Evolving Aesthetic Images using Multiobjective Optimization

Gary R. Greenfield
Mathematics and Computer Science
University of Richmond
Richmond, VA 23173
ggreenfi@richmond.edu

Abstract- We consider the problem of using evolutionary multiobjective optimization to evolve visual imagery. In our method, images (phenomes) are generated from expressions (genomes), and then color segmented so that they can be evaluated under a number of different aesthetic criteria. Our principal task is to formulate fitness functions that make the best use of these elementary aesthetic components. We demonstrate the benefits obtained from using more than one objective function. We also discuss technical issues that arose as a consequence of treating our computational aesthetics problem as a "real world" application of evolutionary multiobjective optimization.

1 Introduction

Although the work of Dawkins [dawk89] set precedent, the real impetus for using the *interactive* genetic algorithm to evolve aesthetic visual imagery gained momentum following the well publicized efforts of Sims [sims91] and Latham [todd92]. Because Sims "evolving expressions" methodology for evolving aesthetic images is more tractable, and has attracted a wider following (see, for example, [gree98] [rowb99] [unem02] [voss95] [witb99]), we will restrict our attention to the problem of guiding the evolution of aesthetic images belonging to populations of two-dimensional *abstract* images generated using Sims' method.[1] The specific aspect of the problem that we are interested in is how to automate the aesthetic decision making process and therefore bypass the use of the interactive genetic algorithm requiring a user to perform the tedious task of making aesthetic decisions generation after generation in order to decide which images will survive and be included in the breeding pool.[2] In this context, automated aesthetic image evolution falls under the realm of *computational aesthetics*, the discipline that investigates methods for assigning quantitative measures of aesthetic value to artistic works.

The first published research that used a computational aesthetics approach for evolving images generated using Sims' method relied on neural nets to make the aesthetic decisions[balu94].[3] Unfortunately, neural nets appeared to be overwhelmed by the information content contained in this type of imagery. This observation prompted the author to apply co-evolutionary methods that relied on analyzing small carefully selected portions of images belonging to populations that were evolved using Sims's method [gree00a] [gree02b].

Cognitive scientists have not yet reached a consensus about how humans make aesthetic decisions [rama99] [zeki99], but it is clear that humans do not make aesthetic judgments without taking the entire image into consideration. Therefore, more recently, the author began investigating methods for distilling the aesthetic content of images in such a way that aesthetic decisions could be made on the basis of the geometric information that these encodings provided [gree02a]. However, the use of the *simple* genetic algorithm often led to premature convergence, and the results were neither wholly successful nor robust. Only one fitness function was used, and the criteria for formulating such functions were haphazard. In this paper, we expand upon our technique by (1) employing multiple fitness functions, and (2) considering more systematic ways of designing such fitness functions.

2 Images from Expressions

The genotype g of an image is a postfix expression constructed using functions of arity zero, one, and two. The zero-ary functions are the floating point constant functions $C_i = i/1000$ for $0 \leq i < 1000$, together with the variables V_0 and V_1. The five unary functions, defined from the unit interval to itself, are $U_0, \ldots, U_4$; and the fifteen binary functions, defined from the unit square to the unit interval, are $B_0, \ldots, B_{14}$. Their definitions may be found in [gree02a]. An example of a "small" genotype is

$$C_{289}\ V_1\ U_2\ V_0\ B_2\ B_{13}\ U_4.$$

In order to generate a $v \times v$ pixel image from an expression g using a color look-up table consisting of N colors, we assign to pixel $p_{i,j}$ the k-th color in the table provided

$$k/N \leq g(i/v, j/v) < (k+1)/N.$$

That is, g is a representation of a function in two variables $g(V_0, V_1)$ from the unit square to the unit interval, and evaluation at the point determined by the pixel's coordinates is scaled to provide an index into the color look-up table of size N. The table we use is the one described in [gree02a]. It is defined in HSV color space by specifying nine different shades for each of fifty different hues.

---

[1] For a rare example of using the interactive genetic algorithm to evolve images that are not abstract see [graf95]. For an interesting example of using the interactive genetic algorithm to evolve virtual organisms with aesthetic visual *behaviors* see [ray98].

[2] For a radically different, albeit untested, approach to this problem see [chao03].

[3] For another early nontrivial example of computational aesthetics and computer generated art see [spro96].

3 Aesthetic Evaluation of Images

To evaluate images for aesthetic purposes, we introduce several elementary aesthetic fitness metrics, or *components*, that are derived from the color segmentations of images. These components will be used to construct our fitness functions.

3.1 Color Segmentation of Images

In this subsection we review a color segmentation algorithm we introduced in [gree02a] and [gree03a]. Augment the genotype for each image with a 6-tuple of real numbers

$$(k_{vv}, k_{vs}, k_{ss}, k_v, k_s, k_h)$$

whose values lie in the interval $[0, 3]$. Color segment the 32×32 "thumbnail" phenotype generated from the expression part of the genotype by assigning a priority $p(e)$ to every pixel edge e, and then using the edge of *minimum* priority to determine two adjacent simply connected regions that can be merged to form a new simply connected region. Initially, each pixel is designated to be a region. A sequence of $v^2 - n$ region merges reduces the thumbnail from v^2 to n regions. All but the last 25 of these merges are induced from edges of minimum priority that are calculated on the basis of region color as described below. The last 25 merges are reserved for "cleaning-up" the image, whence edge priority is calculated on the basis of region size in order that the smallest extant regions can be merged into the segmentation (see [gree03b] for details). Working on the basis of color, we define edge priority by letting

$$p(e) = k_v\Delta_v + k_s\Delta_s + k_h\Delta_h + G(\Delta_v, \Delta_s)\Delta_h$$

where Δ_v, Δ_s and Δ_h are the *magnitudes* of the differences in value, saturation, and hue across edge e, and $G(\Delta_v, \Delta_s)$ is a second-order weighting function applied to the difference in hue. It is obtained by setting

$$G(\Delta_v, \Delta_s) = k_{vv}\Delta_v^2 + k_{vs}\Delta_v\Delta_s + k_{ss}\Delta_s^2.$$

Loosely speaking, the segmented image helps reveal the underlying morphology (i.e., structure and color organization) of the image.

3.2 Aesthetic Components for Images

For a region X of the segmented image, we let $a(X)$, $b(X)$, and $j(X)$ denote its area (number of pixels), boundary length, and number of region adjacencies respectively. We index the regions $X_1, \ldots, X_n$ of the segmentation so that they are sorted according to descending area. Given s and t satisfying $1 \le s \le t \le n$, we define

$$\begin{aligned} A_{s,t} &= \sum_{k=s}^{t} (k+1)a(X_k), \\ B_{s,t} &= \sum_{k=s}^{t} b(X_k), \\ J_{s,t} &= \sum_{k=s}^{t} j(X_k). \end{aligned}$$

Evidently A, B, and J evaluate the image by considering (weighted) areas, boundaries, and adjacencies summed over a consecutive sequence of regions. Loosely speaking, their purpose is to gauge how large, how finely detailed, and how interlocking such a sequence is. Observe that all of our components assume non-negative integer values.

If a component itself is used as an objective function it would typically evolve *degenerate* images. To understand why, we consider the extreme cases.

- For $A_{1,n}$, the weighting factors used typically force evolved segmentations to have either n horizontal or n vertical stripes of approximately equal width.
- For $B_{1,n}$, the fact that area is held constant usually forces evolved segmentations to consist of approximately n concentric circles.
- For $J_{1,n}$, the requirement that regions have maximal contact with other regions, usually forces evolved segmentations to have one small "clump" consisting of all but the largest region.

Such phenomena are not surprising. As pointed out by Zeki [zeki99, p. 59], "aesthetics is modular," a statement we interpret to mean that given an appropriate set of elementary aesthetic components, they must be properly integrated in order to come to an overall aesthetic decision.

4 Multiobjective Framework

By and large, we adopt the elitist non-dominated sorting genetic algorithm (NSGA-II) as described in Deb [deb01]. For comparison purposes, in this section we give a formal description of our implementation of Deb's algorithm. If g is a genotype in the population P, and f is a fitness function, then $f(g)$ denotes the non-negative integer assigned to the $v \times v$ color segmented image determined from g using f. As an example, the fitness function in the simple genetic algorithm that gave the most consistent results in [gree02a] was

$$f = A_{1,3} + B_{1,25}.$$

Henceforth, we assume there are $m \ge 2$ fitness functions $f_1, \ldots, f_m$.

4.1 Dominance and Non-dominated Fronts

We say genotype g_i *dominates* g_j, written $g_i \succ g_j$, provided $f_r(g_i) \ge f_r(g_j)$ for all r, and there exists s such that $f_s(g_i) > f_s(g_j)$. If S is any non-empty subset of the population P, we define the *non-dominated front* $N(S)$ by

$$N(S) = \{s \in S : \text{for all } t \in S, t \not\succ s\}.$$

By convention, $N(\emptyset) = \emptyset$. Next, we define the non-dominated *fronts* $F_0, F_1, \ldots$ of P inductively, by setting

$$F_0 = N(P),$$

and for $i \ge 0$,

$$F_{i+1} = N(P \setminus \bigcup_{j=0}^{i} F_i).$$

F_0 is called the *leading* non-dominated front. Clearly, the non-dominated fronts partition P. If $g \in F_i$ we define the rank of g, written $R(g)$, to be i. The breeding population B of P is formed by adjoining successive non-dominated fronts to the leading front until the mandated size is obtained. Since we repopulate during each generation by using tournament selection to provide genotypes from which to form breeding pairs, we require a mechanism to break ties when both participants have equal rank.

4.2 Crowding Distances and Tournament Selection

Let $F = \{g_1, \ldots g_t\}$ be a front. For each r, where $1 \leq r \leq m$, let

$$\begin{aligned} M_r &= \max_{g \in F}\{f_r(g)\}, \\ L_r &= \min_{g \in F}\{f_r(g)\}, \\ D_r &= \max\{M_r - L_r, 1\}. \end{aligned}$$

Let $(\ell_{1,r}, \ell_{2,r}, \ldots, \ell_{t,r})$ be the vector of indices that results when F is sorted in ascending order under f_r. This means, for $1 \leq i < t$, $f_r(g_{\ell_{i,r}}) \leq f_r(g_{\ell_{i+1,r}})$. Then, if $g = g_{\ell_{i,r}}$, we define

$$d_r(g) = \begin{cases} \frac{f_r(g_{\ell_{i+1,r}}) - f_r(g_{\ell_{i-1,r}})}{D_r} & \text{if } 1 < i < t \\ \infty & \text{otherwise} \end{cases}$$

We use $d_r(g)$ to define the *crowding distance* $C(g)$ by setting

$$C(g) = \sum_{r=1}^{m} d_r(g).$$

The crowding distance is the sum of the dimensions of the “cuboid” surrounding g. It is used to resolve ties in rank during tournament selection in such a way that diversity is fostered. Ties in rank are broken by choosing the genotype in the sparsest region i.e., largest cuboid. Formally, in a crowding tournament selection contest, genotype g_i wins over g_j if either

$$R(g_i) > R(g_j),$$

or

$$R(g_i) = R(g_j) \text{ and } C(g_i) > C(g_j).$$

4.3 Repopulation

With the requisite machinery in place, the repopulation algorithm is anticlimactic. Population size remains constant. The breeding pool is formed from the non-dominated fronts as described in the preceding subsection. For convenience, both the population and breeding pool are assumed to be of even cardinality. The required number of breeding pairs is formed by making clones of winners of crowding selection tournament contests. By cloning contest winners, members of the breeding pool are eligible to participate in more than one mating. Each cloned pair provides two offspring by subjecting the pair to crossover followed by mutation. Since a genotype consists of a postfix expression and a segmentation coefficient vector, crossover consists of subtree swapping for postfix expressions [keit94] coupled with one-point crossover for vectors.

4.4 Simulation Parameters

The results presented in this paper use a fixed population size of 32 and a fixed breeding pool size of 16. Color segmentation halts after the 32×32 thumbnail is reduced to 25 simply connected regions. Mutation rates are constant, but because exploration is the goal, they are high — both nodes in an expression and components in a coefficient vector have a one in five chance of mutating. Populations are initialized using randomly generating genotypes. Expressions are constrained to have between 35 and 65 nodes. Evolutionary runs last for 100, 400, or 800 generations. At fixed intervals (e.g., 25, 50, or 100 generations), images new to the leading front are culled.

5 Discussion and Results

In theory, each fitness function should serve as an attractor of a subpopulation, and the leading front should collect, in addition to the best genotypes from each subpopulation, additional genotypes that are the result of effective “interactions” between the subpopulations. From this point of view, our challenge is to design fitness functions making use of our aesthetic components in such way that both good subpopulations can arise and good interactions can occur. Our most successful images were evolved by using a strategy that first “balanced” the components and then promoted interaction by constructing round-robin fitness functions.

For our first example, consider the two images in Figure 1 that were evolved using components $A_{1,4}$, $B_{1,4}$, and $J_{1,25}$ as follows. As a preliminary step, we let the three fitness functions f_1, f_2, and f_3 be these components and let evolution proceed for fifty generations in order to estimate the maximum values of each of these components. From this, we inferred that $A_{1,4}/5$, $B_{1,4}$, and $10J_{1,25}$ had approximately the same maximums. These became our trio of *balanced* components. We then constructed a round-robin fitness function trio by letting

$$\begin{aligned} f_1 &= 10J_{1,25} + B_{1,4}, \\ f_2 &= B_{1,4} + A_{1,4}/5, \\ f_3 &= A_{1,4}/5 + 10J_{1,25}. \end{aligned}$$

Evolution proceeded for 800 generations, and any new images that appeared in the leading front after each 100 generation interval, or *epoch*, were culled. The two images shown in Figure 1 are aesthetically superior to all of those obtained in [gree02a] that were evolved using only the simple genetic algorithm. What is remarkable is how multiobjective optimization continued to find aesthetic improvements for hundreds of generations. The top image in Figure 1 was the 4,068-th image that was evaluated and clearly owes its strongest allegiance to fitness function f_1, while the bottom image in Figure 1 was the 7,949-th image that was evaluated and clearly owes its strongest allegiance to f_2.

Our second example shows why one must always maintain extreme vigilance. It comes from an evolutionary run that lasted 800 generations and used a partial round-robin

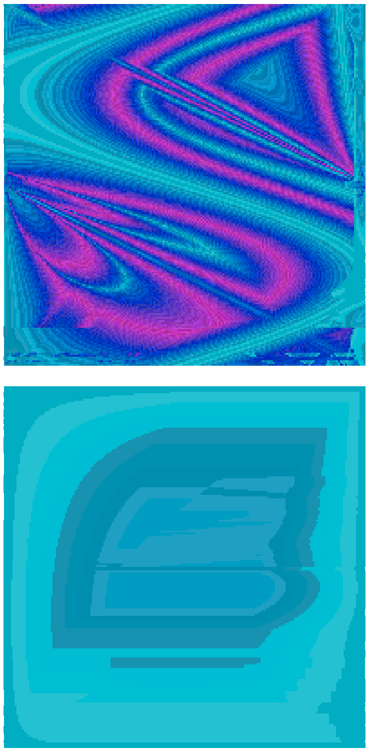

Figure 1: Images that were culled from the leading nondominated front at generations #300 (top) and #500 (bottom) using three "balanced" objective functions in our multiobjective framework.

design where the two fitness functions were given by

$$\begin{aligned} f_1 &= B_{1,25} + A_{1,3}, \\ f_2 &= 10 J_{1,25} + B_{1,25}. \end{aligned}$$

During this run, the average number of genotypes in the leading front was a healthy 11.7, but this diversity was not helpful because the image phenotypes were all degenerate. Since fitness depends on the color segmentation, and the exact nature of the segmentation depends on the (co)evolution of the coefficient vectors, what happened during this run was that the boundary component appearing in the fitness functions exerted undue selective pressure. Figure 2 shows how the boundary trait that this component is responsible for was selected for *in the segmentations* at generations #400 and #700. This example was chosen because, thanks to serendipity, it is a rare instance where the color segmentations were of aesthetic interest in their own right. Somewhat to our surprise, sudden increases in the sizes of leading fronts turned out to be reliable indicators of underlying explosive growths of degenerate images within our populations.

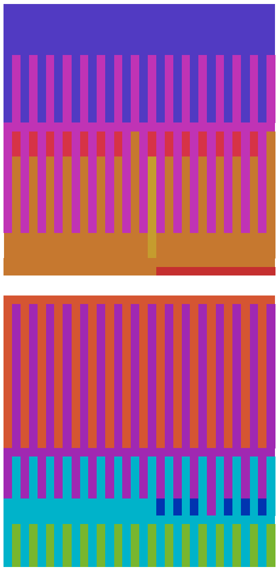

Figure 2: Segmented images from generation #400 (top) and #700 (bottom) showing how a boundary component used in the fitness functions exerted undue evolutionary pressure.

Our third example shows an instance where multiobjective optimization was successful in documenting the occurrence of lineages — sustained evolutionary lines of phenotypes. In the past this has often been difficult to achieve when evolving aesthetic imagery. It is a desirable feature because, in general, during the course of an evolutionary run, one hopes to observe the exploration and development of a visual theme. For this example, we attempted to correct for the component imbalance of the previous example by letting

$$\begin{aligned} f_1 &= B_{1,25} + A_{1,3}, \\ f_2 &= J_{12,25} + B_{1,4}. \end{aligned}$$

During an evolutionary run lasting 400 generations a sequence of five images appeared in the leading front whose phenotypes clearly revealed their genotypic ancestry. Their birth generations were #72, #181, #238, #348, and #398. Figure 3 shows four of these five images. (The image from generation #181 is omitted since its phenotype shows barely perceptible differences from its ancestor.) A simple genetic algorithm that culls only the most aesthetically fit individual at the conclusion of each epoch, would miss this kind of evolutionary development because at the time these images were culled, under fitness function f_1 these five images had fitness indices 2, 3, 2, 3, and 4 respectively within the leading front, while under fitness function f_2 these five images had fitness indices 1, 1, 2, 4, and 3 respectively within the leading front. In other words, even if they were present in the population, a simple genetic algorithm would not have culled any of the images appearing after generation #200.

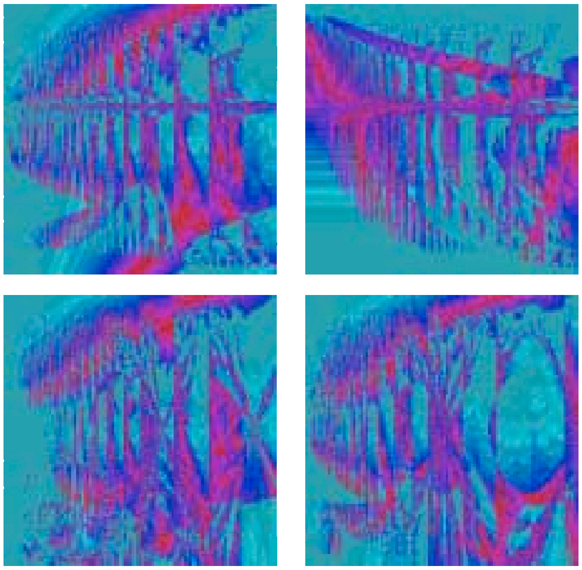

Figure 3: Clockwise from upper left, a phenotypic lineage with images from generations #72, #238, #348, and #398. The use of mulitiobjective optimization caused these images to be culled by ensuring that they were included in the leading non-dominated front.

There is still much to learn about designing fitness functions. Most of our efforts fell short of the mark. It appears that judicious use of *linear* combinations of our aesthetic components yield the best results, but some of our experiments provided evidence that other promising avenues still await discovery. For our fourth and final example, drawn from a series of experiments where sums of products of components were used in a partial round robin design, we set

$$\begin{aligned} f_1 &= (B_{1,4} + A_{1,4}) J_{12,25}, \\ f_2 &= (J_{12,25} + B_{1,4}) A_{1,4}. \end{aligned}$$

These functions produced the promising images shown in Figure 4 that exhibited a cascading “style” we had never before encountered. Unfortunately, convergence was almost immediate since these images appeared within the first twenty generations of a run lasting only 100 generations.

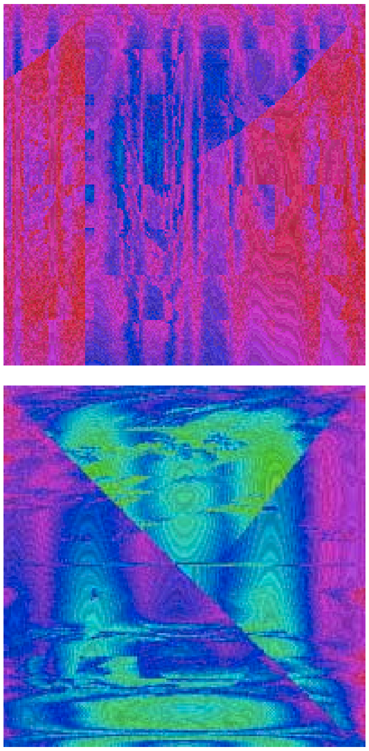

Figure 4: Two promising images that were evolved using fitness functions that were sums of products of our elementary aesthetic components.

6 Technical Issues

Because fitness evaluation of our images is such a costly operation, population sizes are smaller than one would normally expect to find in a genetic algorithm. This has consequences. By definition, once a genotype is included in a front, all genotypes in the population with matching fitness values over all fitness functions – the *equivalence* class — will also appear in this front. Fitness functions are integer valued and we maximize fitness, therefore there is a non-negligible probability that this will occur. This increases the probability that the leading front could be larger than the size of the breeding pool. Since truncation is used to maintain the breeding pool at a fixed size, this, in turn, implies that the *order* in which image genotypes are added to the breeding pool could introduce unexpected breeding biases. Our implementation compensates for this possibility by randomizing the breeding pool prior to its truncation whenever a population yields a *leading* non-dominated front that exceeds the size of the breeding pool. Fortunately, in our experience, this situation will occur only when components do not interact properly and the result is an explosion of degenerate images similar to the ones in Figure 2. There is however another more subtle problem. If a front contains a nontrivial equivalence class, then the order that the genotypes from the class appear within the front affects the assignment of the crowding distances. This, in turn, affects the outcomes of crowding tournament selection contests. We do not know how to best compensate for this bias, nor how to measure whether or not it has any significant impact.

By modifying the definition of the dominance relation so that only one member of an equivalence class is included in any front (i.e., by assuming that $g \not\succ g$ for all g, and then defining, for $g_i \neq g_j$, $g_i \succ g_j$ to mean only that $f_r(g_i) \geq f_r(g_j)$ for all r), we introduce a different bias. Due to small population size, our implementation invokes an unsophisticated algorithm that naively recomputes the sequence of non-dominated fronts from scratch during each generation. Therefore, the first member of any equivalence class that is encountered is the only one that is eligible for inclusion in the front being calculated. In the few runs where we experimented with this modified version of the dominance relation, as a precaution, at fixed intervals, we randomized the order in which the population was examined for dominance. Once again, it was not clear that this was necessary. In the end, we concluded that the lesson to be learned was that it is more important to design fitness functions in such a way that their components can interact properly.

7 Conclusions

We have evolved aesthetic imagery by defining aesthetic components and using them to construct fitness functions in a multiobjective evolutionary framework. We have given examples to show the benefits of culling images from non-dominated fronts of an image population.

Acknowledgments

Thanks to Donald House and the Texas A&M Visualization Lab for hosting a sabbatical during which several of the ideas appearing herein had a chance to germinate and assume tangible form.

Bibliography

[balu94] Baluja, S., Pomerleau, D. and Jochem, T. (1994) Towards automated artificial evolution for computer-generated images, *Connection Science*, **6**, 325–354.

[chao03] Chao, D. and Forrest, S. (2003) Generating biomorphs with an aesthetic immune system, *Artificial Life VIII Conference Proceedings* (eds. R. Standish et al), MIT Press, Cambridge MA, 89–92.

[dawk89] Dawkins, R. (1989) The evolution of evolvability, *Artificial Life* (ed. C. Langton), Addison Wesley, Reading MA, 201–220.

[deb01] Deb, K. (2001) *Multiobjective Optimization using Evolutionary Algorithms*, John Wiley & Sons Ltd., Chichester, England.

[graf95] Graf, J., and Banzhaf, W. (1995) Interactive evolution of images, *Genetic Programming IV: Proceedings of the Fourth Annual Conference on Evolutionary Programming*, (eds. J. McDonnell et al), MIT Press, 53–65.

[gree98] Greenfield, G. (1998) New directions for evolving expressions, *Bridges 1998 Conference Proceedings* (ed. R. Sarhangi), Gilliland Printing, 29–36.

[gree00a] Greenfield, G. (2000) Art and artificial life — a coevolutionary approach, *Artificial Life VII* (eds. M. Bedau et al), MIT Press, Cambridge MA, 529–536.

[gree00b] Greenfield, G. (2000) Mathematical building blocks fo evolving expressions, *Bridges 2000 Conference Proceedings* (ed. R. Sarhangi), Central Plains Book Manufacturing, Winfield KS, 61–70.

[gree02a] Greenfield, G. (2002) Color dependent computational aesthetics for evolving expressions, *Bridges 2002 Conference Proceedings* (ed. R. Sarhangi), Central Plains Book Manufacturing, Winfield KS, 9–16.

[gree02b] Greenfield, G. (2002) On the co-evolution of evolving expressions, *International Journal of Computational Intelligence and Applications*, Vol. 2, No. 1, 17–31.

[gree03a] Greenfield, G. (2003) On attaining maximal fitness or how to evolve virtual redwoods, *Artificial Life VIII* (eds. R. Standish et al), MIT Press, Cambridge MA, 127–130.

[gree03b] Greenfield, G., and House, D. (2003) Image recoloring induced by palette color associations, *Journal of WSCG*, Vol. 11, No. 1, 189–196.

[keit94] Keith, M., and Martin, M. (1994) Genetic programming in C++ : implementation issues, *Advances in Genetic Programming* (ed. K. Kinnear, Jr.), MIT Press, Cambridge MA, 285–310.

[rama99] Ramachandran, V., and Hirstein, W. (1999) The science of art: a neurological theory of aesthetic experience, *Journal of Consciousness Studies*, Vol. 6, No. 6-7, 15–52.

[ray98] Ray, T. (1998) Evolution as artist, *Art@Science*, (eds. C. Sommerer and L. Mignonneau), Springer-Verlag/Wien, 81–91.

[rowb99] Rowbottom, A. (1999) Evolutionary art and form, *Evolutionary Design by Computers* (ed. P. Bentley), Morgan Kaufmann Publishers, San Francisco CA, 261–277.

[sims91] Sims, K. (1991) Artificial evolution for computer graphics. *Computer Graphics*, **25**, 319–328.

[spro96] Sprott, J. (1996) The computer artist and art critic, *Fractal Horizons* (ed. C. Pickover), St. Martin's Press, 77–115.

[todd92] Todd, S., and Latham, W. (1992) *Evolutionary Art and Computers*, Academic Press, San Diego CA.

[unem02] Unemi, T. (2002) SBART 2.4: an IEC tool for creating two-dimensional images, movies, and collages, *Leonardo*, **35**, 189–192.

[voss95] Voss, D. (1995) Sex is best, *WIRED*, December, 156–157.

[witb99] Witbrock, M., and Neil-Reilly, S. (1999) Evolving genetic art, *Evolutionary Design by Computers*, (ed. P. Bentley), Morgan Kaufmann Publishers, San Francisco CA, 251–259.

[zeki99] Zeki, S. (1999) *Inner Vision, An Exploration of Art and the Brain*, Oxford University Press, New York NY.