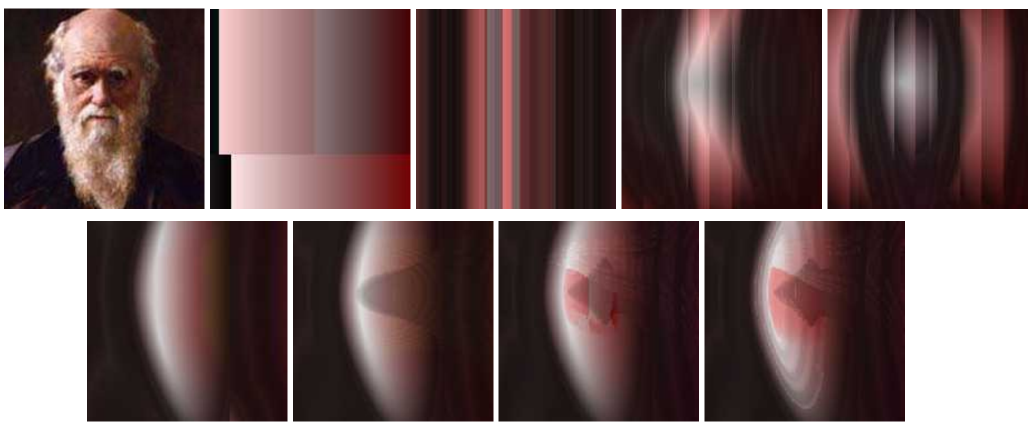
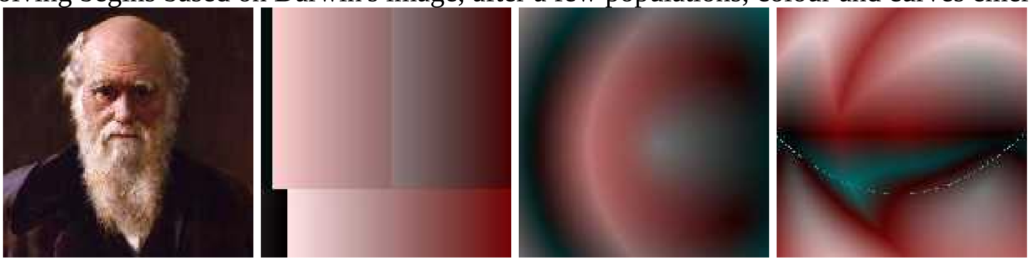


Figure 3. Source Darwin portrait (the fitness function image) followed by an evolved progression of portraits of best 'resemblance' fitness to the Darwin source.

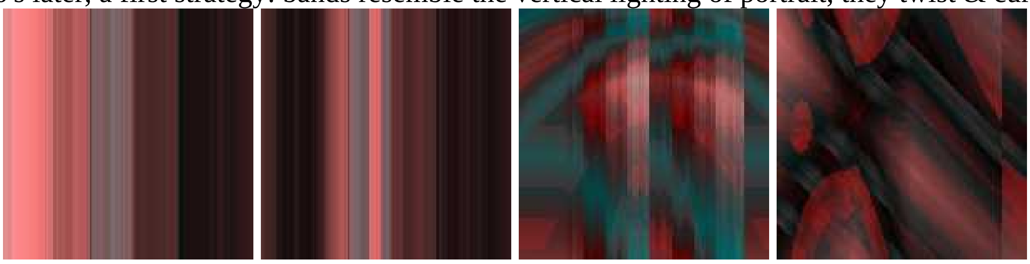
RESULTS

The images below (Figure 4) with their captions, as well as the results website http://www.dipaola.org/evolve/darwin, show selected portraits in order, starting with the first population and moving in chronological order. These represent a larger collection and show both those best at resemblance of the peers, as well as those that are artistically compelling from an abstract portrait perspective. Even these first pass, early results show how the portrait programs evolve recurring, emergent and merged creative strategies to become good abstract portraitists. Our goal is to continually refine the creative or painterly portions of the automatic fitness function in future iterations from lessons learned from past runs.

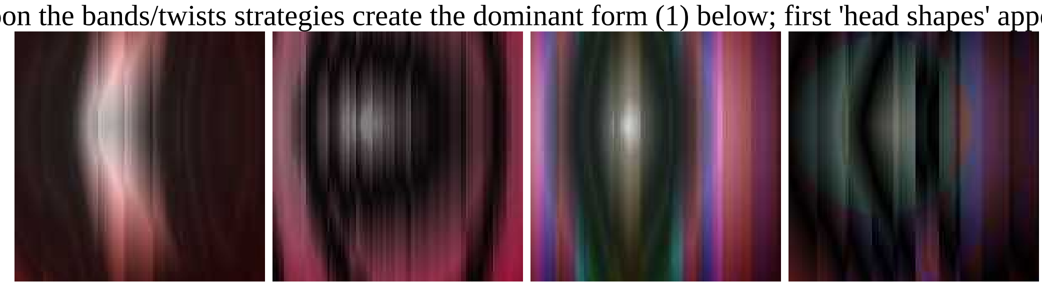
Evolving begins based on Darwin's image, after a few populations, colour and curves emerge.



100's later, a first strategy: bands resemble the vertical lighting of portrait, they twist & curve.

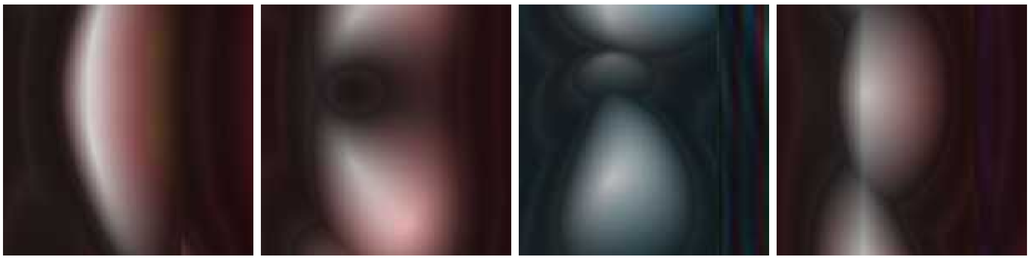


Soon the bands/twists strategies create the dominant form (1) below; first 'head shapes' appear.

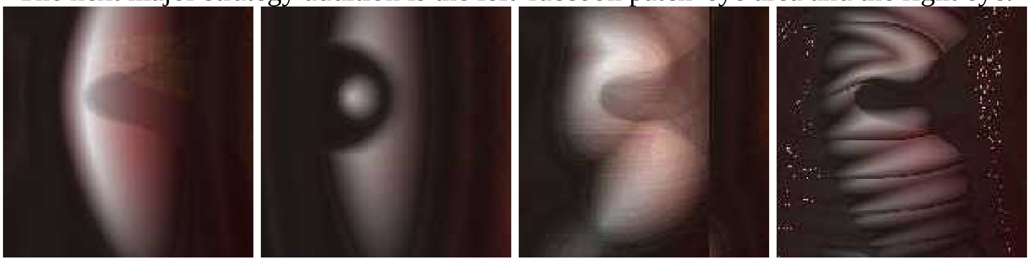


As the early evolution progress slows, genetic drift brings in a colorful and novel phase.

After a while, this ramped dominant strategy takes over (1 below) heralding in the blobby age.



The next major strategy addition is the left 'raccoon patch' eye area and the right eye.



Still evolving, a more painterly age begins, combining the head shape with painterly surface

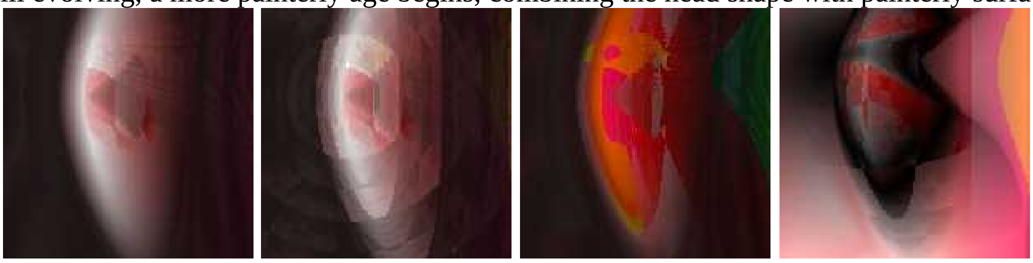


Figure 4. Portraits in chronological order, selected as examples of the process.

FUTURE DIRECTIONS

To better approximate a human portraitist's technique, we are interested in redesigning all functions in the function set to be reactions to the colour and position of the sitter image. This way any decision on a paint stroke output is a direction reaction to the input recognition (what the artist sees in the sitter scene). This would mean, once a pleasing portrait image/individual is created, that portrait program could use its same painterly strategies on any new sitter image, thereby creating a true portrait painter. Imagine then that a successful portraitist program could have 'one-man' shows and take commissions, allowing its human creator to play a background role as its talent agent where they could eventually breed it with other successful portraitist programs similar to racing horses. This 'matching output stroke to input analysis' technique with other modifications should allow for another goal: to have resolution independent portraits, allowing small portrait sizes for speed during the evolving process, but larger sizes that reveal additional painterly and surface details for final artwork.

We are also researching a faster, more painterly, and more targeted resemblance test. Since tonal values are more important than exact colour similarity in painting, Howe [9] has suggested creating filters that compute the presence of specific tones at each pixel of the sitter image, than compute the 'moments' of these tone distributions, giving a short vector that could be used to compute similarity with other images. This would allow us to work in tonal space, then adding colour to the tones after the similarity test based on formal rules of creative painting (i.e. warm colours next to cold colours), better simulating a human painter's creative process.

Building in painterly fuzzy logic rules (an internal artistic painter model which use the formal rules of creative painting) as well as a simple knowledge model of faces is also being investigated.

As discussed early, we believe there are educational uses of the system for 'hands-on" interactive experimentation to learn how evolution works. While this is true for our portrait painting prototype where art museum patrons might explore artistic creativity in an interactive kiosk, but it can be extended into other areas. For instance we have begun working with the Vancouver Aquarium in Canada, planning out an interactive learning system prototype that evolves ocean creature locomotion and survival strategies, allowing children to understand and explore, in a playful and open-ended way, how ocean animals evolve differently.

Our main research direction however is to keep exploring computer creativity as a technique to better understand how human creativity works.

ACKNOWLEDGEMENTS

We thank A. Ashmore and J. Miller [2] for their work, and correspondence.

References

[1] BENTLEY, P and Corne, D (eds.): Creative Evolutionary Systems, *Morgan Kaufmann, San Francisco (2002).*

[2] ASHMORE, A and Miller, J: Evolutionary Art with Cartesian Genetic Programming, *http://www.emoware.org/evolutionary_art.asp*

[3] KOZA, J: Genetic Programming, *MIT Press, London, (1993).*

[4] MILLER, J and Thomson, P: Cartesian Genetic Programming, *Proceedings of the 3rd European Conference on Genetic Programming, Edinburgh, (2000) 121-132.*

[5] SIMS, K: Artificial Evolution for Computer Graphics. *Computer Graphics, Vol. 25, (1991) 319-328.*

[6] ROOKE, S: Eons of Genetically Evolved Algorithmic Images. In: Bentley P. J. and Corne D. (eds.): Creative Evolutionary systems, *Morgan Kaufmann, (2002).*

[7] MACHADO, P and Cardoso, A: NEvAr - The Assessment of an Evolutionary Art Tool. In: Wiggins, G. (Ed.). *Proceedings of the AISB00 Symposium on Creative & Cultural Aspects and Applications of AI & Cognitive Science, UK, 2000.*

[8] YU, T and Miller, J: Neutrality and the evolvability of Boolean function landscape. *Proceedings of the Fourth European Conference on Genetic Programming, Springer-Verlag, Berlin (2001) 204-217.*

[9] HOWE, N: Percentile Blobs for Image Similarity, *IEEE Workshop on Content-Based Access of Image and Video Databases, 1998.*