Bridges
Mathematical Connections
in Art, Music, and Science

Color Dependent Computational Aesthetics for Evolving Expressions

Gary R. Greenfield
Department of Mathematics & Computer Science
University of Richmond
Richmond, VA 23173, U.S.A.
ggreenfi@richmond.edu

Abstract

A well-known method for generating computer art is based on the paradigm of user-guided evolution, a method whereby "artists" interactively search through populations of images in order to select, and subsequently breed, those images which show aesthetic promise. This can be a time consuming endeavor, lacking well-defined principles for controlling evolution and for obtaining images with desired aesthetic characteristics. One way to improve upon this situation is to develop computational criteria for selecting images that conform to user specified aesthetics. We consider a non-interactive evolutionary method that contributes to this field of computational aesthetics by: (1) color segmenting the digital images that we breed so that their color organization can be used to influence aesthetic decision making, and (2) investigating mathematical models for assigning aesthetic fitness to images based on geometric quantities obtained following color segmentation. We provide examples of the aesthetic imagery we obtain.

1. Introduction

Originally conceived of as a tool for studying the concept of "evolvability," Richard Dawkins' interactive simulated evolution program, *Biomorphs*, bred populations of drawing programs to serve as organisms that could assume a visual form [4]. The key idea underlying Dawkins' evolutionary simulation was "user-guided fitness," which meant that the survivability of a drawing program was determined by the user. Subsequently, Karl Sims developed an interactive simulated evolution program for image generation which functioned as an art *medium* with user-guided fitness now taking on the role of user-guided *aesthetics* [16]. Sims freed image generation from procedurally based drawing routines by encoding color images as expression trees.

Many successors followed in Sims' footsteps. Image generation systems that were directly inspired by Sims' expression trees include those of Rooke (as described in [10, 19]), the author [9], Unemi [18], Ibrahim [11], and Mount (as described in [20]). Other well-known image generation systems adopting fitness by aesthetics, but using different image encodings, include Latham's *Mutator* [17] which uses bit string encodings, or Lund's *Artificial Painter* [12] and Machado and Cardoso's *NEvAr* (Neural Evolutionary Art) system [13] both of which use neural net encodings. A definitive survey is well beyond the scope of this paper.

The image generation systems listed above all require a user — the artist — to inspect image populations after each breeding cycle for the purpose of culling the least desirable images, and selecting the most desirable images to breed and pass their genes on to the next generation. In this way, over time, images acquire aesthetic characteristics. It can be a laborious process. The size of the image populations that can be used, the resolution at which images can be displayed to the user, and the time that can be allotted for converting the encoding, or genome, into an image, or phenome, are prohibitive. The paradigm itself has been criticized for its inability to control the evolutionary process and has been maligned for its inability to instill aesthetic intent [2]. For such reasons, research has begun into automating the task of selecting images to be bred for the next generation, thereby introducing *computational* aesthetics as the paradigm for creating aesthetic images. We survey previous work.

Baluja et al examined the possibility of training a neural net to make decisions about which images to breed [1]. Their results were inconclusive, with difficulties stemming from the inability of their neural nets to "digest" the aesthetic content of the imagery their Sims' style image generation system produced. Similarly, Rooke (unpublished) achieved only limited success using a computational aesthetics approach where he attempted to evolve populations of "art critics" to *assist* in breeding image populations. Rooke's art critics assigned aesthetic fitness to his images largely on the basis of various statistics they were able to gather from them. The author investigated assigning aesthetic fitness to images based on a measure of their "complexity" as determined by convolution filtering [7]. A salient feature of this work was that computational aesthetic measures of fitness changed over time due to the fact that the program was co-evolutionary, with images (host expression trees) evolving in competition with their critics (parasitic digital filters). Machado and Cardoso (unpublished) have also investigated assigning aesthetic fitness on the basis of complexity to images generated by their *NEvAr* system [14].

In most interactive evolutionary systems that we are aware of, and certainly in all of our own, image color is extrinsic rather intrinsic. This means that when images are encoded numerically via pixels defined using *one* value, images really have no color at all. Image coloring, even if it is simply gray scale, must be imposed by mappings from the intervals pixel values lie in to colors — a method known as pseudo-coloring or false coloring. From the point of view of assigning aesthetic fitness to images, the use of these pixel values seems undesirable unless one is content to consider only monochromatic images. Here we consider a method, albeit a somewhat sophisticated color segmentation method, for binding the pseudocolor mapping to the fitness evaluation step. Color segmentation allows us to abstract geometry — area, perimeter, and adjacency — from images, which can then be used for developing computational models of aesthetic fitness. We will consider several such models.

This paper is organized as follows. In section two we review the rudimentary features of the evolutionary system we use for generating images. In section three we give a brief overview (full details will appear elsewhere) of our color segmentation algorithm. In section four we consider various ways to incorporate the geometric data we obtain from color segmentation into measures of aesthetic fitness so that they will support *predictable* conclusions about image aesthetics. In section five we present some of the images we obtained. In section six we offer our conclusions.

2. Images from Expression Trees

An expression is a rooted tree. Internal nodes of the tree contain functions, the *primitives*, while external nodes, the leaves, contain either variables (V_0 or V_1) or constants. By requiring that all inputs and outputs of primitives be clamped to the unit interval, we are able to associate a resolution dependent digital image to an expression by deriving values for V_0 and V_1 from pixel coordinates and then evaluating the expression according to the rule of composition of functions. A simple example will help clarify what we mean. For the expression written in prefix notation as $F(V_0, V_1) = \max(\text{mul}(0.758, V_0), \text{sqrt}(V_1))$, we would associate the 32×32 pixel image whose matrix is $(p_{i,j})$, by setting $p_{i,j} = F(i/32, j/32)$, for $0 \leq i, j < 32$. The complete list of primitives we use, including their definitions and resulting gray-scale images may be found in [8]. Our primitives have symbolic equivalents: `C0`, `C1`, ..., `C999` for constants, `U0`, `U1`, ...`U4` for unary functions, and `B0`, `B1`, ...`B14` for binary functions. Using these symbolic equivalents and adopting *postfix* notation allows us to rewrite the example above as `V1 U2 V0 C758 B0 B6`. The manner in which we breed postfix expressions using recombination and mutation operators is described in [8].

In [8] we generated gray-scale images by mapping pixel values lying in the interval $[0, 1)$ to an 8-bit gray scale by mapping those pixel values lying in the interval $[i/255, (i+1)/255)$ to the RGB color $(i/255, i/255, i/255)$. Here too we invoke a color mapping, but one which will we later bind to image fitness. Our color mapping is into HSV color space, a space where each color's saturation and value components lie in the interval $[0, 1]$, but each color's hue component is defined *circularly* on the interval $[0, 6]$. Hue classifies a color according to where it is situated on the perimeter of the HSV color wheel. Our mapping takes pixel values lying in the interval $[0, 1)$ to 450 different HSV colors. These colors comprise fifty hues, with nine shades for each hue. We chose vivid shades by first uniformly spacing hues, and then uniformly spacing each hue's value and saturation starting from 0.7.

3. Biased Color Image Segmentation

In this section we present our method for merging the pixels of an image on the basis of their HSV color into a pre-specified number of *regions* with averaged HSV color. The goal behind this color segmentation is to organize an image into regions of similar hue and value in much the same way that painters are taught to loosely organize their compositions by hue and value. Since our computer generated images are abstract, an example of what we are trying to achieve is most easily demonstrated by examining how our algorithm is able to color segment a familiar image such as Van Gogh's "Starry Night" (See Figure 1). The compositional nature of this Van Gogh helps explain why we use region merging instead of color thresholding to segment images.

To begin our color segmentation, we use each pixel to create a region. These regions have area one, perimeter four, and *average* HSV color equal to the HSV color of their underlying pixels. To achieve our goal we must decide in what order we will merge simply-connected adjacent regions — regions sharing a boundary edge — to form new regions. We determine this by assigning priorities to boundary edges. Merge events are then triggered by the boundary edge with the *lowest* priority. As merges occur, edge priorities change, so segmentation becomes a dynamic process. When a merge occurs, the merged region's HSV color is the area weighted average of the HSV colors of the two regions being merged. This averaging step requires special care because of the way hue is defined.

Edge priority is determined as follows. Let Δ_h, Δ_s, and Δ_v be the *magnitudes* of the differences in average hue, saturation, and value between two regions across a boundary edge e, with the understanding that the difference in hue takes into account its circularly defined scale. We define the edge priority $p(e)$ to be

$$p(e) = (k_h + k_{h,v}\Delta_v + k_{h,s}\Delta_s + k_{h,s,v}\Delta_s\Delta_v)\Delta_h + k_s\Delta_s + k_v\Delta_v.$$

We always set $k_h = 1.0$, and constrain the remaining coefficients of $p(e)$ to lie in the interval $[0, 3]$. Although it is beyond the scope of this paper, we should remark that the coefficients we use for $p(e)$ actually evolve over the course of our aesthetic image generation. One reason we weight the difference in hue by terms involving the differences in value and saturation is because we discovered that by introducing such biases we could tune our priority function on a per image basis so that it could yield acceptable results for a wide variety of photorealistic and non-photorealistic images. Because our image segmentation is slow compared with other color segmentation algorithms (see for example [3] or [5]) we must re-emphasize that region merging allows to control precisely the number of simply-connected regions we will obtain from segmentation.

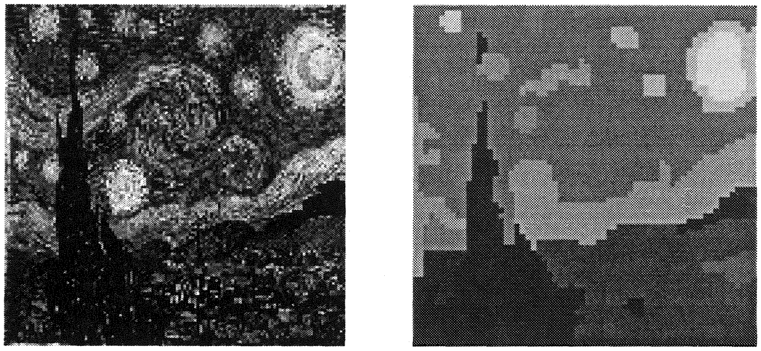

Figure 1: A 128 × 128 pixel digital version of Van Gogh's "Starry Night" is color segmented into twenty-five regions using our region merging algorithm.

4. Computational Aesthetic Metrics

We can now describe the framework underlying our generation of images on the basis of computational aesthetics. To each expression tree we associate a low-resolution 32×32 pixel HSV colored image and a high-resolution 200×200 pixel HSV colored image. We color segment the low-resolution version into an image with $n = 25$ merged regions. Segmentation organizes the image in a way that, hopefully, bears some resemblance to how a user might organize it on the basis of color alone. Figure 2 shows the high-resolution, low-resolution, and color segmented versions of images obtained from the postfix expression

```
C209 C308 V1 V1 B4 V0 V0 B7 B12 U2 U4 C209 B13 V1 V1 B3 C80 V0 V0 B8 C168 B12
U1 V1 B13 C892 B10 V1 B1 B2 B14 C759 B9 V1 V1 B3 U3 B12 U4 U3 V0 B12 B8 B1 B13
```

color segmented using $k_{h,v} = 2.23$, $k_{h,v,s} = 0.94$, $k_{h,s} = 1.22$, $k_v = 2.89$, and $k_s = 2.79$.

Following image segmentation, we attempt to discover whether or not an image is aesthetically promising. In order to do so, we must try to make use of the color, area, perimeter, and adjacency

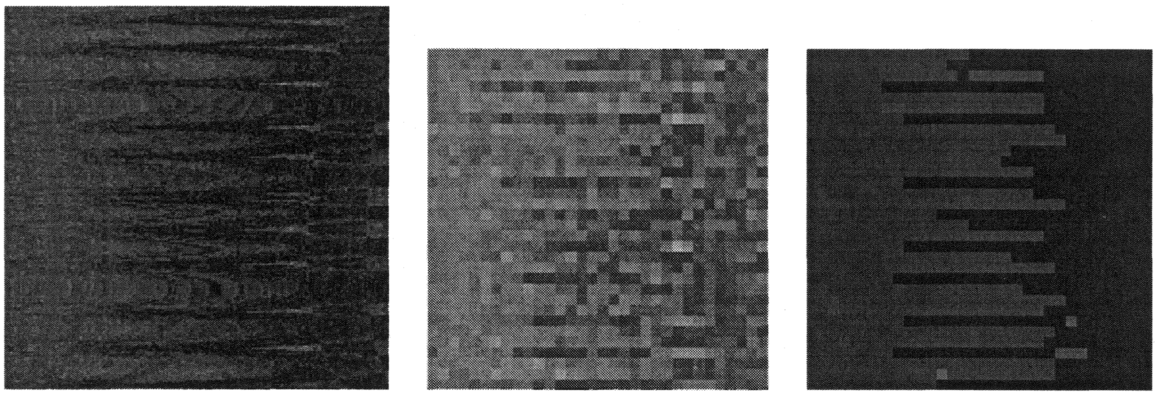

Figure 2: The high-resolution 200 × 200 image obtained from an expression tree, together with its low-resolution 32 × 32 counterpart and its segmentation.

information provided by the segmented version. It appears to be a difficult AI problem to decide how to make further use of the color content — we know of no rules to apply — so we focus instead on the underlying geometry. We want computational fitness measures calculated from the geometry to give rise to predictable imagery, allowing us to breed the aesthetic characteristics we want. Assume segmentation of image I yields regions $R_1, R_2, \ldots, R_n$ having areas $a_1, a_2, \ldots, a_n$ and perimeters $p_1, p_2, \ldots, p_n$. Assume, further, that these regions are *sorted* so that their areas are in descending order $a_1 \geq a_2 \geq \ldots$. For completeness, we remark that since regions may have genus greater than zero, a region's "perimeter" actually refers to sum of the lengths of its boundary edges. We let $f(I)$ denote the aesthetic fitness of image I.

EXAMPLE 1. Consider $f_1(I) = a_1$. In this case we would predict a maximally fit aesthetic image would consist of one large region bounding, hence possibly *enclosing*, $n - 1$ regions of area one. This description only applies to the segmented version of the associated low-resolution image. The associated high-resolution image should have this general *appearance*, but should be augmented with better detail.

EXAMPLE 2. Consider $f_2(I) = \sum p_i$. We are maximizing the sum of the lengths of the perimeters of the regions. Conceivably, this could result in a twisty, interlocking composition which would be very desirable indeed. Surprisingly, we encounter a stumbling block. Image generation, now reduced to an optimization problem, falls into the trap of finding a local maximum from which it is unable to "escape" during the course of its search for an aesthetically superior global maximum. In our tests, images consisting almost entirely of perfectly horizontal or vertical bands were the local maxima from which image breeding could not escape during the course of evolution. One way to try to avoid such uninteresting local maxima is to try to introduce more tension into the fitness measure.

EXAMPLE 3. Consider $f_3(I) = 2a_1 + 3a_2 + \sum p_i$, which rewards images that have two large regions *and* whose perimeter sum is as large as possible. Tension arises here by asking larger regions to sacrifice area in favor of the perimeter sum. This fitness measure, and similar ones (*e.g.*, $f_4(I) = 2a_1 + 3a_2 + 4a_3 + \sum p_i$, which incorporates a third largest area term) produced some of our best images. We also obtained favorable results by using only "tails" of the perimeter sum.

We now consider a third geometric quantity associated with our segmented images. Let A be the total number of *adjacencies* between regions. Formally, if $(\delta_{i,j})$ is the adjacency matrix whose Kronecker delta is one when region R_i is adjacent to region R_j and zero otherwise, then A is the sum of the off-diagonal entries divided by two. By measuring contact, A might help measure how maze-like the image is.

EXAMPLE 4. Consider $f_5(I) = A$. Using this measure of aesthetic fitness we obtained several images consisting of parallel undulating curves or nested rectangles. The refinement $f_6(I) = 5A + a_1$, with which we attempted to "grow" embedded regions into threads, instead clumped all those regions into a ball. By contrast, the metric $f_7(I) = 10A + p_1$ produced marvelous *segmented* images, but the non-segmented versions were rather incoherent. The metric $f_8(I) = 5A + p_1 + p_2$ fared somewhat better in this regard.

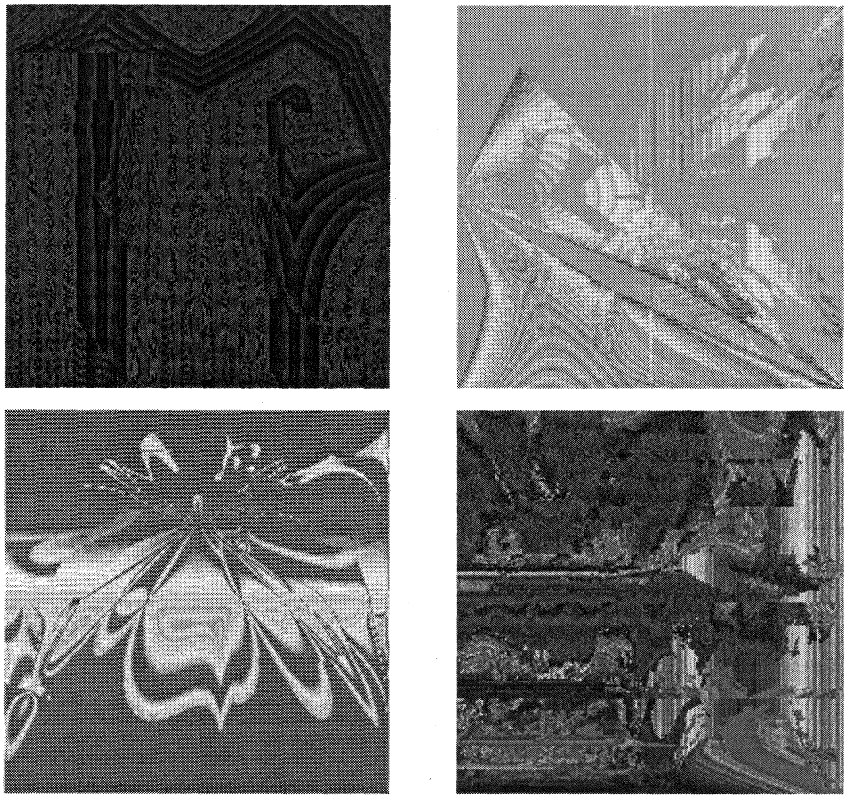

Figure 3: Top left: aesthetic fitness favoring the largest area and a perimeter sum. Top right: aesthetic fitness favoring the two largest areas and a perimeter sum. Bottom left: aesthetic fitness favoring the three largest areas and a perimeter sum. Bottom right: aesthetic fitness favoring the perimeter of the largest area and the number of adjacencies.

5. Images Obtained from Computational Aesthetics

In this section we present images culled from the archives of the aesthetic imagery we obtained during the course of our research on computational aesthetics. Space limitations prohibit showing the segmented versions that were used to assign their fitness. Figure 3 shows three images that were obtained using aesthetic fitness measures of the type described in Example 3 and one image obtained using the adjacencies method of Example 4.

6. Conclusions

It is a truism that evolutionary optimization methods can only be successful if the proper genetic building blocks and genetic recombination operators are in place. In other words, evolutionary optimization only makes sense if there is the potential for organisms to be bred to solve the problem! Our computational aesthetic framework appears sound, and did produce many interesting images, but by testing the theory using images generated from expression trees incorporating our unusual set of building blocks, we may not have availed ourselves of the best choice for image encodings. It would be interesting to apply our methods to a system where image encodings were *stroke* based or *symmetry* based. This should provide better linkage between segmented low-resolution images and their high-resolution phenotypes. The imperative image design language used by Maeda [15] or the functional image design language used by Elliot [6] might sever as good candidates.

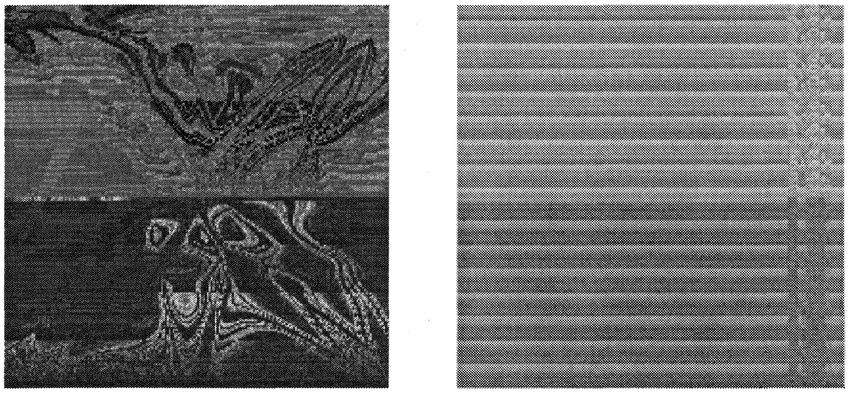

Figure 4: A promising image obtained at generation thirty leads to an evolutionary dead end by generation one hundred and sixty as a perimeter sum term used in calculating aesthetic fitness seizes control.

The problem of deciding how to construct aesthetic fitness measures that yield local maxima that are not "degenerate" images can be a vexing one. Figure 4 illustrates how this problem sometimes develops over time. After thirty generations evolution found an aesthetically promising maximally fit image, but by generation one hundred and sixty the maximally fit image was degenerate and evolution reached a dead end for this lineage. This suggests that our methods might prove best suited for use with large image populations evolved for relatively short periods of time.

References

[1] Baluja, S., D. Pommerleau, and T. Jochem, Towards automated artificial evolution for computer-generated images, *Connection Science*, Volume 6, Numbers 2 & 3, 1994, 325–354.

[2] Boden, M., Agents and creativity, *Comm. of the ACM*, Volume 37, Number 7, 1994, 117–121.

[3] Comaniciu, D., and P. Meer, Robust analysis of feature spaces: color image segmentation, *Proceedings of CVPR '97*, 1997, 750-755.

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

[5] Y. Deng, B. Manjunath, and H. Shin, Color image segmentation, *Proceedings of CVPR '99*, 1999.

[6] Elliot, C., Functional image synthesis, *2001 Bridges Conference Proceedings* (ed. R. Sarhangi and S. Jablan), 2001, 139–158.

[7] Greenfield, G., Art and artificial life — a coevolutionary approach, *Artificial Live VII Conference Proceedings* (ed. M. Bedau et al), MIT Press, Cambridge, MA, 2000, 529–536.

[8] Greenfield, G., Mathematical building blocks for evolving expressions, *2000 Bridges Conference Proceedings* (ed. R. Sarhangi), 2000, 61–70.

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

[10] Hitchcock, N., Painting pictures through time, *Computer Artist*, December/January 1996, 9–9.

[11] Ibrahim, A., GenShade, *Ph.D. Dissertation*, Texas A&M University, 1998.

[12] Lund, H., L. Pagliarini, and O. Miglino, Artificial painter, *Abstract Book of the Third European Conference on Artificial Life (Ecal '95)*, 1995.

[13] Machado, P. and A. Cardoso, Computing aesthetics, in *Proceedings XIV-th Brazilian Symposium on Artificial Intelligence SBIA'98, Porto Allegre, Brazil* (ed. F. Oleiveira), Springer-Verlag, LNAI Series, New York, NY, 1998, 219–229.

[14] Machado, P. and A. Cardoso, NEvAr — the assessment of an evolutionary art tool, in *Proceedings of the AISB '00 Symposium & Cultural Aspects and Applications of AI & Cognitive Science* (ed. G. Wiggins), Birmingham, UK, 2000.

[15] Maeda, J., *Design by Numbers*, MIT Press, Cambridge, MA, 1999.

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

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

[18] Unemi, T., *SBART* — Interactive Art Using Artificial Evolution, User's Manual, 1994.

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

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