Evolutionary Image Synthesis Using a Model of Aesthetics

Brian J. Ross, William Ralph, and Hai Zong

Abstract— **The automatic synthesis of aesthetically pleasing images is investigated. Genetic programming with multi-objective fitness evaluation is used to evolve procedural texture formulae. With multi-objective fitness testing, candidate textures are evaluated according to multiple criteria. Each criteria designates a dimension of a multi-dimensional fitness space. The main feature test uses Ralph's model of aesthetics. This aesthetic model is based on empirical analyses of fine art, in which analyzed art work exhibits bell curve distributions of color gradients. Subjectively speaking, this bell-curve gradient measurement tends to favor images that have harmonious and balanced visual characteristics. Another feature test is color histogram scoring. This test permits some control of the color composition, by matching a candidate texture's color composition with the color histogram of a target image. This target image may be a digital image of another artwork. We found that the use of the bell curve model often resulted in images that were harmonious and easy-on-the-eyes. Without the use of the model, generated images were often too chaotic or boring. Although our approach does not guarantee aesthetically pleasing results, it does increase the likelihood that generated textures are visually interesting.**

I. Introduction

The automatic synthesis of aesthetically pleasing images is investigated. This research contributes to the body of work using evolutionary computation to generate procedural textures. We use genetic programming to synthesize procedural texture formulae [1]. Principles of Darwinian evolution are applied to a dynamic population of textures. Those that generate images with desired visual features are likely to reproduce, and have offspring with even more desireable features. During the evolution process, the rendered output of texture expressions are evaluated by two independent feature analyses. The first analysis uses a novel model of aesthetics by Ralph [2]. This model is derived from an empirical study of fine art, in which it was discovered that many examples of art work exhibit a normal distribution of colour gradients. The other feature analysis is colour histogram scoring, which compares an image's colour distribution histogram to that of a target image. By matching the texture's colour distribution with another image, we are borrowing another artist's colour sensibilities. The combined result of these tests are images that conform to both the bell curve aesthetic and desired colour composition.

The paper is organized as follows. Related work in evolutionary texture synthesis is reviewed in Section II. Section III describes Ralph's aesthetic model, which is used as one of the feature tests by the GP system. System design and experimental details are described in Section IV. Section V presents some example results. Concluding remarks are given in Section VI.

Brian Ross is with the Department of Computer Science, Brock University, St. Catharines, ON L2S 3A1, Canada (phone: 905-688-5550 ext 4284; fax: 905-688-3255; email: bross@brocku.ca).

Hai Zong is a software developer at Teilhard Technologies, St. Catharines, ON, Canada (email: hzbrock2001@yahoo.com).

William Ralph is with the Department of Mathematics, Brock University, St. Catharines, ON L2S 3A1, Canada (email: bralph@brocku.ca).

II. Related Work

Evolutionary texture and image synthesis is well established [3][4]. Most systems are supervised, in which the user interactively views and rates texture images from the population, and possibly controls the rate of mutation [5][6][7][8][9]. Interactive evolution is less suitable for evolving preconceived styles of textures, than it is as a creative tool for discovering new textures. Moreover, it can quickly exhaust the user, who must manually view and evaluate every generated texture.

Automatic texture evolution removes the user from the loop [10][11][12]. Image analysis functions evaluate rudimentary image features such as colour, luminosity, and shape. These scores are then matched with those of a target texture. Another approach is in [13], which evolves procedural textures for 3D models. Instead of image analysis, this system uses training sets of example texture points.

Automated aesthetic texture evolution has not been widely studied. A pioneering work is [14], which uses an artificial neural network (ANN) as an image evaluation function within a GA. Before evolution, the ANN is trained on sets of example images that have been deemed to be aesthetically appealing. Their system does generate textures according to the evaluation by the ANN. It is unclear whether their ANN has learned any relevant aesthetic principles or useful patterns, especially given the vast quantity of training data used.

More recently, genetic programming using a fitness function encoding a model of aesthetics has been used in the NEvAr system [15]. That model posits that images are aesthetically pleasing if they are both visually complex, as well as easy for the brain and visual system to perceive and interpret. Two mathematical measurements are used: (i) image complexity is measured by the JPEG compression ratio; (ii) visual self-similarity is indicated by the fractal compression ratio. Their aesthetic model was applied to the TDA test (Test of Drawing Appreciation), which is a standardized psychological test used to evaluate art appreciation [16]. The results were impressive, as the model scored better than typical art students.

Recently, an aesthetic distance metric was proposed as a means for measuring the information proximity between candidate images and a library of images preclassified to be

aesthetically pleasing [17]. Although the metric was successfully tested against user preferences during interactive texture evolution, it has not been tested extensively in automatic texture evolution.

Measurable aspects of artificial and natural phenomena often exhibit particular kinds of mathematical distributions [18]. For example, the well known *1/f* or pink noise distribution is widely found in diverse areas such as physics [19], natural images [20], and human cognition [21]. One famous example examining 1/f noise in the arts is [22]. In the music examples analyzed, they show that the differences in successive pitches in notes (pitch gradients) exhibit 1/f distributions. Furthermore, stochastically-generated music based on 1/f noise generators is more aesthetically pleasing than that generated by pure random or white noise generators.

III. A Mathematical Model of Aesthetics

A mathematical model characterizing an aspect of fine art has been proposed by Ralph [2]. The following gives a simplified overview of the theory. After analyzing hundreds of examples of fine art, it was found that many works consistently exhibit functions over colour gradients that conform to a normal or bell curve distribution. This is seen with many artists, such has Cezanne and Seurat, who create "painterly" images. On the other hand, the visual responses to photographs and graphic design work usually do not have normal distributions. This normal distribution of gradient response is claimed to be difficult to realize, since it is essentially characterizing the global distribution of gradient throughout an entire image: local changes affect the distribution found throughout the entire painting. It is hypothesized that this bell curve distribution has been an implicit aesthetic ideal of many painters throughout history.

The bell curve model posits that a viewer's response to an image is largely determined by his or her psychoneurological reaction to visual stimuli. When viewing an artwork, a viewer's visual system is stimulated by the details of the image. The bell curve model suggests that a viewer is most attracted to changes in an image, for example, the edges between different colours. The areas with constant colours are of less interest. Furthermore, larger changes are more noticeable than smaller ones. Since it is known that our nervous system tends to have a logarithmic reaction to stimuli, this model likewise treats measurements logarithmically.

An image's bell curve gradient is computed in three steps.

Step 1: An image's colour gradient is found. This is done by computing the following for each pixel (i,j) of an RGB image (ignoring the extrema row and column of the image buffer):

$$|\nabla r_{i,j}|^2 = \frac{(r_{i,j} - r_{i+1,j+1})^2 + (r_{i+1,j} - r_{i,j+1})^2}{d^2}$$

where $r_{i,j}$ is the red value at pixel (i,j). Similar computations are done for the green and blue channels. The d value is a scaling factor that is used to scale the result for images having different dimensions. We take it to be 0.1% of half the diagonal length, scaled for typical pixel densities on CRT monitors. The overall gradient or *stimulus S* is then:

$$S_{i,j} = \sqrt{|\nabla r_{i,j}|^2 + |\nabla g_{i,j}|^2 + |\nabla b_{i,j}|^2}$$

for the separate RGB channel gradients. Finally, the *response R* is computed as:

$$R_{i,j} = log(S_{i,j}/S_0)$$

where S_0 is the threshold of detection, which is taken to be 2. If $S_{i,j} = 0$ (no change in colour at a pixel), it is ignored.

Step 2: The distribution of the response values for an image is determined. The calculation of the distribution is based on the hypothesis that the probability that a viewer pays attention to a detail of an image is proportional to the magnitude of the stimulus that resides at that detail. Hence we use a weighted mean and standard deviation, with $R_{i,j}$ being the weight value for each response. The normal distribution of R is then estimated using a weighted normal distribution, defined by a mean (μ) and standard deviation (σ^2):

$$\mu = \frac{\sum_{i,j}(R_{i,j})^2}{\sum_{i,j} R_{i,j}} \qquad \sigma^2 = \frac{\sum_{i,j} R_{i,j}(R_{i,j}-\mu)^2}{\sum_{i,j} R_{i,j}}$$

Once μ and σ^2 are found for an image, the actual distribution of all $R_{i,j}$ for all pixels in the image is tabulated. Using a bin width of $\sigma/100$, a histogram is calculated, where each $R_{i,j}$ updates its corresponding bin using a weight of $R_{i,j}$.

Step 3: The closeness of fit between the response actual distribution and the hypothesized bell distribution is calculated. This is called the *deviation from normality* or DFN, and it is calculated as:

$$DFN = 1000 \sum p_i log(\frac{p_i}{q_i})$$

where p_i is the observed probability in the i^{th} bin of the histogram, and q_i is the expected probability assuming a normal distribution with the computed mean and standard deviation above. When $q_i = 0$, that bin is ignored. A DFN value of 0 means that a perfect normal distribution exists, while higher DFN values indicate poorer fits to the normal distribution.

There are a number of practical advantages of the bell curve model. The DFN score is easily incorporated as a fitness score within the genetic algorithm. The bell curve model also permits a means for characterizing general artistic styles. As discussed in [2], different styles of art are often characterized by their bell curve fit, as well as the associated mean and standard deviation of the distribution. Many paintings with good DFN's tend to have bell curves with a mean around 3.0 and a standard deviation of 0.75. Photographs have poor bell distributions, with means of around 4.2 and standard deviations of 1.2 or more. Graphic designs have even higher values. These values can be used by the genetic programming system as general stylistic target areas for evolved images.

Float (f):	
Terminals:	x, y, ephem
Basic math:	plus(f,f), minus(f,f), diff(f,f), mult(f,f), div(f,f), neg(f)
Other math:	sin(f), cos(f), mod(f,f), log(f), pow(f,f)
Relational:	min(f,f), max(f,f), if-then-else(f,f,f)
Noise:	noise(f,f), turb(f,f), turbflow(f,f,f), cloud(f,f,f,f)
Misc:	tilerad(f,f), lum(v), chn(v), ichn(f,v)
RGB vector (v):	
Terminals:	colgrad, ephem
Noise:	marble(f,f,v)
Transform:	warprel(f,f,v), warpabs(f,f,v), kaleid(v), vtile(f,f,v)
Other:	rgb(f,f,f), if-then-else(f,v,v), forv(f)

TABLE I
TEXTURE LANGUAGE

IV. EXPERIMENT

A. Texture language

The texture language used is similar to one used by Sims [5]. A summary of the language is given in Table I. The genetic programming system uses strong typing, and so all language components respect the data type conventions specified in the table. Two data types are used – float (f), and RGB vector (v), which is a tuple of 3 float values that is interpreted as an RGB colour.

The float primitives compute floating point values. The terminals include *x* and *y*, which are the current pixel coordinates whose texture colour is being computed. The terminal *ephem* denotes *ephemeral random constants* [1]. These constants are initialized with a random number generator, and retain the initialized value throughout the remainder of their existence within the run. Their range is between -1.0 and 1.0. Most of the math and relational operators are straightforward. The *if-then-else(f,g,h)* expression evaluates expression *f*. If $f > 0$, then the value of *g* is returned. Otherwise the value of *h* is returned. The noise operators use a *pink noise* or Perlin-like noise generator. The basic generator is *noise(f,g)* [23]. It takes 2 explicit arguments, the frequency f and a random seed g, and 2 implicit arguments, the current x and y pixel coordinates. *Turb*, *Turbflow*, and *Cloud* use turbulence and cloud modeling formulae [23][24]. *Tilerad* generates a tiling effect.

The vector functions return RGB vectors as evaluated results. Vector terminals include ephemeral constants, as well as *colgrad*, which computes an RGB gradient based on the current pixel coordinates. The other vector primitives perform a variety of texture and tiling effects, vector construction and deconstruction, and vector iteration.

B. Feature evaluation

The relative worth of a texture image will be measured by its performance on a suite of feature tests. One feature test is the bell curve analysis from Section III. The deviation from normality (DFN) determines how well an image's gradient fits to a normal distribution, and hence shares the gradient characteristics found in many works of art. The gradient analysis also computes mean and standard deviation scores. The mean denotes the range of gradient seen in an image. The standard deviation represents the changes in gradients seen, ranging from gradual shifts to sharp jumps in colour. We find it useful to include one or both of these measurements with the DFN when analyzing images.

The other feature test, CHISTQ, is a colour analysis that compares the colour distribution of a texture image with that of a target image. It uses quadratic histogram fitting for colours, which is a test often used in query by image content systems such as VisualSEEk [25]. The image is first quantized into 1000 colours. Then a histogram of quantized colour frequencies is tabulated. The histograms for two images are compared with one another, and an overall distance between them is calculated. All the histogram entries in both images are compared exhaustively with one another, to determine how closely the colours distributions match. This test is fairly relaxed with respect to colour fitting. It is also position independent, meaning that colours do not have to coincide with respect to their placement on an image.

C. Multi-objective fitness scoring

A multi-objective problem is characterized by having two or more fitness criteria [26]. Multi-objective search strategies consider each feature test as an independent dimension in the search space. This contrasts to approaches that merge scores together, perhaps by a weighted sum. Weights are usually *ad hoc*, introduce undo bias into the search, and can be detrimental to most nontrivial problems.

We interpret the evaluation of textures using the multi-objective approach from [12]. When textures are added to a new population, the DFN and CHISTQ tests are performed on the generated texture from each formulae, and the fitness scores (including mean and standard deviation scores) are saved. These scores are then used to determine a *Pareto ranking* of all the individuals in the population. Pareto ranks are based on the idea of *domination*. One individual dominates another if it is at least as good in all the scores, and better in at least one. Using the notation $u < v$ to mean score u is more optimal than v, then *u dominates v* if:

$$\forall i \in (1, ..., k) : u_i \leq v_i \wedge \exists i \in (1, ..., k) : u_i < v_i$$

Individuals having rank 1 are undominated, and are the current best solutions in the population. Those of rank k > 1 are dominated by all the individuals of ranks $< k$. All the individuals in a rank are incomparable with one another. At the end of a run, all those with rank 1 are considered as valid solutions to the problem.

After the Pareto ranks are determined, the individuals in each rank are evaluated with respect to their diversity. Textures in a Pareto rank are considered superior if they are more diverse with respect to their location in the multi-dimensional search space, as indicated by the fitness scores in each feature test. The idea is that a texture with unique characteristics, likely has correspondingly different feature vector values. Likewise, textures that are duplicates or minor variations of

each other probably have very similar or identical feature test scores.

An indicator of diversity in feature space is the nearest-neighbour distance between feature vectors. Because feature tests differ widely between each other in terms of valuations, the raw scores in the vectors are not used to determine distances. Rather, these distance scores are abstracted into ranks, and the average distance rank is computed instead. This approach will give all feature scores equal weight when determining distance. The following steps are performed to determine diversity for all the textures in each Pareto rank set:

1) For each feature test dimension, the nearest-neighbour distance is determined.
2) All the distances are ranked, with the largest distance assigned the lowest rank (1). Each individual ends up with a vector of distance ranks, with one distance rank per feature test.
3) The average distance rank is deterined for each distance rank vectors. This is used as the "diversity score", where lower averages are preferred.

Once these steps are computed, the textures within each Pareto rank set are re-assigned fitness scores, such that more diverse individuals in the rank have better scores compared to others in the rank set. The desired outcome is a diversity of generated images. Note that diversity is only applied within the Pareto sets, and the Pareto ranks themselves are maintained. In other words, diversity does not affect the overall Pareto ranking between Pareto rank sets.

D. Other experimental parameters

Parameter	*Value*
Population size	1000
Generations	50
Runs/experiment	5
Initialization	ramped half&half
Initial ramped tree depth	2 to 6
Max. tree size (nodes, depth)	200, 20
Crossover rate	0.9
Mutation rate	0.1
Selection scheme	tournament (size 3)

TABLE II
GENETIC PROGRAMMING PARAMETERS (TYPICAL)

Table II lists other typical parameters used in the experiments. Most are standard within the genetic programming literature [1]. Five separate runs using new random number seeds were done per experiment, using a population of 1000 texture expressions. A maximum of 50 generations were used. The initial population is generated using the ramped half-and-half tree generation algorithm, which creates a population of random trees having an assortment of sizes and shapes. Initial trees have depths between 2 to 6 levels. During reproduction, trees can have a maximum of 200 nodes, and a maximum depth of 20. Crossover is used 90% of the time, and mutation is selected the remaining 10%. Fitness-based selection is performed via *tournament selection*. Here, 3 random formulae are selected from the population, and the one with the highest fitness is retained for reproduction. Crossover applies tournament selection twice, to find two parents.

Images rendered during the run are by necessity small, given the computational time needed for their generation and analysis. We often used run-time images with resolutions of approximately 125 by 100, and final rendered output of 1250 by 1000, or 100 times the area. Although smaller run-time images greatly speed up a run, a price is paid in feature test accuracy. A larger resolution image has much more visual detail, which affects the DFN and other scores. Therefore, larger run-time images will result in more accurate results. If an appropriately-scaled Gaussian blur filter is applied to a large image before feature tests are performed, the results obtained are closer to what are seen with smaller resolution versions of the image.

V. RESULTS

Figure 1 shows some results.[1] The search uses four target objectives: DFN=0.0, mean=3.75, std dev=0.75, and colour matching with the image in (a) (ie. CHISTQ=1.0). Textures that are to be analyzed in the run are rendered at a resolution of 126 by 98. The final results are rendered at 1260 by 980. One result is the image in (b). When processed with a 3 by 3 gaussian filter (to approximate the detail in the smaller image analyzed in the run), the resulting statistics are DFN=6.4, mean=1.8, std dev=0.68, CHISTQ=0.82. Such results are typical with multi-objective searches, in which some scores are strong, while others are weak (in this case, the mean score). Usually a CHISTQ score above 0.80 means that the image colour yields an acceptable match to the target. Image (c) shows the gradient response filter applied to the solution image. Figure 2 shows the distribution of the gradient response for this solution. Figure 3 shows how the DFN stays low as the gaussian filter is increased, which simulates the eye viewing the image at farther distances.

Images (d), (e) and (f) in Figure 1 are other results using the same target image as above. Image (d) has a DFN=7.9, mean=3.7, std dev=0.54, and CHISTQ=0.92. Image (e) has a DFN=15.0, mean=4.6, std dev=0.72, and CHISTQ=0.90. Image (f) is taken from an earlier generation (25). Less evolved images such as this one are typically bolder, more primitive, and less refined than latter ancestors, since their scores are still far from the target values.

Figure 4 shows some results from an experiment in which a greyscale image in (a) is used as the colour target.

In Figure 5, the Union Jack in (a) is the colour target. That image is detrimental to both the CHISTQ and DFN tests, as both prefer richer gradients than the 3 colours in the image. One interesting result is in (b). To get a lower DFN score, a 3D smoke effect was placed between the red and yellow bands, as is seen in the detail in (c).

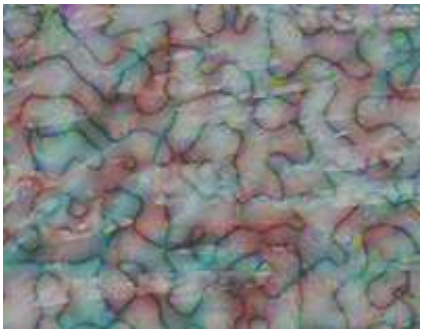
[1]Full-sized colour images can be seen at $www.cosc.brocku.ca/\sim bross/DFN\_gallery/$.

(a) Colour target.

(b) One result.

(c) Colour gradient of (b).

(d)



(e)

(f)

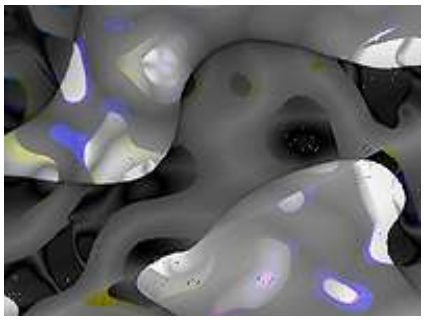
Fig. 1. Results for Monet sea image palette in (a).

(a) Greyscale target.

(b)

(c)

(d) Detail from (b).



(e) Detail from (c)

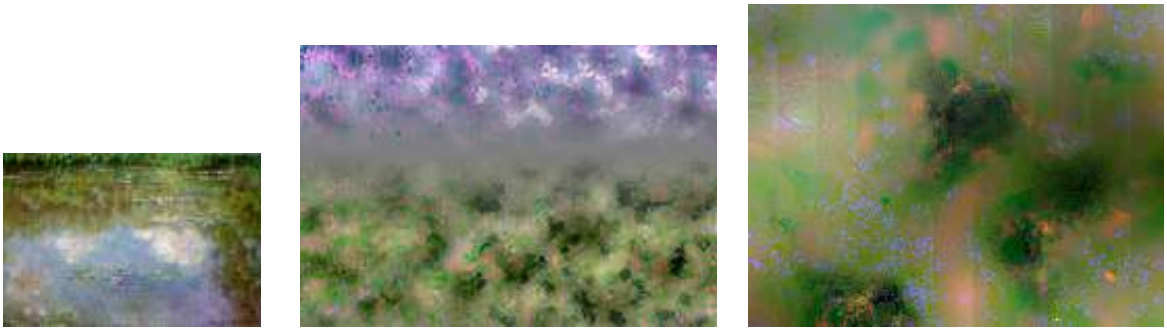
Fig. 4. Greyscale colour target experiment

A classic painting by Monet is used as the target in Figure 6. Image (b) is one result, and a detail from it is in (c).

Figure 7 shows various results from other experiments. Images (a) and (b) are examples of poor results having high DFN scores. The chaotic nature of image (a) gives it a DFN=129. On the other hand, image (b) is overly bland. It was evolved in a run that ignored DFN scores, and its DFN is over 1400. Images in (c) and (d) show how the DFN scoring can produce subtlety and balance. Image (e) is a detail from an image that had a wood-block print effect. The image in (f) is from a run in which the target DFN was inadvertently set too high. The resulting artistic style is more

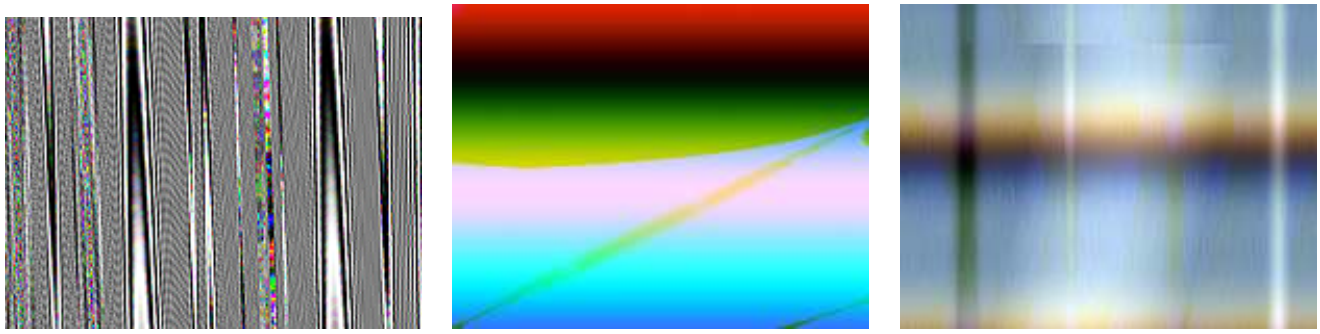
(a) Colour target. (b) Result. (c) Detail from (b).

Fig. 5. Union Jack experiment



(a) Colour target. (b) Result. (c) Detail from (b).

Fig. 6. Monet experiment



(a) DFN=129 (b) DFN $>$ 1400 (c)

(d) (e) (f)

Fig. 7. Miscellaneous results

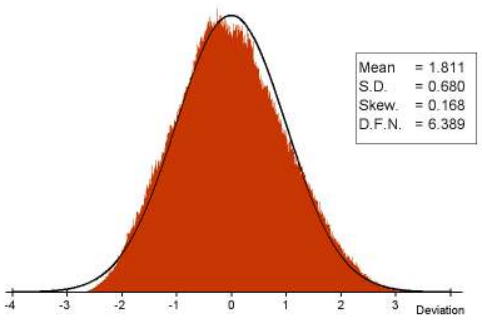


Fig. 2. Gradient plot for Fig.1 (b)

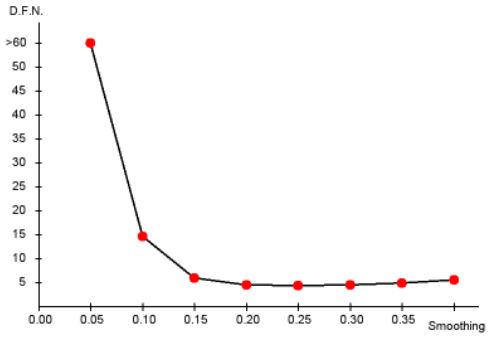


Fig. 3. DFN vs blurring for Fig.1 (b)

graphic design-oriented than the more ambient and painterly styles seen with images having low DFN's (see discussion of Figure 8 below).

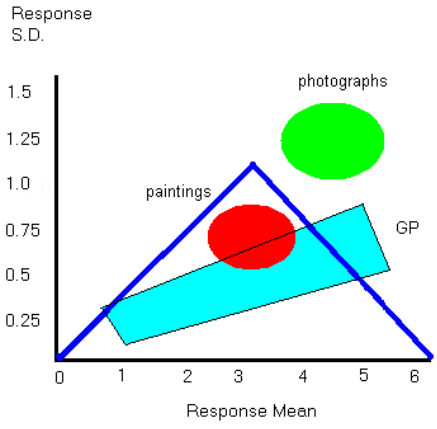


Fig. 8. Gradient response space for art work and evolved images

The research in [2] characterizes the observed normal distributions of art work with the graph in Figure 8. The dark blue pyramid demarcates the area within which the response curves of paintings are often found. Paintings with low DFN scores consistently have means and standard deviations within a "sweet spot" in the pyramid, marked with the red circle. Amateur artists very rarely create pieces that reside in this locale. This suggests that the sweet spot is an aesthetic ideal implicitly striven for by many classical painters. Graphical art and photographs tend to be in the area marked with the green circle, and did not have strong DFN scores. We found that our evolved images tended to reside in the light blue rectangle. When the colour palette scoring was not performed, the DFN scores were consistently in the bottom-left of the triangle, in the area of low mean and standard deviation. The sweet spot was occasionally hit by our images. As with human artists, the sweet spot is also technically challenging for genetic programming.

VI. Conclusion

This paper has shown that aesthetic texture evolution benefits from Ralph's bell curve response model [2]. We conjecture that the area of image space defined by low DFN scores is more densely populated with interesting images, than the areas where the DFN is high. When experiment parameters were appropriately tuned, we conservatively estimate that 10% of our results were visually interesting. On the other hand, when the bell curve analysis is removed, the resulting images are chaotic or bland, artificial, mathematical, and rarely appealing. Thus, the bell curve score is a heuristic that directs the search into regions of texture space that are more likely to be balanced, harmonious, painterly, and "easy on the eyes". This is important for automatic synthesis, where the user has no control over the direction of search. Nevertheless, the user still makes the ultimate decision whether the synthesized images are appealing. Art is in the eye of the beholder.

Ralph's bell curve response models the degree of visual non-uniformity that is resident in a large number of classical paintings. This is not to suggest, however, that all synthetic images with low DFN's are necessarily interesting. For example, we performed runs in which the DFN was the sole criteria, and the colour score was ignored. Although the resulting images had excellent DFN scores, they were inevitably bland and boring, due to the use of a narrow bandwidth of RGB space. This was advantageous for obtaining low DFN scores, but not very interesting to the human eye. The colour test introduces the need for a palette of colours, which is the norm in visual art. In fact, the mood of a target image as intended by the original artist is often injected into the texture image, visaviz the colour palette.

Interestingly, we found that some images with unusually poor scores were visually fascinating. This does not suggest that the fitness criteria is irrelevant. These particular images have visually pleasing traits inherited from more well-behaved and refined ancestors. Furthermore, sometimes the chosen colour target image is "DFN unfriendly". In such cases, the DFN and colour tests work against each other. The resulting "creative tension" often produces the most surprising results (image (c) in Figure 5).

The images in [15] share some stylistic characteristics with many of our results. The aesthetic models of both systems are unfavourable to images that are either too static or chaotic. Their model's use of fractal compression tends to favour

images that are self-similar in nature. The NEvAr system uses greyscale, while we work with a target colour palette.

Both the Bell distribution and NEvAr's aesthetic models likely reside within 1/f space. The *1/f* distribution is applicable to a wide variety of phenomena that reside between total order and total chaos [18]. Its generality makes it too coarse for modeling more refined phenomena, such as fine art.

Research in [27] has applied many of this paper's ideas, including the Bell curve aesthetic model, toward the evolution of non-photorealistic image filters. The results of that research confirm that the Bell curve model acts as a taming influence to the kinds of filter effects obtained. Furthermore, that research discovered that paint brush strokes are conducive to Bell curve gradient distributions. This partially explains why we occasionally obtained paint stroke effects on textures, which are not an easy effect to obtain with our GP texture language and its noise generators.

There are a number of future directions for this research. Other measurable aspects of aesthetics could be incorporated that address composition and colour selection in more detail. The evolution of images that have non-normal distributions (higher DFN's) is worth considering. Image (f) in Figure 7 shows that interesting graphical designs are possible with non-bell distributions. New texture languages would result in new styles of images. The language used here is predominated by noise generators, which directly influences the style of abstract art obtained.

ACKNOWLEDGMENT

Thanks to Andrea Wiens and Han Zhu for their earlier work on Gentropy. B.J. Ross was supported by NSERC Operating Grant 138467-1998, and H. Zong by an NSERC USRA award.

REFERENCES

[1] J. Koza, *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. MIT Press, 1992.

[2] W. Ralph, "Painting the Bell Curve: The Occurrence of the Normal Distribution in Fine Art," *In preparation.*, 2006.

[3] A. Dorin, "Aesthetic Fitness and Artificial Evolution for the Selection of Imagery from the Mythical Infinite Library," in *Advances in Artificial Life – Proc. 6th European Conference on Artificial Life*. Springer-Verlag, 2001.

[4] M. Whitelaw, "Breeding Aesthetic Objects: Art and Artificial Evolution," in *Creative Evolutionary Systems*, P. Bentley and D. Corne, Eds. Morgan Kaufmann, 2002, pp. 129–145.

[5] K. Sims, "Interactive evolution of equations for procedural models," *The Visual Computer*, vol. 9, pp. 466–476, 1993.

[6] J. Graf and W. Banzhaf, "Interactive Evolution of Images," in *Proc. Intl. Conf. on Evolutionary Programming*, 1995, pp. 53–65.

[7] A. Rowbottom, "Evolutionary Art and Form," in *Evolutionary Design by Computers*, P. Bentley, Ed. Morgan Kaufmann, 1999, pp. 330–365.

[8] M. Lewis, "Aesthetic Evolutionary Design with Data Flow Networks," in *Proc. Generative Art 2000*, 2000.

[9] S. Rooke, "Eons of Genetically Evolved Algorithmic Images," in *Creative Evolutionary Systems*, P. Bentley and D. Corne, Eds. Morgan Kaufmann, 2002, pp. 330–365.

[10] A. Ibrahim, "GenShade: an Evolutionary Approach to Automatic and Interactive Procedural Texture Generation," Ph.D. dissertation, Texas A&M University, December 1998.

[11] A. Wiens and B. Ross, "Gentropy: Evolutionary 2D Texture Generation," *Computers and Graphics Journal*, vol. 26, no. 1, pp. 75–88, February 2002.

[12] B. Ross and H. Zhu, "Procedural Texture Evolution Using Multiobjective Optimization," *New Generation Computing*, vol. 22, no. 3, pp. 271–293, 2004.

[13] A. Hewgill and B. Ross, "Procedural 3D Texture Synthesis Using Genetic Programming," *Computers and Graphics*, vol. 28, no. 4, pp. 569–584, 2004.

[14] S. Baluja, D. Pomerleau, and T. Jochem, "Towards Automated Artificial Evolution for Computer-generated Images," *Connection Science*, vol. 6, no. 2/3, pp. 325–354, 1994.

[15] P. Machado and A. Cardoso, "All the Truth About NEvAr," *Applied Intelligence*, vol. 16, no. 2, pp. 101–118, 2002.

[16] ——, "Computing Aesthetics," in *Proc. XIVth Brazilian Symposium on AI*. Springer-Verlag, 1998, pp. 239–249.

[17] N. Svangard and P. Nordin, "Automated Aesthetic Selection of Evolutionary Art by Distance Based Classification of Genomes and Phenomes using the Universal Similarity Metric," in *Applications of Evolutionary Computing: EvoWorkshops 2004*. Springer, 2004, pp. 447–456, lNCS 3005.

[18] M. Schroeder, *Fractals, Chaos, Power Laws*. W.H. Freeman and Company, 1991.

[19] L. Kiss, Z. Gingl, Z. Marton, J. Kertesz, F. Moss, G. Schmera, and A. Bulsara, "1/f Noise in Systems Showing Stochastic Resonance," *Journal of Statistical Physics*, vol. 70, no. 1/2, pp. 451–462, 1993.

[20] D. Tolhurst, Y. Tadmor, and T. Chao, "Amplitude spectra of natural images," *Ophthalmic & Physiological Optics*, vol. 12, no. 2, pp. 229–232, 1992.

[21] J. Pressing, "Sources for 1/f noise effects in human cognition and performance," *Paideusis: Journal for Interdisciplinary and Cross-Cultural Studies*, vol. 2, 1999.

[22] R. Voss and J. Clarke, "1/f noise in music: Music from 1/f noise," *J. Acoustical Society of America*, vol. 63, no. 1, pp. 258–263, 1978.

[23] D. Ebert, F. Musgrave, D. Peachey, K. Perlin, and S. Worley, *Texturing and Modeling: a Procedural Approach*. Academic Press, 1994.

[24] H. Elias. (2005) Cloud Cover. [Online]. Available: http://freespace.virgin.net/hugo.elias/models/m_clouds.htm

[25] J. Smith and S.-F. Chang, "Visualseek: a fully automated content-based image query system," in *Proc. ACM Intl. Conf. on Multimedia (ACM-MM)*, 1996, pp. 87–98.

[26] C. C. Coello, D. V. Veldhuizen, and G. Lamont, *Evolutionary Algorithms for Solving Multi-Objective Problems*. Kluwer Academic Publishers, 2002.

[27] C. Neufeld, B. Ross, and W. Ralph, "The Evolution of Artistic Filters," 2005, submitted.