Robot Paintings Evolved Using Simulated Robots

Gary Greenfield

Mathematics & Computer Science,
University of Richmond,
Richmond VA 23173, USA
ggreenfi@richmond.edu
http://www.mathcs.richmond.edu/∼ggreenfi/

Abstract. We describe our efforts to evolve robot paintings using simulated robots. Our evolutionary framework considers only the initial positions and initial directions of the simulated robots. Our fitness functions depend on the global properties of the resulting robot paintings and on the behavior of the simulated robots that occurs while making the paintings. Our evolutionary framework therefore implements an optimization algorithm that can be used to try and help identify robot paintings with desirable aesthetic properties. The goal of this work is to better understand how art making by a collection of autonomous cooperating robots might occur in such a way that the robots themselves are able to participate in the evaluation of their creative efforts.

1 Introduction

Open Problem #3 of McCormack's five open problems in evolutionary music and art (EMA) [1] requires one, "To create EMA systems that produce art recognized by humans for its *artistic* contribution (as opposed to any purely technical fetish or fascination)." The recent publicity garnered by the *robot paintings* of Moura, Ramos, and Pereira that resulted from their ARTSBOT (ARTistic Swarm roBOTS) Project might at first glance be seen as a solution to McCormack's third open problem since the paintings are described on the web (see http://alfa.ist.utl.pt/ cvrm/staff/vramos/Artsbot.html) as "artificial art," and in print as "non-human art" [2] or "symbiotic art" [3]. Note that here the symbiosis is intended to be between human and robot. The site http://www.lxxl.pt/artsbot/ where the images of the robot paintings with the best resolution can be found also provides a "Symbiotic Art Manifesto" written by Moura and Pereira.

It is unfortunate that some of the hyperbole associated with the ARTSBOT project detracts from what is potentially a promising new development in evolutionary art. At the center of the ARTSBOT Project lies an implementation of a collective robotics art making *system* to create what are known as *swarm paintings*. The ARTSBOT team reveals this by saying [4] — to paraphrase and polish slightly — that the artworks are made by "a swarm of autonomous robots,

F. Rothlauf et al. (Eds.): EvoWorkshops 2006, LNCS 3907, pp. 611–621, 2006.

that 'live' [by] avoiding simply [executing] streams [of commands] coming from an external computer, [and] instead actually co-evolve within the canvas [environment]; acting [by] laying ink according to simple inner threshold stimulus response functions, [while] simultaneously reacting to the chromatic stimulus present in the canvas environment [left by other robots], as well as by the distributed feedback, [that] affect[s] their future collective behavior." We note that ARTSBOT was one of the few *collective* robotics entries in the most recent international ArtBot art exhibition for "robotic art and art-making robots" (see http://artbots.org/2004/participants/). Moura and Pereira claim that they have created organisms that generate drawings without any intervention on their part thereby creating "a new kind of art" based on a paradigm of non-human autonomous entities that allow personal expression by human artists to be abandoned [3]. Perhaps this somewhat of an exaggeration. Since the controllers for their robots were not evolved, ARTSBOT is not an *evolutionary* art system, but rather an art making system consisting of human programmed autonomous agents that reside and function in an artificial ecosystem. There is a close connection here between *stigmergy* [5] — the situation where autonomous agents alter their environment either accidentally or on purpose in such a way that they influence other agents to perform actions that achieve an objective such as nest building — and swarm painting. The principal difference is that stigmergy is usually associated with a clearly defined task or objective while swarm painting is usually associated with the more poorly defined objective of producing aesthetic imagery.

While in our opinion the question of whether or not the ARTSBOT robot paintings are more than what McCormack referred to as a "technical fascination" has not yet been satisfactorily answered, what is most significant to us is the fact that ARTSBOT does not address McCormack's penultimate challenge, Open Problem #5, which requires one: "To create artificial ecosystems where agents create and recognize their own creativity." In this paper using *simulated* collective robotics and taking for motivation the penultimate problem of how autonomous robots engaged in making swarm paintings might eventually go about learning to *recognize* their own creativity, as a first step we investigate an evolutionary framework that is designed to show how simulated robots might be able to formulate ways to *evaluate* the aesthetic quality of their paintings. Unlike the aesthetic evaluation system for agent produced art studied by Saunders and Gero where each agent produced its own paintings and the evaluation model was based on social dynamics [6], we consider an aesthetic evaluation system where the collective agents are given shared access to a set of image evaluation parameters which can then be used either individually or collectively to modify the image making process. To help understand the consequences of our design, we consider what effect different types of computations made using our set of evaluation parameters have on our robot paintings.

This paper is organized as follows. In section two we provide some background on the use of swarms and the non-interactive genetic algorithm for image making. In section three we give the specifications for our simulated robots. In section four we describe how their controllers work. In section five we present our evolutionary framework. In section six we define our set of image evaluation parameters and then proceed to give examples of some of the robot paintings we evolved using various fitness functions formulated based on these parameters. In section seven we consider the implications of our work for the problem of how robot swarms might go about evaluating their creative efforts. In section eight we offer our summary and conclusions.

2 Background

The notion of swarm paintings was first introduced in an image processing paper by Ramos that was contributed to an ant colony optimization (ACO) conference [7]. Related work appeared in [8] and [9]. In this problem domain the use of the interactive user-guided evolution paradigm that was originally proposed by Sims [10] (i.e. the interactive genetic algorithm) was first studied by Aupetit et al [11]. They investigated an ant colony simulation where the virtual ants deposited and followed color – the *scent* — while exploring a toroidal grid in order to produce "ant paintings." Greenfield [12] non-interactively evolved ant paintings by evolving the genomes required for governing the behaviors of the virtual ants using fitness functions. His observation that only elementary techniques were needed to measure ant exploration and ant cooperation capabilities offers hope that relatively simple behavioral assessment parameters can be used to help identify increased image complexity or well organized image compositions in other evolutionary swarm painting scenarios. The use of the (non-interactive) genetic algorithm in evolutionary art was first considered by Baluja et al [13]. Using this technique for evolving two-dimensional imagery, interesting results have been obtained by Greenfield [14] using co-evolution and the image generation method known as "evolving expressions", by Machado and Cardosa [15] using neural nets, and by Bentley [16] in order to identify cellular automata "patterns." In general, the question of how to evaluate aesthetics on the basis of scientific principles and computational methodologies is a difficult one. To sample several different author's thoughts on the matter and help gauge the scope of the debate see [17, 18, 19, 20, 21].

3 S-Robot Specification

The design of our simulated robots, or S-robots, is loosely based on a software model for Khepera robot simulation by Harlan et al [22]. An S-robot is a virtual circular entity with four binary valued *proximity* sensors together with a three-channel color sensor. Three of the proximity sensors are located at the front of the S-robot and the fourth is located at the rear. The forward and backward sensors scan a field 120° wide and the two side sensors scan a field 45° wide in such a way that there is a 15° overlap with the forward sensor. Thus the forward facing 'field of vision" is from −90° to +90° up to a distance of twenty units and the rear facing field of vision is from −60° to 60° also up to a distance of twenty units. Proximity sensors detect other robots and environmental obstacles or boundaries but do not distinguish between the two. The color sensor is mounted directly beneath center (r_x, r_y) of the S-robot. The S-robot's forward direction is determined by the *unit* vector (d_x, d_y). For all of the images shown here, the robot's two pens were operated synchronously so that either both were up or both were down. The reason for this was so that when the S-robot was mark making, the pen colors could be chosen so that that the mark had an automatic built-in highlight. An S-robot's painting mark is five units wide. An S-robot can swivel (i.e rotate in place) $10°$ clockwise or counterclockwise per clock cycle and can move v units per clock cycle, $-1 \leq v \leq 1$, in either the forward or backward direction in accordance with the sign of v. The S-robot roams on an $n \times m$ unit gridded world.

4 S-Robot Controllers

The "onboard computer" for an S-robot is an interrupt driven controller whose job is to place a sequence of commands in an execution queue, sleep until the queue is empty, and then plan and load the S-robot's next sequence of commands when it is awoken. An S-robot is autonomous because it can place commands in the queue to request sensors readings so that when it is awoken it can perform actions based on these sensor values. The controller loads commands of the form <mnemonic> <argument> where the mnemonic is chosen from the list:

MOV	Move
SWI	Swivel
SPD	Set Speed
SNP	Sense Proximity Vector
SNC	Sense Color Vector
PUP	Pen Up
PDN	Pen Down

Only the MOV, SWI, and SPD commands actually make use of the argument, in all other cases it is treated as a dummy argument. By having the controller indicate how far it wants the S-robot to travel, or how many degrees it wants the S-robot to swivel, the burden of timing shifts to the simulator itself. The simulator calculates how many clock cycles these actions will take so that it can manage the discrete event scheduling, synchronize the movements of all the S-robots, detect collisions, and update the sensors accordingly.

While in the future we would like to evolve the controllers themselves, in this paper we make use of two controllers that we wrote ourselves in order to consider how the cooperation between two S-robots was affected by their initial placement and direction headings. Each of our controllers has four pre-planned painting sequences it can load into the queue. For ease of managing simulated evolution and evaluating the results, at run time we made only one of the four painting sequences available to each controller. The four sequences can produce an elongated double hooked curve, a wedge, a segment of a spiral, and a zigzag

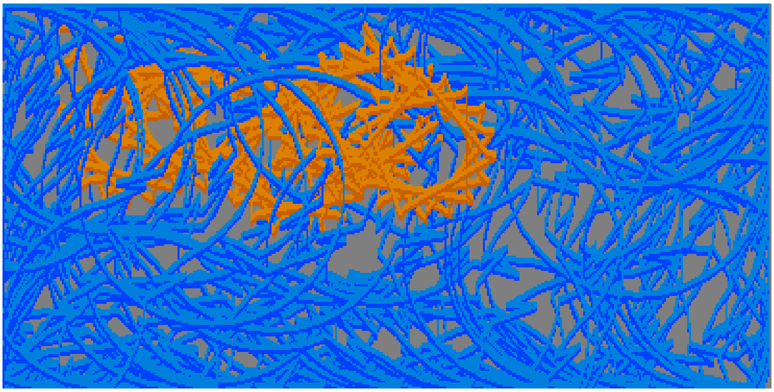

Fig. 1. Two S-robots using different controllers and different painting motifs. Note that one of the S-robots did most of its painting by leaving its pens *down* while executing a back-up and swivel sequence following boundary collisions.

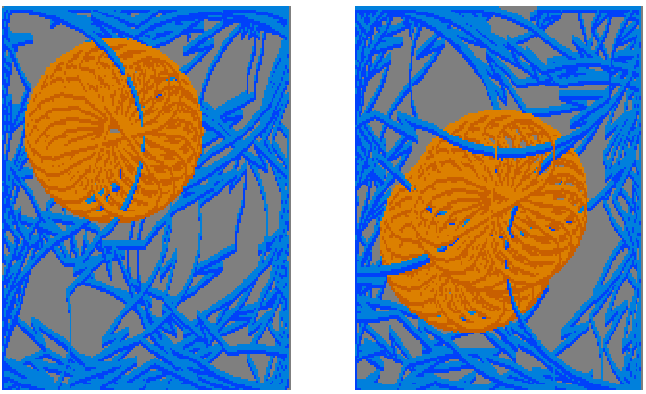

Fig. 2. Images of two S-robots painting with, and without, exhibiting robot interaction. On the left, the S-robot painting the closed figure is oblivious to its companion, while on the right it collides with its companion and gets bumped into a new painting trajectory.

motif. Figure 1 shows an early S-robot test painting made using two S-robots where one used the double hooked curve to draw closed figures and the other used the zigzag sequence as it tried to roam more freely over the canvas. The latter left the pens down during a back-up obstacle avoidance sequence which explains the appearance of the long curving trails.

We now describe our two controllers. Controller A always first checks the forward sensor. If it is clear, it queues the assigned painting command sequence followed by commands to swivel, move a short distance away, and take a proximity reading. If the forward sensor is set, but the backward sensor is clear, it queues a back-up sequence followed by swivel sequence and again requests a proximity reading. Otherwise, having concluded it is boxed in, it swivels and tries to move only a short distance away before taking a new proximity reading. Controller B, on the other hand, can be set up so that it uses the color channel sensors to search either for areas of the canvas that have not yet been painted or for areas that have been painted by one of its companions. Whenever it locates pixels of the type it is searching for, it queues the assigned painting command sequence followed by a swivel sequence, otherwise it swivels and moves a short distance from its present location. In both cases it again queues a color reading. Figure 2 shows what happens when an S-robot with an A controller that is drawing a closed figure gets bumped off course when an S-robot with a B controller that is trying to fill in unpainted canvas gets too close.

5 Evolutionary Framework

The S-robot paintings described below were all painted on 200×200 canvases. The S-robots were permitted to paint for 150,000 clock cycles. The genome for an individual S-robot is the vector (s_x, s_y, d) where (s_x, s_y) is its initial position and d is its initial true compass heading, $-180 \leq d < 180$. For a collection, or swarm, of N S-robots the genome g is the concatenation of the genomes of the individual S-robots. Thus g is a vector with $3N$ components. The point mutation operator applied to g displaces each component of g by a small amount, while the crossover operator applied to genomes from two swarms implements the usual *uniform* crossover operator for two vectors with the same number of components.

Our evolutionary framework uses a population of size $P = 16$. Some evolutionary runs set the number of S-robots at $N = 2$ while others use $N = 4$. For each of $G = 30$ generations, the painting made by the swarm of S-robots with genome g is assigned fitness F_g using one of the calculation methods described below. Then the $P/2$ least fit genomes are discarded and $P/4$ breeding pairs are formed by cloning from the pool of $P/2$ survivors. Breeding within each pair is performed using crossover. Finally all P genomes are subjected to point mutation. Thus an evolutionary run considers $G \cdot P = 30 \cdot 16 = 480$ S-robot paintings. The painting associated with the most fit genome is logged after every five generations. Since point mutation is applied to every genome in the population at the conclusion of every generation, the implicit genetic algorithm is non-elitest and therefore the generation in which the most fit genome will appear during the course of a run cannot be predicted in advance.

6 The S-Robot Fitness Calculation

When a group of N S-robots is making an S-robot painting, the following data is collected: n_p, the number of squares of the grid that were painted; n_b, the number of times an S-robot reacted to the situation where the forward proximity bit was set but the backward proximity bit was clear; n_s, the number of times an S-robot reacted to the situation where the forward and backward bits were both set; and n_c, the number of times an S-robot was successful at color sensing. Figure 3 shows an example of the improvement in image "complexity" that occurred over

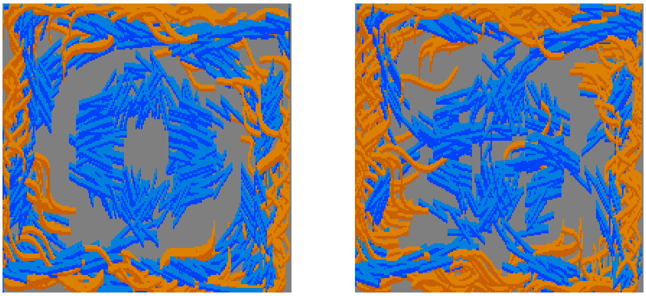

Fig. 3. S-robot paintings where fitness was determined by the two S-robot's ability to cover the canvas. The image on the left is the most fit image in the initial randomly generated population, the image on the right is the most fit image after ten generations.

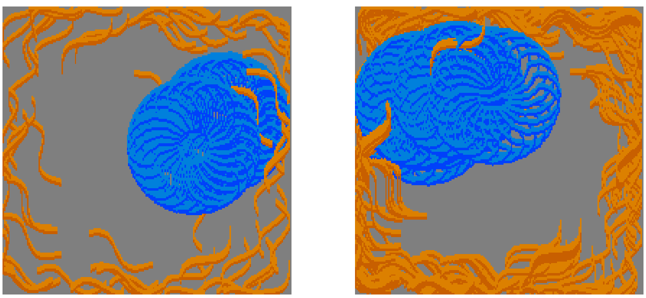

Fig. 4. Two S-robot paintings where image fitness was determined using a linear combination of the S-robot behavioral assessment terms. The goal is to evolve a composition. The image on the left is the most fit image from the original population, the image on the right is the most fit image from the twentieth generation.

time simply by letting $F_g = n_p$, thereby ensuring that the proportion of canvas that was painted was optimized. Figure 3 shows a comparison of the most fit image from the initial randomly generated genome population with the most fit image after ten generations.

Figure 4 shows two S-robot paintings obtained using the fitness function given by $F_g = n_p - n_b + 100n_s + 1000n_c$. Over time evolution causes the canvas to fill in more and locates the closed figure in such a way that maximal S-robot interaction can occur. Figure 5 shows the two most fit S-robot paintings after fifteen and thirty generations from a run using fitness given by $F_g = n_p - 100n_s + 1000n_c$. They show two different "solutions" to the optimization problem posed. One exhibits mutual following behavior by the two S-robots and the other exhibits avoidance behavior since one S-robot retreats to a corner and lingers there. Figure 6 shows the synergy that resulted when the fitness function $F_g = n_s n_c$ was used and the color sensing robot was initialized to seek the paint trails of its companion. Finally, Figure 7 shows an example using fitness function $F_g = n_s n_c + n_p n_b$, which adds a new term to the previous fitness

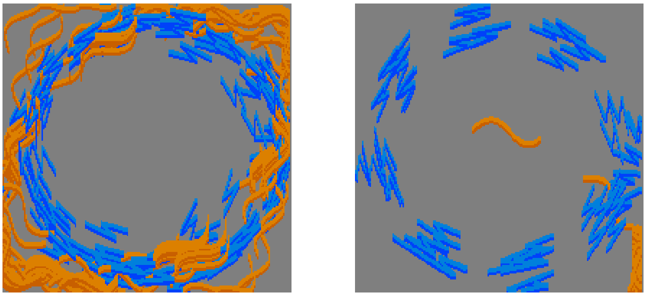

Fig. 5. S-robot paintings from the fifteenth and thirtieth generations obtained from a run that used a fitness function that maximized the assessment terms n_p and n_c, while minimizing n_s

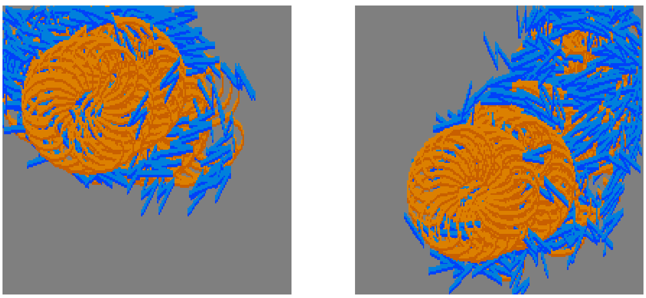

Fig. 6. S-robot paintings from the tenth and twentieth generations obtained from a run that used a fitness function that maximized S-robot interaction by using a product of the behavioral assessment terms n_b and n_c

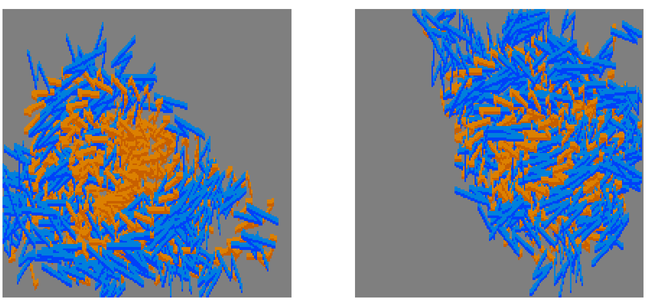

Fig. 7. S-robot paintings from the fifth and twentieth generations obtained from a run that used a fitness function with interaction terms to maximize both S-robot interaction and canvas coverage

function in order to increase canvas coverage in an effort to compensate for the fact that one of the S-robots is now being restricted to making a smaller mark when it paints. We believe these examples help support our contention that it is possible to impose a *style* on robot paintings by carefully devising the fitness functions. That is, following a relatively brief period of experimentation to discover how weighting and combining the parameters affects S-robot paintings, one can become reasonably competent at formulating fitness functions using the parameters in such a way that the evolved imagery will matching one's own aesthetic tastes.

7 On Autonomous Fitness Calculation

The previous section showed how the evolution of our S-robot paintings occurs by using optimization to select the initial configurations of the individual S-robot settings. This optimization treats the fitness calculation as a computation that assigns an aesthetic value to each painting by the swarm of S-robots that created it. Even though it would be a very time consuming process, we feel it is important to make the observation that this fitness calculation could be performed by the S-robots themselves, because we believe that any collection of robots that is engaged in evaluating or recognizing their own creativity would need to include some kind of aesthetic evaluation capability such as ours. Of course, to fully implement the protocol that our S-robots would need to follow in order to achieve this aesthetic evaluation goal, the functionality of the S-robots would need to be enhanced so that they could exchange data with one other, make use of a pseudo random number generator, and have their initial position and heading correctly calibrated. Assuming this were done, an outline of the protocol would be:

1. S-robots save their initial positions and headings.
2. While the painting is being executed, S-robots save information needed to *collectively* calculate image fitness.
3. Designated S-robot traverses entire canvas to determine global statistics needed for fitness calculation (e.g. paint coverage of canvas).
4. S-robots share data in such a way that each is able to calculate image fitness.
5. S-robots compare current fitness value to their saved fitness values to decide, if necessary, which two of *their* saved genomes to cross, before mutating their genomes for the next painting.
6. S-robots are placed on a new canvas with the correct desired initial positions and headings.

It should be clear that it would not be too difficult to design more sophisticated protocols for robot genomes involving controller settings, planning algorithms, or painting sequences in addition to the initial configuration data.

8 Summary and Conclusions

We considered the problem of how to evolve swarm paintings. We did so by developing an evolutionary framework using simulated autonomous mark-making robots. To use the non-interactive genetic algorithm within this framework, we introduced global image assessment parameters and local behavioral assessment parameters that could be used for formulating fitness functions to evaluate, or rank, images on the basis of criteria intended to identify aesthetically interesting paintings. Even though evolution was only able to control the initial placement and positioning of the robots, we gave examples to show how the use of different fitness functions could affect the aesthetic qualities of the robot paintings. We also explained how, in principle, our evolutionary fitness scheme could be managed by the robots themselves. We believe this represents a first step towards reaching the eventual goal of having autonomous robots collectively evaluate and recognize their own creative efforts.

References

1. McCormack, J. (2005): Open problems in evolutionary music and art. In Rothlauf, F. et al (eds.) Applications of Evolutionary Computing, EvoWorkshops 2005 Proceedings, Springer-Verlag Lecture Notes in Computer Science, LNCS 3449, 428–436.
2. Moura, L. and Ramos, V. (2002): Swarm paintings — nonhuman art. In Maubant, J. et al (eds.), Architopia: Book, Art, Architecture, and Science, Institut d'Art Contemporain, Lyon/Villeurbanne, France, 5–24.
3. Moura, L. and Pereira, H. (2004): Man + Robots: Symbiotic Art. Institut d'Art Contemporain, Lyon/Villeurbanne, France.
4. Ramos, V. (2003): Self-organizing the abstract: canvas as a swarm habitat for collective memory, perception and cooperative distributed creativity. In Rekalde, J. et al (eds.) First Art & Science Symposium, Models to Know Reality, Bilbao, Spain, 59.
5. Theraulaz, G. and Bonabeau, E. (1999): A brief history of stigmergy. In Artifical Life **5:2** 97–116.
6. Saunders, R. and Gero, J. (2001): Artificial creativity: a synthetic approach to the subject of creative behavior. In Gero, J. (ed.) Proceedings of the Fifth Conference on Computational and Cognitive Models of Creative Design, Key Centre of Design Computing and Cognition, Sydney, Australia, 113–139.
7. Ramos, V. and Almeida, F. (2000): Artificial ant colonies in digital image habitats - a mass behaviour effect study on pattern recognition. In Dorigo, M. et al (eds.) Proceedings of ANTS'2000 - 2nd International Workshop on Ant Algorithms (From Ant Colonies to Artificial Ants), Brussels, Belgium, 113–116.
8. Ramos, V. (2002): On the implicit and on the artificial - morphogenesis and emergent aesthetics in autonomous collective systems. In Maubant, J. et al (eds.), Architopia: Book, Art, Architecture, and Science, Institut d'Art Contemporain, Lyon/Villeurbanne, France, 25–27.
9. Ramos, V. and Merelo, J. (2002): Self-organized stigmergic document maps: environment as a mechanism for context learning. In Alba, E. et al. (eds.) AEB2002, First Spanish Conference on Evolutionary and Bio-Inspired Algorithms, Mérida, Spain, 284–293.
10. Sims, K. (1991): Artificial evolution for computer graphics. In Computer Graphics **25** 319–328.

11. Aupetit, S., Bordeau, V., Slimane, M., Venturini, G. and Monmarche, N. (2003): Interactive evolution of ant paintings. In McKay, B. et al (eds), 2003 Congress on Evolutionary Computation Proceedings, IEEE Press, 1376–1383.
12. Greenfield, G. (2005): Evolutionary methods for ant colony paintings. In Rothlauf, F. et al (eds.) Applications of Evolutionary Computing, EvoWorkshops 2005 Proceedings, Springer-Verlag Lecture Notes in Computer Science, LNCS 3449, 478–487.
13. Baluja, S., Pomerleau, D. and Jochem, T. (1994): Towards automated artificial evolution for computer-generated images. In Connection Science **6** 325–354.
14. Greenfield, G. (2002): On the co-evolution of evolving expressions. In International Journal of Computational Intelligence and Applications **2:1** 17–31.
15. Machado, P. and Cardoso, A. (1998): Computing aesthetics. In Oleiveira, F. (ed.) Proceedings XIV-th Brazilian Symposium on Artificial Intelligence SBIA'98, Porto Allegre, Brazil, Springer-Verlag, LNAI Series, New York, NY, 219–229.
16. Bentley, K. (2002): Exploring aesthetic pattern formation. In Soddu, C. (ed.), Proceedings of the Fifth International Conference of Generative Art (GA 2002), Milan, Italy. (See http://www.generativeart.com/papersGA2002/20.pdf.)
17. Boden, M. (1994): Agents and creativity. In Communications of the ACM **37:7** 117–121.
18. Boden, M. (1996): The Philosophy of Artificial Life. Oxford University Press, New York, NY.
19. Dorin, A. (2001): Aesthetic fitness and artificial evolution for the selection of imagery from the mythical infinite library. In Keleman, J. and Sosik, P. (eds.) Advances in Artificial Life - ECAL 2001 Proceedings, Springer-Verlag, Berlin, LNAI 2159, 659–668.
20. Ramachandran, V. and Hirstein, W. (1999): The science of art: a neurological theory of aesthetic experience. In Journal of Consciousness Studies **6** 15–52.
21. Whitelaw, M. (2004): Metacreation: Art and Artificial Life. MIT Press, Cambridge, MA.
22. Harlan, R., Levine, D., and McClarigan, S. (2000): The Khepera robot and kRobot class: a platform for introducing robotics in the undergraduate curriculum. Technical Report 4, Bonaventure Undergraduate Robotics Laboratory, St. Bonaventure University, New York.