Niche Constructing Drawing Robots

Jon McCormack(✉)

sensiLab, Faculty of Information Technology,
Monash University, Caulfield East, Australia
Jon.McCormack@monash.edu
http://jonmccormack.info

Abstract. This paper describes a series of experiments in creating autonomous drawing robots that generate aesthetically interesting and engaging drawings. Based on a previous method for multiple software agents that mimic the biological process of niche construction, the challenge in this project was to re-interpret the implementation of a set of evolving software agents into a physical robotic system. In this new robotic system, individual robots try to reinforce a particular niche defined by the density of the lines drawn underneath them. The paper also outlines the role of environmental interactions in determining the style of drawing produced.

1 Introduction

The idea of autonomous machine creativity has occupied human thought at least since the ancient Greeks, although the concept of creativity and its interpretation has changed significantly [25]. In more recent times, the field of computational creativity has sought a scientific understanding of various aspects of human creation and creativity, and a major goal of this discipline is to create autonomous creative artists [8]. However, the emphasis – and the standard benchmark – is human creativity, most often through the lens of classical and modernist art or established genres of design or music [19].

The research described here investigates the concept of autonomous machine creativity through the design and implementation of autonomous drawing robots. In these experiments, the motivation and inspiration is not the mimicry or replication of human drawing or human creativity.[1] The investigations described are motivated by a desire to understand *poesies* in a post-anthropocentric sense [24], one that considers creative expression more broadly than the examples exhibited by humans and human cultures. In this sense, human and machine creativity may be seen as synergistic or complementary, contributing to an on-going dialogue on the fundamentals of creativity, and expanding how we consider creativity in a post-anthropocentric worldview.

---

[1] Although of course humans may appreciate the aesthetics and creativity too!.

2   Background

The research described here relates to a class of drawing machines, known as *autonomous drawing robots* (often referred to as "Drawbots"), where one or more autonomous robots interact to create a drawing. Typically the resultant drawings are abstract, created through a pattern-making process rather than figurative representations drawn from captured images, although such distinctions normally represent a continuum of possibilities rather than a binary division. In some cases, many individual robots are involved in the generation of a single drawing. Much like the swarm [15] and multi-agent or individual-based modelling systems [14], these systems draw on the emergent interactions between multiple robotic agents and their environment.

Also important is the *performative* nature of autonomous, artistic drawing systems. People are fascinated by machines that appear to show intent, agency and autonomy – even if it is often alien or ambiguous. Drawbots play on this fascination to produce artefacts that provide a visual record of that agency and autonomy. Unlike their human counterparts robots are always eager and ready to perform: they do not need to wait until "the right moment" or search for inspiration in time or from place in order to create. Hence, many systems are readily exhibited "in the act of drawing", rather than exhibiting only the finished drawings they have created, as is the norm with conventional human artists.[2]

2.1   Related Work

The artist Jean Tinguely developed a series of autonomous drawing machines ("Méta-Matics"), first exhibited in Paris in 1959 [9]. Through these machines Tinguely deliberately expressed an ironic distance "from the abstract-gestural painting of his contemporaries". The visual form and kinetic behaviour of these machines fascinated audiences, explicitly prompting the question of the automation of artists and artistic processes.

A extensive visual survey of drawing machines can be found in the *Drawing Machines Vimeo Channel*, moderated by Gary Warner (https://vimeo.com/channels/drawingmachines). Drawing machines are "autonomous machines and devices that make purposeful marks on surfaces, and leave drawings in the world as evidence of their agency" and while being a broader categorisation than autonomous drawing robots, capture much of the essence of human fascination with machine autonomy, intent and creativity.

Perhaps the most sustained study of modern autonomous artistic drawing robots can be found in the work of Leonel Moura, who has been developing drawing robots since 2001 [20,21]. Many of Moura's works draw upon concepts from biology, situated awareness, collective self-organisation and stigmergy found in

---

[2] This fascination does extend to conventional human artists using computers as well – recent David Hockney exhibitions have included numerous works painted using iPhone and iPad apps that are exhibited showing the paint strokes the artist made in real time, revealing the artist's drawing process as it occurred.

the collective behaviour of ants and other social insects. These robots use infrared (IR) and colour sensors to detect the marks locally and feed them autonomously.

The *Drawbots* system [2] was an investigation into autonomous drawing using evolutionary robotics as a way to remove "the personal signature of the artist" [3, Chap. 6]. In this work "implicit" fitness measures were defined that did not restrict or fix the type of marks the robot drawer should make, including an "ecological model" involving interaction between environment resource acquisition and expenditure through drawing. However, the results demonstrated only minimal creativity, and the authors concluded that fitness functions that embodied "artistic knowledge about 'aesthetically pleasing' line patterns" would be necessary if the robot were to make drawings worthy of exhibition.

The work described in this paper has its origins in software-based agent simulation, described in [17]. Greenfield has worked extensively with virtual drawing robots, principally modelled on Khepera robots [11,13]. In [12] he describes a series of experiments for autonomous "avoidance drawings" whereby virtual drawing robots generate complex patterns by generating random walks that avoiding the lines already drawn by themselves and other robots. This concept of self-avoiding walks has also been explored by Chappell, who described pattern-formation processes using self-interacting curves [5] and spatially rhythmic parametric curves described by Whewell equations that simulate the meandering of bends and rivers [6].

What many of these systems share in common is the idea of avoiding interactions with other lines, or stopping drawing if a line intersects with an existing line. Such rules give the drawings structure and assist in the perception of intent in robot behaviour.

2.2   Niche Construction

Niche construction in biology is the process whereby organisms modify or influence their local, heritable environment. Proponents argue for its importance in more fully understanding the feedback dynamics of evolutionary processes in nature [22]. Niche construction processes determine feedback systems that can modify the dynamics of the evolutionary process, because ecological and genetic inheritance co-influence the evolutionary process. Computational models of niche construction show that it can influence the inertia and momentum of evolution and introduce or eliminate polymorphisms in different environments [7].

In previous work, niche construction was employed to enhance the diversity and variation in a virtual autonomous drawing system (Fig. 1) [17]. In this work, a collection of software drawing agents were randomly placed on a blank canvas. Each agent consisted of a six-allele genome that determines the agent's behaviour. Individual alleles – normalised real numbers – determine drawing style (rate of curvature, propensity to meander), rate of fecundity, mortality and, importantly, a local preference for drawing density. As the agents move around the canvas, they leave a trail of ink. If an agent intersects with a line drawn by another agent or itself, it dies. At any time a line-drawing agent may spawn an offspring, with the resultant daughter agent's genome a mutated version of the
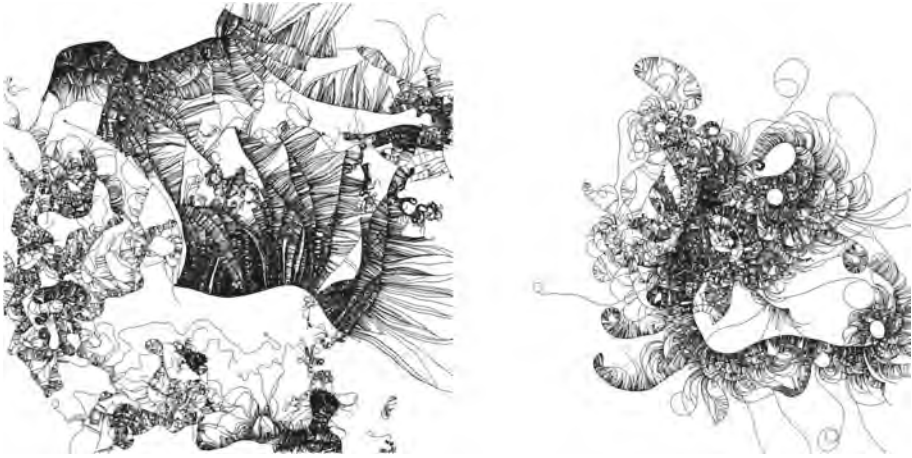
Fig. 1. Two sample outputs from the software niche constructing drawing system.

parent. Mutation is performed with a probability of $\frac{1}{L}$, where L is the genome length. So-called "creep mutation" is used, whereby a normally distributed random number with mean 0 is added to an allele selected for mutation.

The allele for local drawing density preference is used to determine what constitutes an acceptable "niche" for the agent. One can think of this as being an ideal value at which the agent is most "happy". The further away from this ideal value the less happy the agent is and the less likely it is to reproduce.[3] Even greater deviations from the preferred value may effect survival, as shown in Fig. 2.
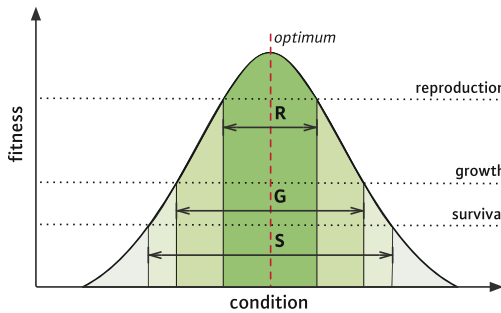

Fig. 2. An idealised agent viability curve for a single dimensional environmental condition (redrawn from [1]).

---

[3] The propensity to reproduce is also determined by a separate allele in the genome, which effectively controls the density of offspring lines created by the agent.
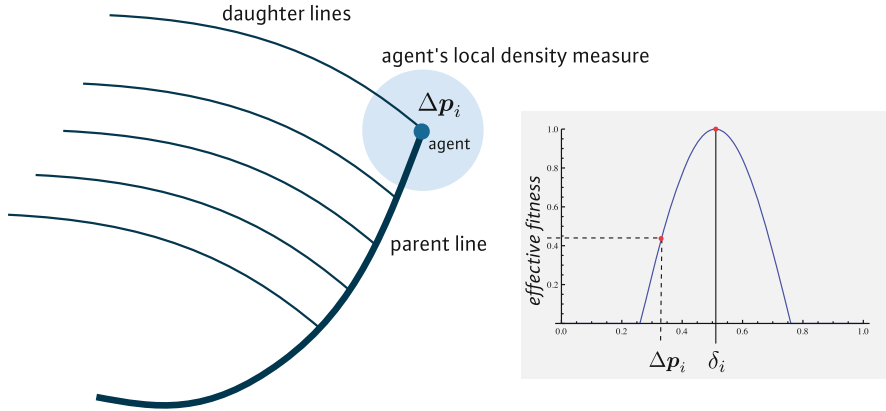
Fig. 3. The niche construction mechanism for drawing agents, who try to construct a niche of local density that satisfies their genetic preference.

For each agent, i, δ_i defines its preferred niche. Local density, defined as the ratio of inked to blank canvas per unit area, is measured over a small area surrounding the agent at each time step (Fig. 3). Proximity to the preferred niche determines the probability of reproduction, given by

$$Pr(rep) = f_i \cdot \cos^\omega(\texttt{clip}(2\pi(\Delta_{\boldsymbol{p_i}} - \delta_i)), -\frac{\pi}{2}, \frac{\pi}{2}), \qquad (1)$$

where $\Delta_{\boldsymbol{p_i}}$ is the local density around the point $\boldsymbol{p_i}$, the agent's position, ω a global parameter that varies the effective niche width, f_i is the agent's fecundity and $\texttt{clip}$ is a function that limits the first argument to the range specified by the next two. Being in a non-preferred niche similarly increases the probability of death.

3   Robot Design

Reinterpreting a software simulation as physical robot designs involves many constraints and challenges. The aim of this initial research was to use the niche construction method to control physical robot behaviour. More advanced aspects of the software simulation, such as self-reproduction were removed from the experiments reported here.

For the initial experiments we used a collection of modified Pololu *m3pi* robots.[4] The m3pi combines a *3pi* robot base with an expansion board that sits above the base (Fig. 4). The base board uses an Atmel ATmega328P microcontroller running at 20 MHz. This microcontroller handles low-level functionality such as driving the motors, poling the QTR-RC reflectance sensors (used to detect lines and line density, see Sect. 3.2 below) and writing to a small 8×2 character LCD display, used for diagnostic messages and debugging.

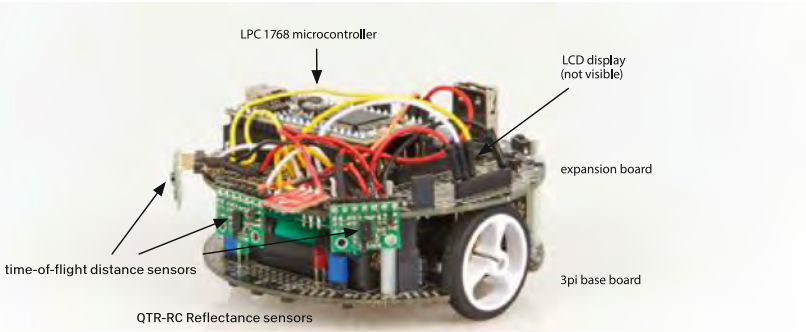[4] https://www.pololu.com/product/2150.

Fig. 4. The modified Pololu m3pi drawing robot (shown without pen).

The expansion board houses a significantly more powerful microcontroller (an 80 MHz ARM mbed NXP LPC1768). This microcontroller communicates with the base board via a serial protocol, commands include reading the reflectance sensors, writing to the LCD and driving the motors at different speeds (both forwards and backwards). Additionally, this microcontroller manages a series of three distance sensors that calculate the distance between the robot and any obstacles. The mbed system[5] supports a full C++ development environment, making it ideal for more complex robotic programming tasks.

Each robot is powered by four rechargeable NiCad batteries and will run continuously for up to 35 min on a full charge. While automated charging was investigated (where upon reaching a reserve power condition the robot seeks out the battery charger and automatically charges itself) for the experiments reported here we relied on manual charging for reliability purposes. A lasercut mounting stage allows mounting of a marker pen at the front of the robot. In this configuration, the pen is in permanent contact with the paper, so is always drawing (i.e. it is not possible for the robot to not draw when moving in this configuration).

3.1 Navigation and Collision Avoidance

Commands control the speed of the two geared motors that turn the wheels on the left and right side of the robot. The software allows continuous control of left and right motor speed independently, and in both directions (a differentially steered drive system). This independent control allows the robot to turn according to the differential speed between the left and right motors [16]. The robot has no turn counter on the wheels, nor IMU or Compass, so it has no accurate sense of its exact physical location within the drawing area. Three time-of-flight distance sensors provide real-time information on the proximity of the robot to

---

[5] See http://www.mbed.com for more details.

obstacles and allow it to react accordingly to avoid them. The sensors have an effective range of approximately 15 cm. A distance-proportionate Virtual Force Field (VFF) method [4] is used to avoid walls and other obstacles. Typically, the robots are set up in a rectangular "drawing arena" that houses a large paper floor surrounding by a low-height wall to stop the robots escaping. The obstacle avoidance algorithm successfully allows the robot to avoid bumping into the surrounding walls, other robots and obstacles.

When the obstacle avoidance task is not in effect (i.e. when the robot is not in close proximity to any obstacles), the robot is free to roam according to a meandering algorithm, similar to that used for the virtual niche constructing agents described in [17], but with modification (explained in Sect. 3.2). In the software version, a combination of alleles determined the agent's basic motion, based on constant rate of curvature and a degree of "irrationality".

For the robot implementation described here, a new movement algorithm was devised, based on a weighted sum of line following and meandering, determined by the suitability of the current niche to the individual's preference.

3.2   Niche Construction Implementation

As detailed in Sect. 2.2, the niche construction drawing algorithm revolves around an agent's local density preference δ_i measured against the actual line density in the immediate area of the robot $\Delta_{\boldsymbol{p_i}}$. To estimate the local density, each robot is fitted with a $5 \times$ QTR-RC reflectance sensor array, which measures the reflectance level directly below each sensor from an infrared light integrated into the sensor. The sensors are arranged around the front edge of the base board and cover a distance of approximately 6 cm. To calculate the local density the average reading across all sensors is read into a ring buffer and the buffer is updated proportionally to the forward speed of the robot (Fig. 5). Hence the average value of all values currently stored in the ring buffer gives the local density in the area directly passed under the robot. We also experimented with centre-weightings for the reflectance sensor array, which did not result in better perception of density.
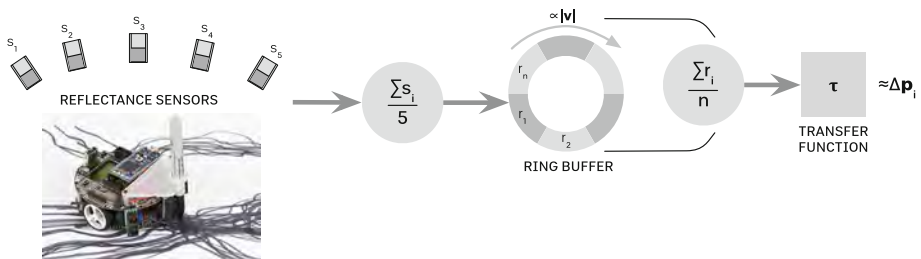

Fig. 5. Calculation of local density.

Fig. 6. Perceptual density samples used for calibration and calculation of the transfer function.

To map the reflectance sensor readings to a perceptual reading of line density a simple calibration sheet with five sample areas of pen-drawn lines of increasingly higher density was designed (Fig. 6). A computer-calculated density match of random lines drawn over a 6 cm × 6 cm area was used to create each sample. The test samples vary in density from normalised values of 0.1–0.9. The robot was moved directly over each sample and the raw sensor values were recorded, along with the cumulative averages while moving over the area of constant density. The calibration was repeated several times at different angles for each sample to remove any directional bias in the readings. These values were used to build a transfer function, τ, that maps raw sensor readings to a normalised perceptual measure of line density. The transfer function is applied to the readings obtained from the raw readings accumulated in the ring buffer to give a normalised reading of the local density of lines in the robot's immediate vicinity. This corresponds to Δ_{p_i} as detailed in Eq. 1.

Responding to a "niche". In the original niche constructing work, the preference of an agent to its niche was determined by an individual allele. The closer the agent to its preferred niche, the higher the probability of reproduction, which daughter lines spawned from the parent agent as it draws along the canvas. In the robotic version self-reproduction was not used, so the emphasis was placed on the individual agent *constructing* its niche. To do this, a density preference was set for each robot via a small potentiometer installed at the base of the robot. Changing the potentiometer allowed easy experimentation with different density preferences without the need to self-reproduce or reprogram the robot. The LCD display displaying the current density preference as a normalised value.

Once a density preference was set, it remained constant for the entire charge of the robot, in simple terms the metaphorical equivalent to an agent's lifespan in the software simulation. The algorithm used to determine the robot's current behaviour is outlined in Algorithm 1.

The variable μ (line 9) represents a measure of the robot's current fitness: how close it is to its preferred niche, analogous to the similar function shown in Eq. 1. μ is used to determine the robot's behaviour: if in a preferred niche, the robot will reinforce that niche by following existing lines where possible (following an existing line draws over the top without a lot of change to the local density). The proximity to the preferred niche determines how closely the robot follows the line, as expressed in the **lineFollow**(μ) function (line 11).

Algorithm 1. Robot Niche Constructing algorithm

1: $\epsilon \leftarrow 0.2$ ▷ Transition range for preferred niche
2: $\delta \leftarrow \text{getDensityPreference}()$ ▷ Read potentiometer value from base of robot
3: **while** batteryHasCharge() **do**
4:      $[s_1 \ldots s_n] \leftarrow \text{readDistanceSensors}()$ ▷ read ToF distances into $s_1 \ldots s_n$
5:      **if** objectDetected($\boldsymbol{s}$) **then**   ▷ check if an object detected by $s_1 \ldots s_n$
6:          steerToAvoid($\boldsymbol{\min(s)}$)   ▷ Avoid collisions
7:      **else**
8:          $\Delta_{\boldsymbol{p_i}} \leftarrow \text{getLocalDensity}()$ ▷ Read local density via transfer function
9:          $\mu \leftarrow \cos^{0.5}(2\pi(\Delta_{\boldsymbol{p_i}} - \delta))$ ▷ μ represents niche proximity
10:          **if** $\mu \geq (1 - \epsilon)$ **then**   ▷ in the required niche – so reinforce this niche
11:              lineFollow(μ)
12:          **else if** $\Delta_{\boldsymbol{p_i}} < \delta$ **then**   ▷ not enough lines in this area
13:              steerInDirection($\nabla\boldsymbol{s}$)   ▷ turn in the direction of increasing density
14:              meander($\boldsymbol{p_i}, \mu$)   ▷ try to find area of closer density match
15:          **else**   ▷ too much density in this area, steer to avoid
16:              steerInDirection($-\nabla\boldsymbol{s}$)   ▷ turn in the direction of decreasing density
17:              meander($\boldsymbol{p_i}, \mu$)   ▷ try to find area of closer density match
18:          **end if**
19:      **end if**
20: **end while**

The **meander()** function is based on a similar function to that developed for the software agents. In the original software agents [17], two alleles determined the agent's constant rate of curvature (σ) and "irrationality" (r). These values were used to calculate the rate of change of angular velocity:

$$\nabla\theta = \sigma + \texttt{fracSum}(\boldsymbol{p}, k \cdot r)^{0.89r^2}, \tag{2}$$

where $\boldsymbol{p}$ is the agent's current position, k a constant known as the *octave factor*, and $\texttt{fractSum}$ a function that sums octaves of 2D noise [23]. For the robots, a similar method was used, based on an estimate of the robots relative position. The position vector $\boldsymbol{p}$ is calculated by estimation. To perform this estimate it is assumed that the robot's starting position and orientation – wherever that may be – defines the origin and direction of an orthogonal, 2D coordinate system, with the robot initially heading in the positive x direction (i.e. in the direction of the vector $[1, 0]$).

The current position, $\boldsymbol{p}$, is calculated using Euler integration, based on the estimated velocity vector, $\boldsymbol{v}$, derived from the differential speed of the left and right motors [16]. The **meander()** function takes two arguments: the robot's current postion, $\boldsymbol{p_i}$, and a degree of irrationality, r, a normalised value. In the algorithm shown, the irrationality (how much the agent meanders) is determined by the proximity to a preferred niche: the further away from the preferred niche, the higher the irrationality, hence the more the robot is likely to twist and turn, writhing to try and find a more preferred niche.

4 Results

The niche constructing drawing robots were exhibited in 2015 as part of the group exhibition ARTE@IJCAI, held at the Centre Cultural Borges in Buenos Aires, Argentina. The exhibition, curated by Luc Steels, highlighted the role of AI and machine creativity in art, and was a companion event to the International Joint Conference on Artificial Intelligence (IJCAI 2015).

In this exhibition, the robots were placed within a $4\,\text{m} \times 4\,\text{m}$ drawing arena and left to draw with a fully charged battery (Fig. 7). The large size of the drawing area allowed the robots to roam freely without the need to constantly avoid the edges of the arena.

Drawings typically took several hours to complete, which required charging the robots several times during the course of the day. During recharges the density preference was changed in some of the drawings to achieve an overall variation in the drawing "style" of each completed drawing. Once a drawing was complete, it was hung on the gallery wall and a new drawing started. Over the course of the exhibition seven different large-scale drawings were completed.

Generally, people were fascinated by the drawing process as much as the resultant drawings. They described intent and autonomy in interviews carried out while the work was running, ascribing purposeful behaviours to the robots as they drew. Often people would place their hands or other objects into the drawing arena, causing any robot nearby to go scurrying away.

Changing the density preference had significant effects on the resultant drawings, with values in the middle range proving the most visually satisfactory. As the robot is unable to raise the pen from the canvas, it is always drawing, hence, always increasing density. However, after a time of drawing over an existing line, the density does not increase. This forces the robot into a conundrum: for a low density preference one should just keep drawing over existing lines (line following), but that places the robot typically over areas of high density that seeks to avoid. Low density preference robots tend to be "explorers", always trying to move away from areas where lines exist. Middle to high density preference robots tended towards being "reinforcers", endlessly drawing around existing lines in increasingly complex patterns, reminiscent of insect trails (Figs. 7 and 9).

As with the original software agents, "founder lines" (those drawn first) have significant impact on the resultant drawing, particularly when the robot agents seek to reinforce existing lines, since these become the main pathways that are reinforced by drawing lines around them or by line following. Of equal importance to the final results as each robot's programmed behaviour was its interaction with the environment. As already mentioned, people would interact with the robots, placing their hands in their pathway, or try to catch them (thankfully difficult given that the robot can move at up to $1\,\text{m/s}$). Moreover, inconstancies in the paper surface would often cause the robot to change course due to slippage or the pen getting caught up on the edges of the paper.

These kind of interactions point to the importance of the environment in determining overall behaviour, furthering the previously introduced concept of creative ecosystems [18]. In subsequent performances of the robots we
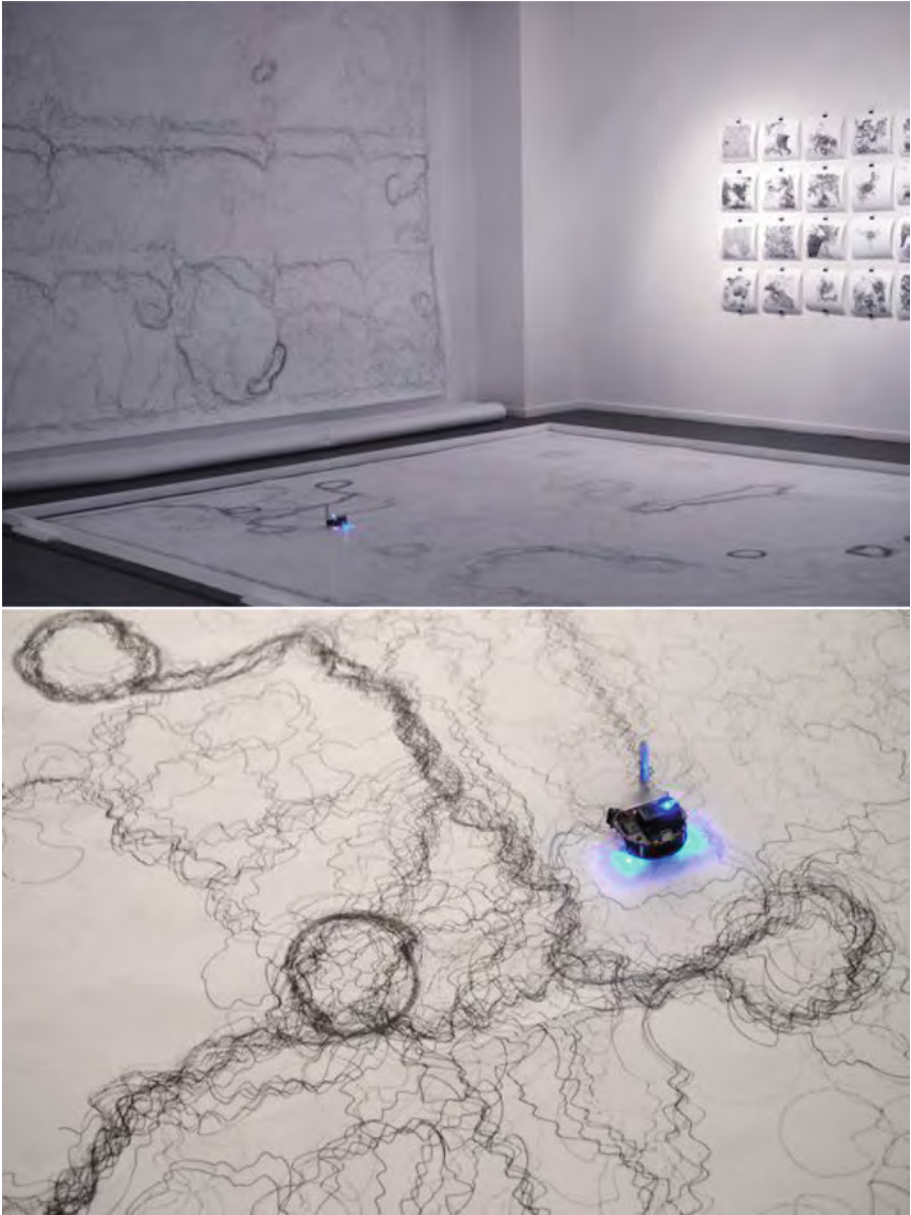
Fig. 7. Exhibition at ARTE@IJCAI in Argentina. The top image shows the drawing arena and a previously drawn image hanging on the wall. The bottom image shows detail of an image drawn by a robot with a niche density preference of 0.7. Notice how the robot has reinforced the niche by repeated drawing over earlier patterns.

Fig. 8. A "rock garden" version of the drawing arena (left) and the resultant drawing (right). Photo curtesy of Gary Warner.
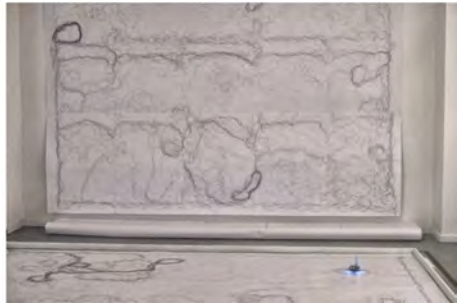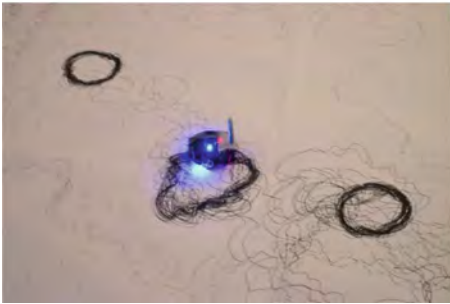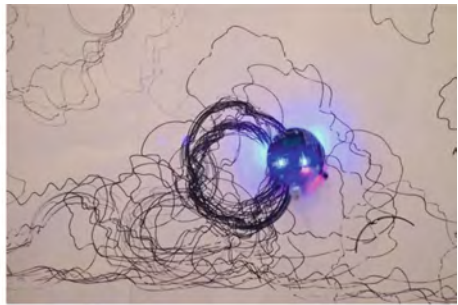

Fig. 9. Various stages of constructing a "niche" of high density.

experimented with placing rocks inside the drawing area, similar to that of the *kare-sansui* dry landscape zen rock gardens in Japan. As expected, the nature of the drawings reflected the interaction of the robots and their environment, finding niches in between rocks in spaces large enough that allowed reinforcement of a specific niche preference (Fig. 8).

5   Conclusions and Future Work

From the initial experiments reported here, it is clear that a number of interesting directions for further research present themselves. Some experimentation is in progress with a directional light sensor so a swarm of robots can use a light source as a navigational aid (in a similar way to some organisms use various tropisms). In the implementation described here, there is little robot-robot interaction beyond collision avoidance. We have begun to experiment with more sophisticated forms of robot-robot and robot-environment interactions. These robots use coloured light and audible sounds that the robots emit and react to (Fig. 10), creating the possibility of a sonic niche. Specific sounds or light frequencies can be used to identify groups – all members of a particular group use the same frequency signature – with different groups responding in different ways to the proximity of outsiders.

A behaviour currently under investigation is inspired by insect behaviour, specifically the defensive behaviour of the African bombardier beetle (*Stenaptinus insignis*) [10]. These beetles emit a hot, acrid spray when attacked. The bombardier is particularly intriguing as it is able to precisely direct its spray in the direction of the attacker. Inspired by these natural wonders, we have designed a number of insect-like robots that squirt different coloured inks,

Fig. 10. Further designs for the niche constructing drawing robots, developed by Nick Jones, based on the insect species *Stenaptinus insignis*. These insects fire a toxic acid spray when attacked. The microphone, which responds to sounds, can be seen in the middle of the front of the robot. These robots are currently under development.

Table 1. Comparison of software agent and physical robot implementations.

Feature	Software virtual agents	Physical robots
Emphasis	Collective behaviour	Individual drawing behaviour
Niche mechanism	Line density	Line density
Reproduction?	Yes	No
Behavioural mechanism	Genome	Decision tree
Behaviour change	Evolution	Environment
Drawing style	Complex structure	Niche reinforcement

as shown in the figure. In addition to microphones and small speakers, these robots have colour sensors underneath, allowing the possibility of more complex, colour-dependent niche construction. Experiments with these new, more complex niche constructing robots are currently ongoing.

In conclusion, in this paper we have described how the original, software-based niche constructing agent model can be interpreted in real physical robots. A summary of the major differences is outlined in Table 1. Obviously features such as self-reproduction, and hence evolution, are difficult to achieve in a physical robotic system, so as a first step, we have focused on the role of niche preference in the individual. Even with this limitation, the resultant drawing and robot behaviour show interesting results. While the style of the drawings is quite different between the software and physical robot versions, it is easy to see the important role that niche construction plays in determining the overall features of the drawings produced. In this respect there are similarities in the way niche construction provokes interesting and unexpected regions of organisation by trying to build niches of suitable density in both the software and physical robot versions.

Our new experiments that broaden the conditions under which the robots seek to niche construct (e.g. sound and colour), suggest that this method is a rich and useful technique for generating autonomous creative behaviour in artificial systems.

References

1. Begon, M., Townsend, C., Harper, J.: Ecology: from individuals to ecosystems. Wiley-Blackwell, Maldon (2006)
2. Bird, J., Husbands, P., Perris, M., Bigge, B., Brown, P.: Implicit fitness functions for evolving a drawing robot. In: Giacobini, M. (ed.) EvoWorkshops 2008. LNCS, vol. 4974, pp. 473–478. Springer, Heidelberg (2008). doi:10.1007/978-3-540-78761-7_50

3. Boden, M.A.: Creativity and Art: Three Roads to Surprise. Oxford University Press, Chicago (2010)
4. Borenstein, J., Koren, Y.: Real-time obstacle avoidance for fast mobile robots. IEEE Trans. Syst. Man Cybern. **19**(5), 1179–1187 (1989)
5. Chappell, D.: Taking a point for a walk: Pattern formation with self-interacting curves. In: Greenfield, G., Hart, G., Sarhangi, R. (eds.) Bridges 2014 Conference Proceedings, pp. 337–340. Tessellations Publishing, Phoenix, Arizona (2014)
6. Chappell, D.: Sinuous meander patterns: a family of multi-frequency spatial RHYHMS. J. Math. Arts **9**(3–4), 63–76 (2015)
7. Day, R.L., Laland, K.N., Odling-Smee, J.: Rethinking adaptation: the niche-construction perspective. Perspect. Biol. Med. **46**(1), 80–95 (2003)
8. d'Inverno, M., McCormack, J.: Heroic vs collaborative AI for the arts. In: Proceedings of IJCAI 2015 (2015)
9. Dohm, K., Hoffmann, J.: Kunstmaschinen Maschinenkunst (Art Machines, Machine Art). Kehrer Verlag, bilingual (German/English) edn. (2008)
10. Eisner, T.: For Love of Insects. The Belknap Press of Harvard University Press, Cambridge (2003)
11. 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). doi:10.1007/978-3-642-29142-5_10
12. Greenfield, G.: Avoidance drawings evolved using virtual drawing robots. In: Johnson, C., Carballal, A., Correia, J. (eds.) EvoMUSART 2015. LNCS, vol. 9027, pp. 78–88. Springer, Cham (2015). doi:10.1007/978-3-319-16498-4_8
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). doi:10.1007/11732242_58
14. Grimm, V., Railsback, S.F.: Individual-based Modeling and Ecology, Princeton Series in Theoretical and Computational Biology. Princeton University Press, Princeton (2005)
15. Kennedy, J., Eberhart, R.C., Shi, Y.: Swarm Intelligence. Morgan Kaufmann Publishers, San Francisco (2001)
16. Lucas, G.W.: An elementary model for the differential steering system of robot actuators. Technical report, The Rossum Project (2000). http://rossum.sourceforge.net/papers/DiffSteer/DiffSteer.html
17. McCormack, J.: Enhancing creativity with niche construction. In: Fellerman, H., Dörr, M., Hanczyc, M.M., Laursen, L.L., Maurer, S., Merkle, D., Monnard, P.A., Stoy, K., Rasmussen, S. (eds.) Artificial Life XII, pp. 525–532. MIT Press, Cambridge (2010)
18. McCormack, J.: Creative ecosystems. In: McCormack, J., d'Inverno, M. (eds.) Computers and Creativity, Chap. 2, pp. 39–60. Springer, Heidelberg (2012)
19. McCormack, J.: Aesthetics, art, evolution. In: Machado, P., McDermott, J., Carballal, A. (eds.) EvoMUSART 2013. LNCS, vol. 7834, pp. 1–12. Springer, Heidelberg (2013). doi:10.1007/978-3-642-36955-1_1
20. Moura, L.: Machines that make art. In: Herath, D., Kroos, C., Stelarc, (eds.) Robots and Art, Cognitive Science and Technology, pp. 255–269. Springer, Heidelberg (2016)
21. Moura, L., Pereira, H.G.: Man + Robots – Symbiotic Art. LxXL, Black and white facsimile edn. (2014)
22. Odling-Smee, J., Laland, K.N., Feldman, W.M.: Niche Construction: The Neglected Process in Evolution. Monographs in Population Biology. Princeton University Press, Princeton (2003)

23. Perlin, K.: Improving noise. ACM Trans. Graph. (TOG) **21**(3), 681–682 (2002)
24. Roudavski, S., McCormack, J.: Post-anthropcentric creativity. Digital Creativity **27**(1), 3–6 (2016)
25. Still, A., d'Inverno, M.: A history of creativity for future AI research. In: Pachet, F., Cardoso, A., Corruble, V., Ghedini, F. (eds.) Proceedings of the Seventh International Conference on Computational Creativity (ICCC 2016), pp. 147–154, June 2016