Avoidance Drawings Evolved Using Virtual Drawing Robots

Gary Greenfield(✉)

University of Richmond, Richmond, VA 23173, USA
ggreenfi@richmond.edu

Abstract. We introduce a generative system for "avoidance drawings", drawings made by virtual drawing robots executing a random walk while simultaneously avoiding the paths of other robots. The random walk method is unique and is based on a curvature controlling scheme initially introduced by Chappell. We design a fitness function for evaluating avoidance drawings and an evolutionary framework for evolving them. This requires us to follow principles we elucidate for simulated evolution where the generative system is highly stochastic in nature. Examples document the evolutionary system's efficacy and success.

1 Introduction

To execute his self-avoiding random walk drawings, Chappell [1] introduces a model for random walks based on curvature. Chappell's point that moves in the plane and executes a self-avoiding random walk can be viewed as virtual drawing robot. Using a variation of Chapelle's random walk model we introduce a generative system that uses a small number of virtual drawing robots that each executes a random walk while simultaneously avoiding all other robot paths. Thus, in effect, n such robots create paths that partition the canvas into n simply-connected regions. Examining examples of these "avoidance drawings" that are generated using several different initial configurations leads to an aesthetic evaluation criterion that can be implemented as a fitness function. However, due to the chaotic nature of the generative system, we are forced to evolve avoidance drawings using only a limited number of the available parameters and adopting a very conservative evolutionary framework. The evolutionary system is successful in evolving novel and interesting avoidance drawings.

This paper is organized as follows. In section two we review self-avoiding random walks. In section three we consider Chappell's model for random walks based on curvature. In section four we provide some background on drawing robots. In section five we give the details of the design of our virtual drawing robot. Section six shows examples of avoidance drawings. In section seven we develop our fitness function and in section eight we describe our evolutionary framework. Section nine discusses the evolved avoidance drawings we obtained while section ten gives our summary and conclusion.

C. Johnson et al. (Eds.): EvoMUSART 2015, LNCS 9027, pp. 78–88, 2015.
DOI: 10.1007/978-3-319-16498-4_8

2 Self-Avoiding Random Walks

Self-avoiding walks arise in the physics and chemistry literature because of their application to protein folding and other polymer-related problems [2]. Usually self-avoiding walks are implemented using lattices [3], and in this case it is known that there are algorithms for infinite self-avoiding walks [4]. In order to generate smooth self-avoiding random walks Chappell introduced a model for random walks in the plane based on curvature [1]. Then, by formulating rules based on readings from sensory apparatus — two sets of "feelers" attached to the point executing the walk — and incorporating a stylized line, Chappell generated drawings of points executing prolonged self-avoiding walks such as the one shown in Fig. 1. More recently, Greenfield used a similar random walk curvature model, but a different sensory system and line stylization method to generate self-avoiding walks resulting in labyrinths such as the one also shown in Fig. 1.

3 Random Walks Based on Curvature

The random walk algorithm based on curvature introduced by Chappell [1] is parametrized by arc length. To cut to the chase, if after the point has traveled a distance s, the point has tangential angle (i.e., direction) $\theta(s)$, curvature $\kappa(s)$ and position $(x(s), y(s))$ then at distance $s+\Delta s$ the behavioral update equations are given by:

$$\begin{aligned}\kappa(s+\Delta s) &= \kappa(s) + \kappa_0 X(s),\\ \theta(s+\Delta s) &= \theta(s) + \kappa(s+\Delta s)\Delta s\end{aligned}$$

where $X(s)$ is a stochastic random variable assuming values $+1$ and -1 and κ_0 is a "small" constant, while the positional update equations are given by:

$$\begin{aligned}x(s+\Delta s) &= x(s) + \cos(\theta(s+\Delta s))\Delta s,\\ y(s+\Delta s) &= y(s) + \sin(\theta(s+\Delta s))\Delta s.\end{aligned}$$

In fact, it is through the use of more elaborate update formulas for θ that Chappell implements his self-avoidance rules, but the details will not concern us here since we will consider a simpler model below.

4 Drawing Robot History

Because they have sensory apparatus, obey rules that act as controllers, and incorporate stylized line drawing methods, it is clear that points executing self-avoiding walks such as the ones described by Chappell and Greenfield can be viewed as virtual drawing robots. If the walks were in 3D rather than 2D, they might also fall within the category of agents in swarms potentially following rules such as the flocking rules of Reynolds [5] or Jacob et al. [6].

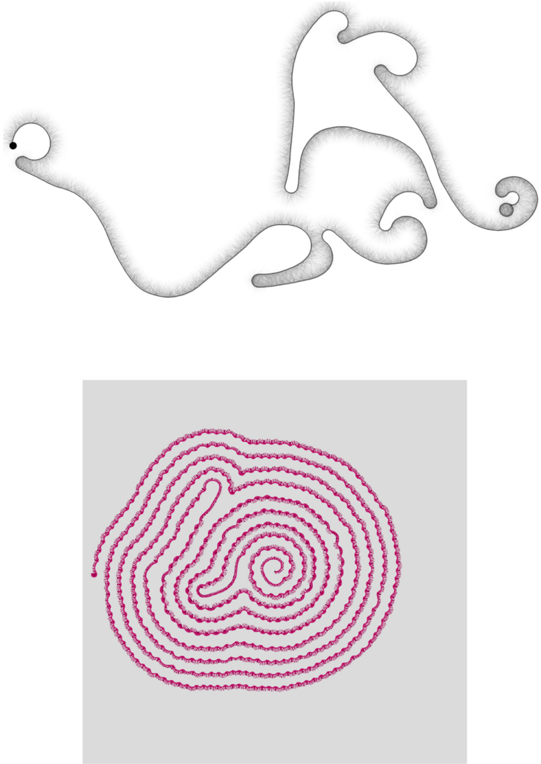

Fig. 1. Top: An example of a point in the plane performing a self-avoiding random walk using Chappell's model. (Copyright 2014 David Chappell. Reprinted from [1] with permission.) Bottom: SA Labyrinth #8352, a self-avoiding random walk using Greenfield's model. (Copyright 2014 Gary Greenfield. Reprinted from http://gallery.bridgesmathart.org/exhibitions/2015-joint-mathematics-meetings/gary-greenfield.)

There is a brief history of drawing robots in evolutionary art. The most famous *physical* drawing robots are undoubtedly those of Moura, Ramos, and Pereira [7,8] who referred to their collective robotic swarm drawings as "non-human art". Bird et al. [9] engaged in a more scientific experiment by attempting to evolve controllers for line drawing primitives for a single robot. Recently, a team led by Monmarché has posted video of experiments conducted with small groups of drawing robots [10]. We assume details of their work will be forthcoming. On the virtual drawing robot front, numerous agent based generative art systems might be viewed as qualifying for virtual robot status. Due to their focus on agents marking and establishing territory two deserve particular attention: Annuziato's system [11] which drew attention in the graphics and artificial life communities at the start of computer generated generative art, and more recently an homage to that work done by McCormack [12]. A series of papers by Greenfield was devoted to virtual robot drawings. His virtual robots were modeled after Khepera robots. He evolved robot paintings using as genomes initial placement configurations [13], evolved encoding tables such that DNA data could be used to drive controllers [14], explored a variation of the robot wherein the pen moved transversely to the robot's straight line motion in order to provide the ability to draw curvilinear lines [15], and evolved programs for his virtual robots in a higher level language modeled after video game controller languages [16].

5 Our Virtual Drawing Robot

With one exception, the behavior for our point mass virtual robot is exclusively controlled by updating κ as its position changes. The exception is an avoidance turn. If the robot is not executing an avoidance turn, then the update equation for κ uses 0.4 for the distance increment Δs, lets $\kappa_0 = 0.04$, and takes $X(s)$ to be zero for seven consecutive updates before letting it be a random number between -1 and $+1$. Over time this smoothness condition on the robot's turning yields a random walk that causes the robot to transition back and forth between a flowing line and a tight spiral. If the robot is executing an avoidance turn, then it does not modify κ at all, so $\kappa(s + \Delta s) = \kappa(s)$, but when it completes the avoidance turn it resets κ using the formula

$$\kappa(s + \Delta s) = -\kappa(s)/4 + \kappa_r/2 + \kappa_0 Y(s)$$

where κ_r is the value saved prior to initiating the turn, Δs and κ_0 are as above, and $Y(s)$ is a random number between -1 and $+1$. The idea is to try to transition from the avoidance turn back to the previously interrupted course.

Avoidance turns are induced from sensing. Sensing occurs after each position update. The robot has two "feelers" extending 17 units from its point mass and located 15° to each side of its forward heading. The feelers can sense canvas boundaries and the paths of other robots. The feelers also transmit the distance to such obstacles. If an obstacle is detected that is distant, then the robot initiates a turning sequence away from it by saving the current value of κ and assigning κ the magnitude 0.15 with sign opposite to that of the saved value. This effectively causes the robot to veer off and initiate a "tight" turn that lasts 25 update steps. If an obstacle is detected that is deemed close, then it takes more drastic action by completely reversing course; that is, it increments θ by $180° \pm 30°$. Note that it does not otherwise cancel any avoidance turn which is in progress. The demarcation between distant and close is 10 units.

The mark making ability of our drawing robot is anti-climactic. It has an assigned color, and after each change in position it deposits a 3×3 pixel splat of that color as well as an identifier so that its path can be sensed by other robots. Note that since it takes multiple updates to traverse a pixel, the use of a splat helps anti-alias the curve the robot draws.

6 Examples of Avoidance Drawings

To explore the type of drawings the virtual robots described above can create, we conducted three experiments using a 600×600 canvas and allowing 400,000 position updates per robot. For all three experiments there were four drawing robots, each assigned a distinct color. By using different seeds for the pseudo random number generator, we generated 40 drawings where (a) the robots were uniformly spaced around a circle all pointing outward (b) the robots were uniformly spaced around a circle all pointing inward and (c) and the robots were uniformly spaced along a horizontal bisector all pointing upward. Figure 2 shows the drawings from the outward pointing and upward pointing experiments that we felt were most aesthetically pleasing. It also reveals how the drawings might be interpreted as the outcome of a competition among the four robots to stake out a simply-connected region of the canvas where they are free to meander in.

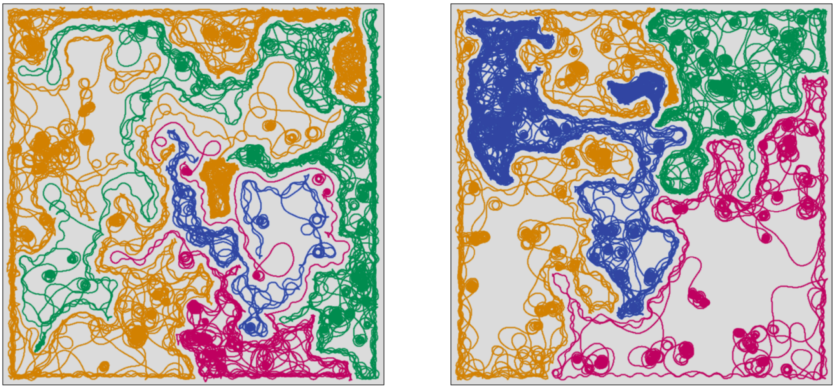

Fig. 2. Left: Avoidance drawing where the four robots were initially uniformly spaced around a circle all pointing outward. Right: Avoidance drawing where the four robots were initially uniformly spaced along a horizontal bisector all pointing upward.

7 The Fitness Function

In considering the 120 drawings we had pseudo randomly generated we came to the realization that the ones we found most interesting were the ones where the robot paths were most clearly delineated and no robot had gotten hemmed in resulting in its path looking like a region that had been flood filled. One crude way to measure the extent to which flood fill has been *avoided* is to consider $N(i)$, the number of adjacent background pixels summed over all pixels of the canvas that have been marked by robot i. The robot with the *smallest* value of N should be one that was most hemmed in. Therefore, if we maximize over the minimum of the N values, we should be able to identify drawings where all four paths are most clearly delineated. This argument suggests assigning drawing D fitness

$$f(D) = \min_i \{N(i)\}.$$

As we shall see, this fitness function does not address aesthetic issues such as region balance or canvas coverage. Be that as it may, even though the problem of designing fitness functions is known to be hard [17], past experience suggests simplistic fitness criteria can often yield positive results.

8 The Evolutionary Framework

The 120 avoidance drawings we generated from our preliminary experiments were ample testimony to the chaotic nature of our generative process and its sensitivity to perturbations and initial conditions. To apply evolutionary techniques we formulated principles to help must constrain the system in order to have some hope of obtaining meaningful results. The rationale for the principles is to yield drawings that arise from an evolutionary process, not random search.

Our first principle is that the fitness landscape be well-defined. To that end, our implementation first sets aside a fixed sequence of pseudo random numbers to present to the robots as the need arises by reserving the first 400,000 pseudo randomly generated numbers from the `lrand48()` generator using seed 121314. Note that using month/day/year format, 12/13/14 is the last sequential date of the 21st century. We then re-seed the pseudo random number generator so that it can oversee the genetics of our evolutionary algorithm.

Our second principle is that the initial configuration is the same for every drawing, save for parameters under evolutionary control. We initialize robots so that their initial headings all point upward i.e., we set $\theta(0) = \pi/2$ and $\kappa(0) = 0$ for each robot. This means that our avoidance drawing is now completely determined by the initial positions of the four robots, whence if robot i has *integral* initial position $(x_i(0), y_i(0))$ where each coordinate lies in the interval $[20, 580]$, there are essentially $560^8 \approx 9.6 \times 10^{21}$ genomes of the form

$$(x_1(0), y_1(0), x_2(0), y_2(0), x_3(0), y_3(0), x_4(0), y_4(0))$$

to consider.

Designing a genetic framework to use for our genomes also presents difficulties. We adopt a very conservative approach which arises as a consequence of our third principle: at least at the genomic level, change must be gradual. In our implementation, a new genome is produced from an old one by cloning it and mutating one randomly selected robot position as follows: 75 % of the time a new position that lies in the 15 × 15 neighborhood of the old position is randomly selected, and 25 % of the time a completely new position is randomly selected. As this portends, our re-population scheme selects and retains the most fit genome from the population of size P and then clones and applies our mutation operator to provide $P-1$ genomes for the next generation. Thus an evolutionary run lasting G generations considers $P + (G-1)(P-1)$ genomes. Clearly this scheme does make evolution slow and gradual and make its dynamics easy to trace and understand. Note that as a bonus the evolutionary process can easily be interrupted and resumed simply by writing out the most fit genome and subsequently reading it back in.

9 Evolved Avoidance Drawings

For the evolutionary runs described here we set the population size $P = 12$ and the number of generations $G = 90$, so that each run considered $12 + (89)(11) = 991$ genomes. A run takes approximately 12 hours. A run usually yields no more than 3 genomes beyond a transient phase lasting, say, 10 generations indicating fitness improvements for the best individual after that stage of evolution typically occur infrequently. This reflects the slow, gradual and careful hill-climbing nature of our evolutionary design. It also reinforces our belief that the constrained fitness landscape is very rugged with local maximums abundant and easily found. Further evidence is obtained by noting that within each generation the population of mutated clones exhibits fitness values that fall off dramatically. Thus, it is not unusual to observe a fairly uniform spread in the 12 individual fitnesses ranging from, for example, 4000 to 24000. We discuss the results from two sample runs.

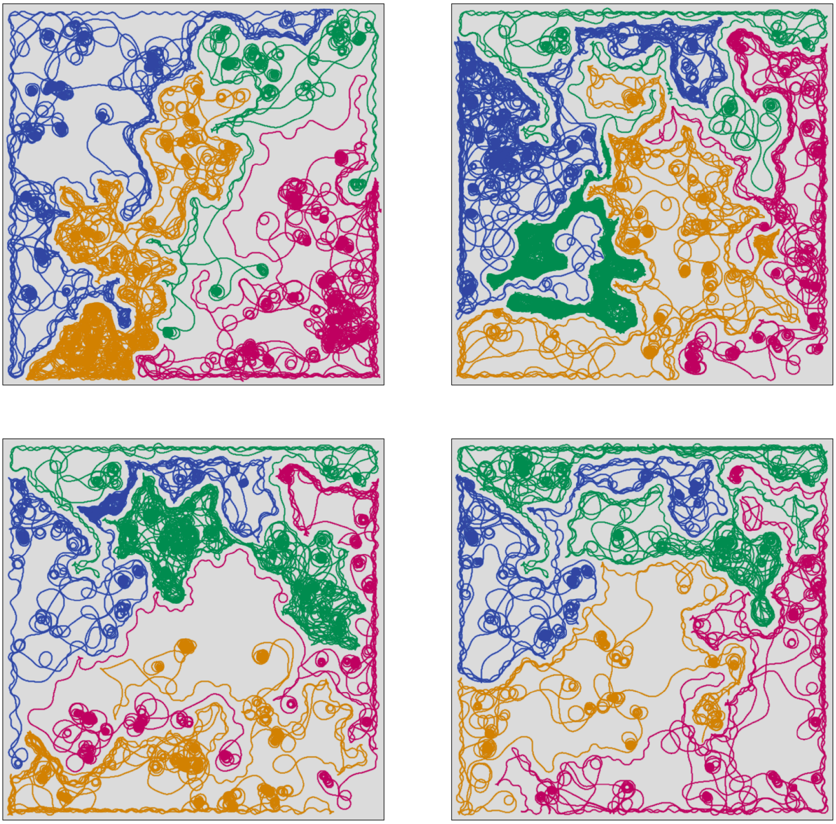

Fig. 3. Clockwise, starting after the most fit individual in the initial population at top left, are the three sequentially mutated improved individuals found at generations 3, 23, and 79 respectively.

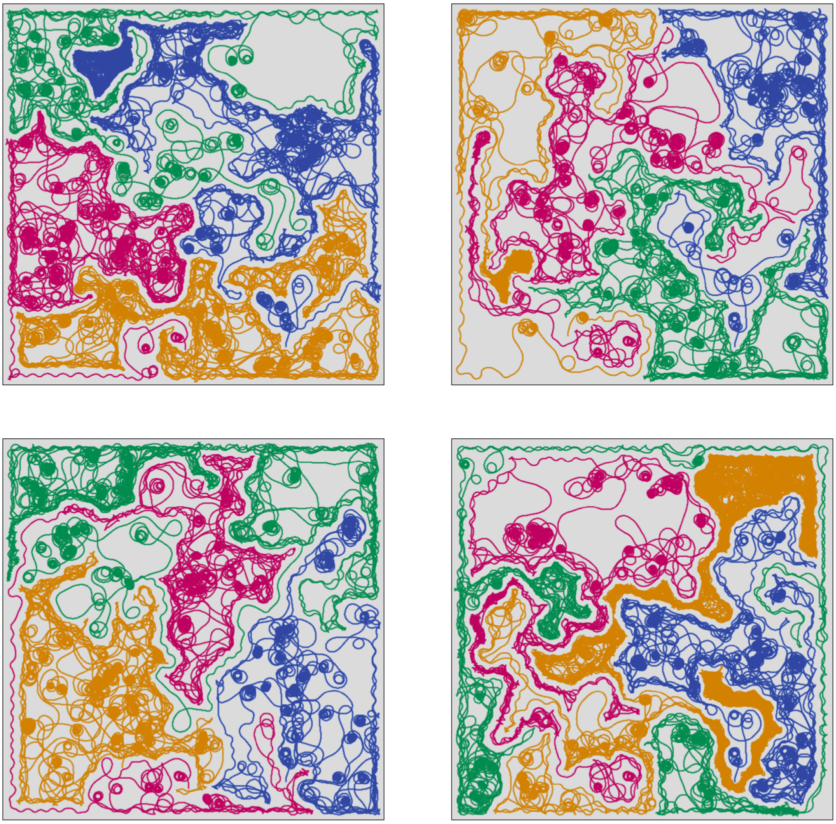

Fig. 4. Clockwise, starting after the most fit individual in the initial population at top left, are the three sequentially mutated improved individuals found at generations 49, 87, and 88 respectively.

Table 1. The most fit genomes from an evolutionary run. Leftmost column shows generation number g, rightmost column shows fitness value f, and column r_i shows the initial position of the i-th robot.

g	r_1	r_2	r_3	r_4	f
0	$(268, 321)$	$(24, 356)$	$(524, 533)$	$(528, 38)$	24372
3	$(268, 321)$	$(24, 356)$	$(517, 537)$	$(528, 38)$	28956
23	$(394, 122)$	$(24, 356)$	$(517, 537)$	$(528, 38)$	29008
79	$(390, 129)$	$(24, 356)$	$(517, 537)$	$(528, 38)$	29444

For the first run, Table 1 shows that the most fit individual in the initial population had fitness 24372 and there were three subsequent improvements occurring at generations 3, 23, and 79 leading to a top fitness score of 29444. Further, these improvements resulted from finding a nearby new starting position

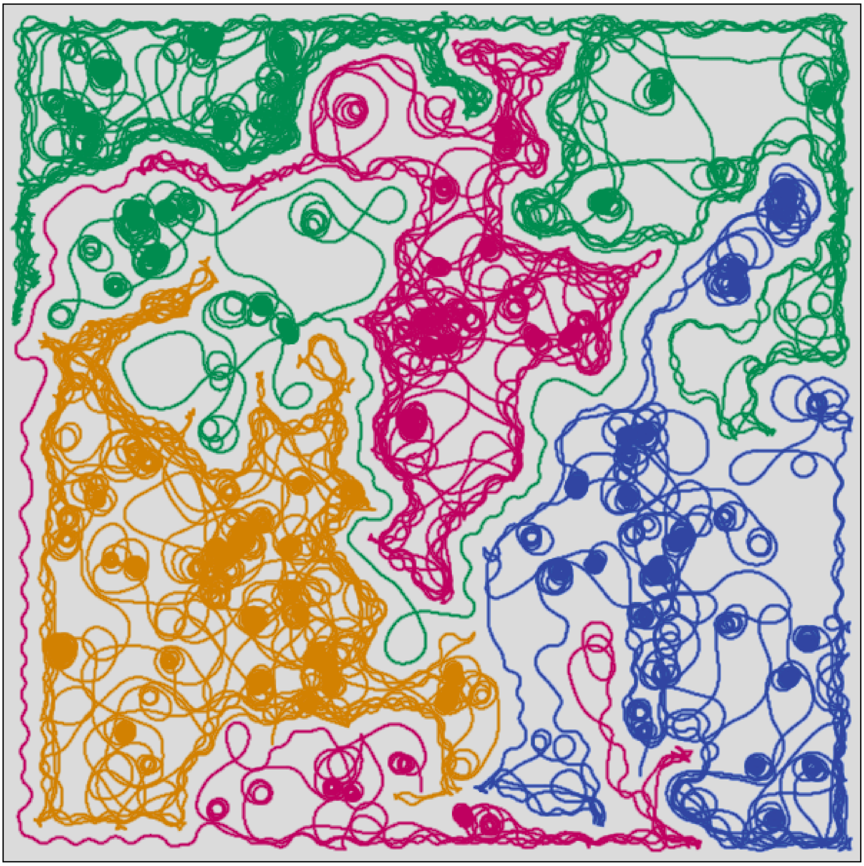

Fig. 5. Enlargement of the bottom left avoidance drawing from the previous figure — the most fit individual we have successfully evolved.

for robot #3 at generation 3, and a completely new position for robot #1 at generations 23 and 79. This means the mutated individual from generation 23 could have been skipped if the one from generation 79 had been encountered earlier. But, in looking at the avoidance drawings (phenomes) for these four individuals shown in Fig. 3, in our opinion that would have been a loss because we feel it was the best of the four. On the other hand, even though the two later drawings greatly reduce the flood fill phenomena found in the two earlier ones, they are both too top heavy with too much open space in the center of the composition.

The second run, again yielding only four genomes, seemed to avoid such problems, and in the next to last generation found the best avoidance drawing we have evolved to date. The four genomes are shown in Table 2 and the phenomes in Fig. 4. This time we observe that first a nearby position for robot #4 was found, then a new position for robot #3 was located, and finally a nearby position for robot #1 was obtained. To better appreciate the evolved result, and to give a sense of the detail found in avoidance drawings, we show an enlargement of the most fit individual from this run in Fig. 5.

Table 2. The most fit genomes from an evolutionary run. Leftmost column shows generation number g, rightmost column shows fitness value f, and column r_i shows the initial position of the i-th robot.

g	r_1	r_2	r_3	r_4	f
0	$(183, 77)$	$(445, 60)$	(456.245)	$(287, 55)$	29144
49	$(183, 77)$	$(445, 60)$	$(456, 245)$	$(292, 49)$	30022
87	$(183, 77)$	$(445, 60)$	$(194, 319)$	$(292, 49)$	30570
88	$(188, 82)$	$(445, 60)$	$(194, 319)$	$(292, 49)$	31544

10 Summary and Conclusion

We introduced a generative system for avoidance drawings, drawings made by virtual drawing robots executing a random walk based on a curvature while simultaneously avoiding the paths of other robots. We designed a fitness function for evaluating such drawings and an evolutionary framework for evolving them. We provided examples to document the system's efficacy. The main contribution of this work is the set of principles we developed to design an evolutionary framework for a generative system that is highly stochastic in nature.

References

1. Chappell, D.: Taking a point for a walk: pattern formation with self-interacting curves. In: Greenfield, G., et al. (eds.) Bridges 2014 Conference Proceedings, pp. 337–340. Tessellations Publishing, Phoenix (2014)
2. Madras, N., Slade, G.: The Self-Avoiding Walk. BirkHauser, Boston (1993)
3. Vanderzande, C.: Lattice Models of Polymers. Cambridge University Press, New York (1998)
4. Kremer, K., Lyklema, J.: Infinitely growing self-avoiding walk. Phys. Rev. Lett. **54**, 267–269 (1985)
5. Reynolds, C.: Flocks, herds, and schools: a distributed behavioral model. Comput. Graph. **21**(4), 25–34 (1987)
6. Jacob, C., Hushlak, G., Boyd, J., Sayles, M., Nuytten, P., Pilat, M.: Swarmart: interactive art from swarm intelligence. Leonardo **40**(3), 248–254 (2007)
7. Moura, L., Pereira, H.: Man + Robots: Symbiotic Art. Institut d'Art Contemporain, Lyon/Villeurbanne (2004)
8. Moura, L., Ramos, V.: Swarm paintings – nonhuman art. In: Maubant, J., et al. (eds.) Architopia: Book, Art, Architecture, and Science, pp. 5–24. Institut d'Art Contemporain, Lyon/Villeurbanne (2002)
9. Bird, J., Husbands, P., Perris, M., Bigge, B., Brown, P.: Implicit fitness functions for evolving a drawing robot. In: Giacobini, M., et al. (eds.) EvoWorkshops 2008. LNCS, vol. 4974, pp. 473–478. Springer, Heidelberg (2008)
10. Monmarché, M. et al., 2014. http://youtu.be/GrxthHngARU
11. Annunziato, M.: The Nagual experiment. In: Soddu, C., (ed.) Proceedings 1998 International Conference on Generative Art, pp. 241–251 (1998)

12. McCormack, J.: Creative ecosystems. In: McCormack, J., d'Inverno, M. (eds.) Computers and Creativity, pp. 39–60. Springer, Heidelberg (2012)
13. Greenfield, G.: Robot paintings evolved using simulated robots. In: Rothlauf, F., et al. (eds.) EvoWorkshops 2006. LNCS, vol. 3907, pp. 611–621. Springer, Heidelberg (2006)
14. Greenfield, G.: Evolved look-up tables for simulated DNA controlled robots. In: Li, X., et al. (eds.) SEAL 2008. LNCS, vol. 5361, pp. 51–60. Springer, Heidelberg (2008)
15. Greenfield, G.: On simulating drawing robots with straight line motion but curvilinear pen paths. In: Roeschel, O., Santos, E., Yamaguchi, Y., (eds.) 14th International Conference on Geometry and Graphics, International Society for Computer Graphics, Conference DVD (2010)
16. Greenfield, G.: A platform for evolving controllers for simulated drawing robots. In: Machado, P., Romero, J., Carballal, A. (eds.) EvoMUSART 2012. LNCS, vol. 7247, pp. 108–116. Springer, Heidelberg (2012)
17. McCormack, J.: Open problems in evolutionary music and art. In: Rothlauf, F., et al. (eds.) EvoWorkshops 2005. LNCS, vol. 3449, pp. 428–436. Springer, Heidelberg (2005)