

# An Interactive Electronic Art System Based on Artificial Ecosystemics

Taras Kowaliw,  
Institut des Systèmes Complexes - Paris Île-de-France  
Centre national de la recherche scientifique,  
57-59 rue Lhomond, 75005, Paris, France  
Web: <http://kowaliw.ca>

Jon McCormack, and Alan Dorin  
Centre for Electronic Media Art,  
Faculty of Information Technology, Monash University  
Clayton, Victoria 3800, Australia  
Web: <http://www.csse.monash.edu.au/cema>

**Abstract**—In this paper, we explore a generative art system designed to promote the creation of a diverse range of aesthetically pleasing images. We introduce our system, *EvoEco*, an agent-based pixel-level means of generating images based on artificial ecosystems. This art system is driven by interactive evolutionary computation, and further augmented using special measures to promote the diversity of the individuals. Following previous work, we explore a tractable definition of creativity and its addition to this interactive search. *EvoEco* was released online, and used by forty-one anonymous users to generate artwork. Here we present some of the discovered results.

**Index Terms**—Interactive evolution, computational creativity, electronic art, generative art, artificial ecosystem

## I. INTRODUCTION

Generative art involves the use of a generative process (e.g. an algorithm, stochastic process, or dynamical system) for the creation of artifacts. In this way, a simple seed (e.g. a parameter set, a genetic representation) is transformed into an aesthetically rich and compelling form. This increased capacity, however, comes at the expense of prediction, and necessarily involves the introduction of model biases [1]. Here we explore the use of an artificial ecosystem as a generative system for the production of novel images.

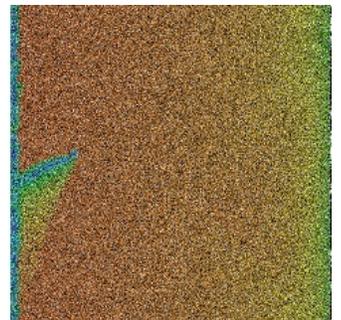
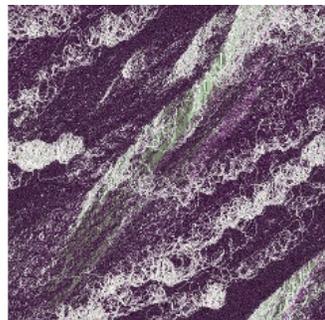
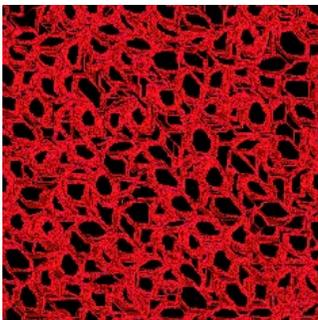
Biological ecosystems are an integral component of evolutionary diversity, where, for example, niche construction is known to support stable polymorphisms and unusual evolutionary dynamics relative to non-niche enabled models [2]. Artificial ecosystems have the capability to support a rich array of emergent dynamics quite independent of evolutionary pressures. That is, in a simple particle-based environment using only fixed agent rules, the resulting dynamics alone are

capable of generating complex patterns and life-like properties [3].

Due to their ability to generate complex structures and interactions through carefully coupled feedback components, “ecosystemic” models have become popular in generative art over recent years [4], [5], [6], [7], [8]. In a review on the topic, Dorin argues that use of ecosystemics allows artists to use a machine as a complex dynamical system, promoting: coherence and unity in the face of perturbation; multi-scale temporal complexity; the autonomous production of novelty; and susceptibility to external control [9]. Our primary concern in this paper is their ability to generate novel and multi-scale patterns autonomously.

Driessens and Verstappen’s *E-volver*, for example, is an interactive evolutionary artwork based on ecosystemic principles, and often displays unexpected and aesthetically rich behaviours. This system is driven by user-guided aesthetic rejection of the least preferable of a population of animations that are produced by multiple cellular automaton-like agents. Agents’ behaviours are coupled via their mutual modification of a shared environment [10].

In this paper we introduce an ecosystemic means of generating images, loosely inspired by Driessens and Verstappen’s work. Our *EvoEco* system utilizes a generative process based on the interaction of heterogeneous agents in a shared world. Several pixel-sized agents act and react locally in this world, over time generating an animation and, ultimately, a final image which we term the output artwork. This system is controlled via an interactive evolutionary algorithm, allowing a user to influence the gradual development of aesthetically



pleasing images. Our system was augmented with a measure designed to promote diverse designs in the evolutionary populations based on a new theoretical definition of computational creativity.

## II. COMPUTATIONAL CREATIVITY

The literature on creativity is vast. A popular definition of creativity involves the generation of *appropriate novelty* [11]. That is, a creative system reliably generates artifacts or ideas that are *new* relative to some personal or historical pretext, but also appropriate, useful, or valuable to a particular context or task.

It is important to note a subtle, but important distinction, between *aesthetically pleasing* and *creative* systems. The latter does not preclude the former, but they are in general, independent. The possibility exists that some well-chosen formal notion of creativity can be used to drive automated systems to greater utility: some progress has already been made based on psychological principles [12].

Dorin and Korb have recently introduced a new definition: “Creativity is the introduction and use of a framework that has a relatively high probability of producing representations of patterns that can arise only with a smaller probability in previously existing frameworks”. Hence, a *framework* can be creative, based on its ability — according to some measure — of reliably generating novel patterns. This definition, unlike many, considers creativity as independent of notions of value and appropriateness [13].

Previously, we have shown that this definition of creativity can be interpreted in manner that is computationally tractable. Our interpretation is slow to compute, but, an approximation of the formal definition has been shown experimentally to approximate the exact measure in an interactive electronic art task. Further, the use of this approximate measure in an evolutionary art task at times returns systems which many humans deem “creative” [14].

Unlike our previous approach, here we explore a more robust generative art mechanism, more likely to generate responses which users find genuinely novel or creative (based on the successes of previous ecosystemic artworks). Our interpretation of the Dorin & Korb definition is developed below.

Firstly, we define a feature space on which we will evaluate patterns. Drawing image features from image processing [15] and content-based image retrieval [16], [17], we can choose a set well suited but independently developed for human perception. Here we select three successful measures for detecting “creative” regions in a previous evolutionary art task, adapted for colour images:  $var(H(f))$ ,  $var(S(f))$ , and  $E(B(f))$ , where  $f$  is a colour image of any dimensions,  $var$  is a measure of the variance,  $H$ ,  $S$  and  $B$  are the hue, saturation and brightness, and  $E$  is the entropy of the histogram. Despite success in previous applications, we do not use spatial moments, since for reasons made clear later, our developed images show little respect for image regions.

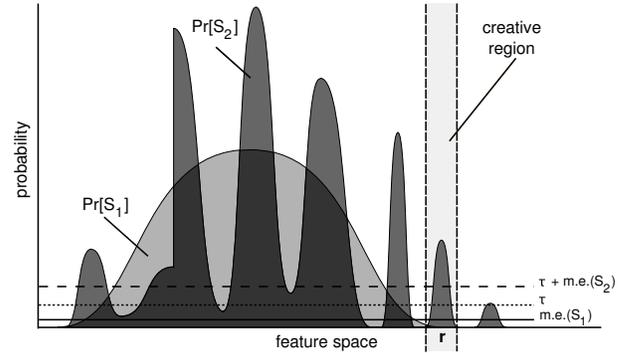


Fig. 1. An illustration of the concept of a creative region. Two hypothetical distributions are shown,  $S_1$  in light grey,  $S_2$  in medium grey, with their intersection highlighted in dark grey. A single “hump” exists in which  $Pr[S_2(x) \in r] > \tau + m.e.(S_2)$  and  $Pr[S_1(x) \in r] = 0$ , so  $r$ , drawn between dotted lines, is a creative region.

We are interested in the creativity of *systems*: generators which accept some seed and (possibly stochastically) generate some pattern  $p \in P$ . Our notion of creativity considers the capacity of a system  $S_2$  to reliably generate patterns which a system  $S_1$  cannot. One can interpret  $S_1$  as the cognitive worldview of an audience when presented with the work of a new artist, for instance, or as a memory of systems already explored in an interactive design task. Under this perspective, we make nearly no assumptions regarding the distribution of system outputs, save that non-trivially similar patterns lie in some small but minimally sized interval in feature space. Notably, we will make no other assumptions about the relationship between distance and dissimilarity.

Note that in an evolutionary computation task, any individual, along with the genetic operators, can be considered a system. That is, assume we have an individual with genotype  $g \in G$ , generative process  $\delta$ , phenotype  $p = \delta(g) \in P$ , and some stochastic genetic operator, say  $mutate : G \rightarrow G$ . Then, we can consider  $(g, \delta, mutate)$  a system which generates output  $\delta(mutate(g))$ . That is, individual  $g$  is viewed as a system for producing child patterns.

Our interpretation of the Dorin & Korb definition concerns the (Boolean) capacity of  $S_2$  to reliably generate patterns that  $S_1$  cannot. We restrict our attention to intervals of a minimum size<sup>1</sup>. Given some candidate interval,  $r$ , we can estimate the probability that  $S_1$  and  $S_2$  generate points in said region. If  $S_2$  can be shown to reliably generate points in  $r$  (i.e. the probability of finding points in  $r$  exceeds some threshold  $\tau$  and the margin of error associated with the sample  $m.e.(S_2)$ ) while the estimated probability of  $S_1$  generating points in  $r$  is zero, then we claim that  $S_2$  is creative relative to  $S_1$ . An illustration of this concept of a creative region is illustrated in Figure 1.

We are left with the question of how to find such intervals  $r$ . We do so using our samples in feature space. An attempt

<sup>1</sup>The minimum size was chosen so as to ensure that a change of this minimal amount was perceptible. We estimated this distance to be approximately  $\beta = \frac{\sigma_i}{5}$ , where  $\sigma_i$  is the standard deviation of a sample over the  $i$ -th feature.

is made to find intervals surrounding each sample point from  $S_2$  provided. If our sample pattern is  $p = (F_1(p), \dots, F_k(p))$ , then we initially define our interval about  $p$  as

$$\mathbf{b}(p) = \left[ F_1(p) - \frac{\beta}{2}\sigma_1, F_1(p) + \frac{\beta}{2} \right] \times \quad (1)$$

$$\dots \times \left[ F_k(p) - \frac{\beta}{2}\sigma_k, F_k(p) + \frac{\beta}{2}\sigma_k \right]$$

where  $\sigma_i$  is the standard deviation of the system  $S_2$  in the  $i$ -th dimension. For each such created interval, we ask if it does not contain points from  $S_1$  (as we know it contains at least one point from  $S_2$ ). If so, we attempt to generalize it. For each dimension in turn, we widen the width of the interval by a factor of 2. If we successfully define a new interval containing equal or greater points from  $S_2$  and none from  $S_1$ , we retain this new interval. Once we have traversed all dimensions, if our new interval is an improvement on the original, we traverse the dimensions again. This process will terminate either when a locally maximal interval is found, or when the interval covers the entirety of all dimensions.

If we can find an interval containing sufficient points, then we declare  $S_2$  creative relative to  $S_1$ . Alas, in cases where this technique is unable to find a creative interval, we cannot conclude that  $S_2$  is **not** creative relative to  $S_1$ , since the possibility of an interval discoverable through some other means, although unlikely, cannot be excluded. Generally, expending additional effort to make such a guarantee is of dubious value, since there is already a possibility of missing creativity in the choice of features to describe the space, *i.e.* that some other significant but unconsidered feature might exist that separates  $S_2$  from  $S_1$ .

Since searching for our formally defined creativity is a slow process, we also consider a multi-valued procedure for estimating relative creativity quickly. *Creativity lite* will take a smaller sample, and return the maximum number of samples from set  $S_2$  that can be found in some region containing no samples from  $S_1$ . We use a sample of 1,200 points from  $S_1$  and 10 points from  $S_2$ . Intervals are constructed using the technique described above. The more points from  $S_2$  that are returned, the more *creative lite* the generator. We have previously shown that use of *creativity lite* in an interactive evolutionary algorithm tends to push the optimization toward individuals considered creative by our formal definition [14].

### III. AN ECOSYSTEMIC GENERATIVE ART SYSTEM

Like Driessens and Verstappen’s *E-volver*, our image generation system is based on metaphors from biological ecosystems. *EvoEco*, is an evolutionary platform which evolves multi-agent ecosystemic generative art. It is influenced by *E-volver* in that we have borrowed the concept of a collection of distinct pixel-sized agents altering a shared HSB-defined world. Otherwise, all algorithms are our own. We break the description of *EvoEco* into two parts: the ecosystemic (generative) stage, through which a genome is transformed into a phenotype (image); and the evolutionary stage, during which particular individuals are selected and evolved.

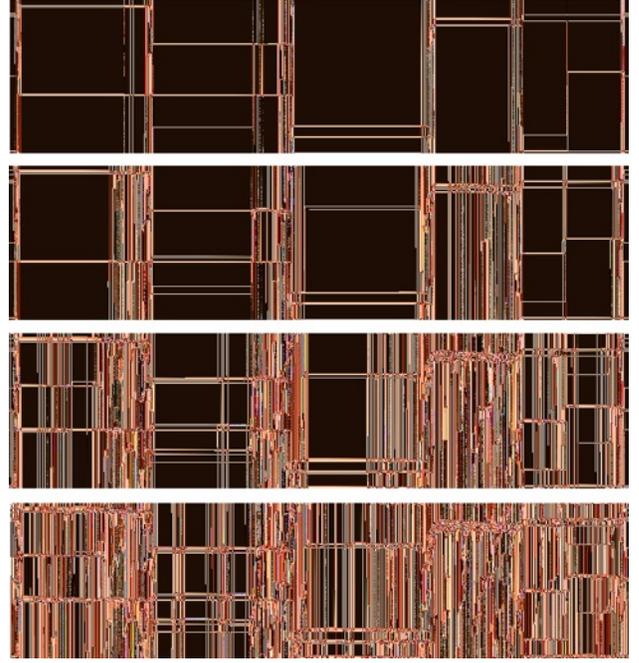


Fig. 2. Four timeslices of the development of the final phenotype of an evolved individual. Individual time proceeds downwards. This individual consists of five agents, one of whom draws an initial blocking pattern, while the remaining four elaborate on the edges.

#### A. Ecosystemic Development in EvoEco

An “individual” is an ecosystem: a collection of agents on a toroidal grid. Over discrete time, the agents and their interactions on the grid will produce an image, referred to as a phenotype. An individual’s grid is of size  $w \times h$  of HSB values<sup>2</sup>, initialized as a field of colour ( $ind_H, ind_S, ind_B$ ), an individual’s genetically specified preferred colour.

Each individual is a collection of  $k$  agents, placed in one of six-by-six equally spaced grid positions. Each agent then executes its program for  $2wh$  time steps, more than sufficient time to (theoretically) visit every pixel in the image. Agent actions are executed serially, according to their priority in the genome (in practice, given the small number of agents in a large space, there is little to no difference between this and buffered parallel execution). Development can be viewed as an animation, proceeding from a simple flat colour to the final image. The mapping between individual and grid to phenotype is deterministic, however different grid sizes lead to different phenotypes. Some snapshots of the development of an evolved individual is shown in Figure 2.

1) *Agents*: An agent is the size of a single pixel, located somewhere in the image. At every time step, it: (a) queries its local neighbourhood, collecting a description as input; (b) queries its genome, mapping the input to some output variables; (c) colours the current pixel according to those output variables; and (d) moves by a single pixel in the

<sup>2</sup>Typically we use a size of  $224 \times 168$ , chosen for a 4:3 aspect ratio, and such that a square of  $4 \times 4$  individuals can be displayed easily on a monitor of resolution  $1024 \times 768$ .

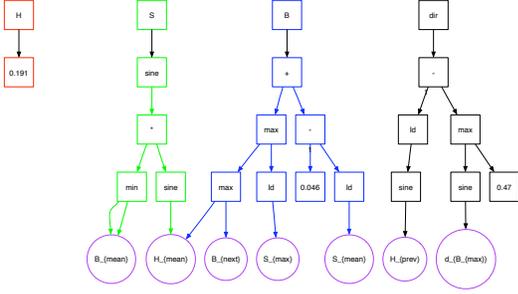


Fig. 3. An example of an agent’s genetic programs visualized as a tree. Redundancy (*i.e.* neutral code, or “junk DNA”) has been trimmed. This same individual can be written in mathematical notation as:

$$\begin{aligned}
 a_H &= 0.191 \\
 a_S &= \sin(\sin(H_{mean})B_{mean}) \\
 a_B &= (0.046 - S_{mean}) + \max\{S_{max}, \max\{B_{next}, H_{mean}\}\} \\
 a_{dir} &= \sin(H_{prev}) - \max\{\sin(d_{B_{max}}), 0.47\}
 \end{aligned}$$

where the leaf nodes in the trees correspond to some of the twenty-six inputs.

direction specified by its output. Each agent has a distinct genome, meaning each executes different behaviours in a shared world.

An agent can be written:

$$a = (g, t_1, t_2, sD, delay, stop, (a_H, a_S, a_B), a_{dir})$$

where  $sD$ , is a starting direction,  $delay$  and  $stop$  indicate when an agent should start or stop execution during its lifespan,  $(a_H, a_S, a_B)$ , and  $a_{dir}$  are four GP-trees used to determine the colour that an agent draws in the current pixel, and the direction it travels next, and  $g, t_1$ , and  $t_2$  are floating values. The latter two are time periods used in looping functions.

First, the agent collects a description of its Moore neighbourhood. This description consists of a list of twenty-six variables, which include: the maximum, mean, and minimum hue, brightness and saturation of the neighbourhood, the associated directions, the genetically controlled constants and looping variables, the previous direction of travel, a (consistently computed) random value, and the directions associated with edge-following for dark and light edges.

Next, the agent will paint its current pixel to a new HSB colour triple determined via three genetic programs, named  $a_H, a_S$ , and  $a_B$ .

Finally, the agent will move in a direction specified by a fourth genetic program,  $a_{dir}$ .

The mapping between the inputs and the four outputs is accomplished through four binary trees of depth four. The leaves of all four trees are mapped to input variables. The non-leaf nodes contain an integer specifying a function type, and a floating point number specifying a constant value. All arithmetic is done on the toroidal number space  $[0, 1]$ , meaning that the space is treated as a ring. An example of an agent’s GP trees is shown in Figure 3.

2) *Genetic Initialization and Operators*: An individual is the  $(k + 4)$ -tuple:

$$I = (k, (ind_H, ind_S, ind_B), a^1, \dots, a^k)$$

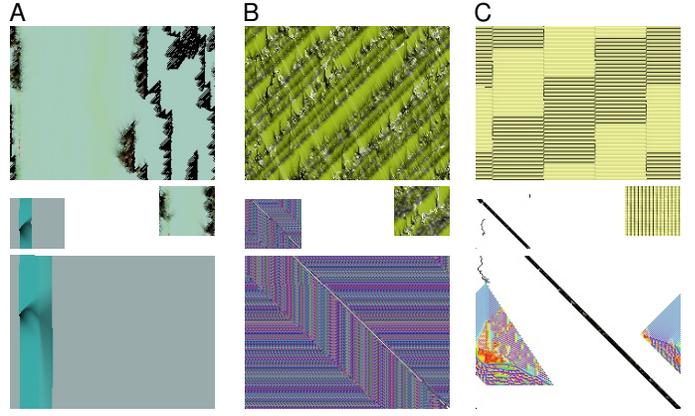


Fig. 4. Examples of select randomly-generated genotypes grown at two different sizes.

Let  $urand$  be a function which chooses with a uniform random selection from its arguments. We initialize an individual by setting  $k = urand\{2, \dots, 6\}$ ,  $ind_X = urand([0, 1])$ , and then generating  $k$  randomly initialized agents.

An agent, we recall, can be written

$$a = (g, t_1, t_2, sD, delay, stop, (a_H, a_S, a_B), a_{dir})$$

An agent can be randomly initialized by generating  $g, t_1, t_2, sD$  all as  $urand([0, 1])$ ,  $delay = urand([0, 0.5])$ ,  $stop = urand([0.5, 1])$ , and then randomly generating the four action trees.

The trees  $(a_H, a_S, a_B)$ , and  $a_{dir}$  are each represented by a list of numbers:

- $2^{d-1}$  input nodes, each represented by an integer  $\in \{0, \dots, 25\}$ .
- $2^{d-1} - 1$  non-leaf nodes representing functions in the tree, represented by integers  $\in \{0, \dots, 11\}$ .
- $2^{d-1} - 1$  floating-point constant values in  $[0, 1]$

Tree connectivity is complete. All of the values are initialized uniformly and randomly, save the functions in the non-leaf nodes. These are chosen from the following set of (maximum) two parameter functions, defined on inputs  $x$  and  $y$ :

$$\left\{ id, x + y, x - y, xy, \max\{x, y\}, \min\{x, y\}, \sin(x), \frac{x}{y}, incdec, \frac{x+y}{2}, const \right\}$$

where  $id$  simply returns the value  $x$ , division is “safe”,  $const$  returns the associated constant value, and  $incdec(x, y) = x + \frac{1}{8}$  if  $y > 0.5$  or  $x - \frac{1}{8}$  otherwise. Each function has a 0.07 probability of being chosen, save the  $id$  function, chosen with probability 0.23. We purposefully increased the frequency of  $id$  nodes as we expected them to be used as structural nodes in the otherwise fixed topology trees, and hence to be of greater importance.

Figure 4 shows the phenotypes produced by a sample of randomly generated individuals. We may also describe the behaviour of random initialization of any individual in feature space: they have mean values for  $var(H)$  of 0.1043 (s.d. 0.1108); for  $var(S)$  of 0.1007 (s.d. 0.1087); and for  $E(B)$  of 0.2241 (s.d. 0.2622).

An interesting feature regarding the development of images via our ecosystemic techniques is that a change to the size of an individual’s world leads to a completely new development. This redevelopment is also illustrated in Figure 4. We computed 10,000 random genotypes, generating their phenotypes at two sizes:  $224 \times 168$  (large) and  $60 \times 54$  (small). Speaking informally, the redevelopment of the phenotype at a smaller size usually generates an image that resembles a sub-image of the larger original (examples *A*). In some cases, the smaller image resembles a scaled-down version of the original (*B*). Occasionally, there exist completely new patterns at one environmental size that do not exist at another (*C*). Regardless, there is a clear relation between the images grown from the same genotypes, at least on their location in feature space. Between unrelated genotypes developed in the large environments, we computed a mean distance of 0.479 (s.d. 0.310). Between phenotypes of the same genotype developed at large and small environments, we computed a mean distance of 0.082 (s.d. 0.111).

The use of an artificial ecosystem implies the creation of patterns via the interaction of specialized agents, and the reaction of agents to the niches built by others. As such, genetic operators based on agents are strongly desirable, especially the possibility of combining specialized agents from one individual into another. Hence, unlike many interactive art tasks, we lust after a meaningful crossover operator.

We can apply *crossover* to a pair of agents,  $a^1$  and  $a^2$ , by performing sub-tree crossover on the GP trees. We initially define child agent  $a^3$  to be an exact copy of agent  $a^1$ . Next, for each of the  $a_H, a_S, a_B, a_{dir}$  trees, we randomly, uniformly, and independently select an index in the range  $i \in \{0, \dots, 2^d - 1\}$ . For each index and tree, we replace the node value referenced by that index, and all children, with the appropriate values from agent  $a^2$ . We use the same index and children to swap the constant values associated with the chosen tree as well. Finally, we randomly choose values from either parent for the new variables  $g, t_1, t_2, sD, start$  and  $delay$ .

Given two individuals,  $I^1 = (k^1, (ind_H^1, ind_S^1, ind_B^1), a^1, \dots, a_{k^1}^1)$  and  $I^2$ , we can generate a new individual  $I^3$  via *crossover*:

```

 $k^3 = urand(k^1, k^2)$ 
 $ind_X^3 = urand(ind_X^1, ind_X^2)$ 
for  $m \in \{1, \dots, k^3\}$  do
   $i = urand(1, \dots, k^1), j = urand(1, \dots, k^2)$ 
   $a_m^3 = urand(a_i^1, a_j^2, crossover(a_i^1, a_j^2))$ 
end for

```

We can apply *mutation* to an agent by applying it to the variables comprising the collection of trees. For each variable, each element has a *probMut* chance of being mutated, independently. If selected, its value is replaced by a new uniform randomly selected value.

Given some individual  $I^1$  and a probability of mutation *probMut*, we can generate a new individual  $I^2$  via *mutation*: first, with probability *probMut*, we alter the individual’s genetically specified colours by some Gaussian noise of variance

0.2. Next, we mutate each agent with probability *probMut*. Next, so long as the individual is not too large or small (*i.e.* so long as  $2 \leq a_k^2 \leq 15$ ) we add a new agent, or delete one, with probability *probMut*. Adding a new agent is done either through cloning an existing agent in the genome, or by adding a newly initialized agent, allowing for both duplication of desirable properties and increase in complexity.

## B. Evolution

In designing our IEA, we were motivated by three constraints. Firstly, since the intermingling of agents from preceding individuals is an intuitive means of defining a new individual, we expected that crossover was an important genetic operator. Secondly, we designed a simple, single-click interface to make the survey accessible to a broad audience. Finally, we hoped to allow the user to devote their focus on their preferred individuals, rather than on the least favoured, as seems to be demanded by aesthetic rejection. To that end, we chose a model in which users selected their preferred individual from a population, from which any next generation individual was generated through mutation or crossover with the preceding individual in its place. This IEA is described below.

Our system is initialized with two background structures. Firstly, a memory consisting of 12,000 points in feature space. These points correspond to pre-computed randomly generated individuals. Secondly, a database consisting of 75 pre-evolved individuals selected by the authors. These were included to help generate appealing individuals in the initial generations.

Our GA consists of a two-part interface: a population of sixteen individuals, and a history, initially empty. The population is initialized randomly, and at any generation, a user may choose to select an individual from the population, select an individual from the history, or respawn the entire population. A screenshot of the interface is shown in Figure 5. A selected individual from the population or the history will be referred to as *the Khan*<sup>3</sup>.

Once a generation is completed (*i.e.* a user selects an individual or “respawn”), every population member is added to the memory. Each is slotted into a random location, replacing a previous entry. Hence, the memory size is constant.

Given the selection of a Khan, the next population is generated as follows: First, the Khan is added as the top item of the history; Next, a single individual is added to the population via creativity search, discussed below; Next, the remainder of the population is generated via *crossover*, with probability 0.33, or otherwise, via mutation of the Khan. The crossover operation typically utilizes the Khan and the original member of the population being replaced (with probability 0.6). Also possible is a crossover between the original population member and a randomly chosen member of the history (with probability 0.2) or with a randomly chosen member from a database of pre-evolved individuals (with probability 0.2).

<sup>3</sup>Named in honour of Genghis Khan, who geneticists believe to be a direct male-lineage ancestor of approximately 0.5% of the world’s population [18]. It’s good to be the Khan!

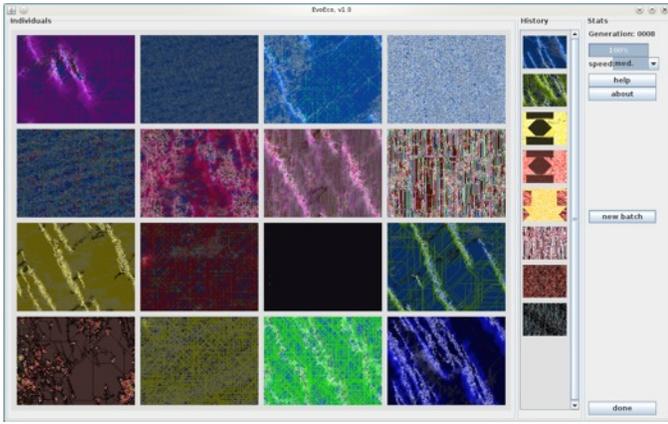


Fig. 5. A screenshot of the EvoEco Applet. On the left is the population of sixteen individuals; towards the right is the history, showing the previous eight choices of Khan; on the far right are the user controls.

Creativity search consists of the following process: First, sixteen individuals are generated via mutations of the preceding population (or randomly, in the initial population). For each of these sixteen individuals, ten mutated children are created. The potential individual with the highest *creativity lite* score over its ten children compared to the memory is selected, and added as a member of the population.

The EvoEco system was instantiated as a Java applet, and linked from a webpage. The webpage was advertised via email, fliers, and postings on the authors' websites. The applet was also installed as a part of a group gallery exhibit, *Biotope*, at the Guilford Lane Gallery in Melbourne, where it ran for approximately three weeks. Little attempt was made to ensure a representative sample: our advertising was aimed at an artistically-literate audience.

#### IV. RESULTS

We obtained forty-one responses from anonymous users, having excluded several responses due to technical difficulties. Despite our biased advertising we obtained results from users in relatively diverse demographics, as measured by metrics included in a short survey.

As expected, the genetic operators on the space allowed for a form of evolvability; That is, fit individuals under these operators tend to produce varied but fit children. Indeed, many particular agents were found to play specific and transferable roles in the development of images. For instance, we identified agents specialized at the creation of an initial form. Some drew outlines of circles, rectangles or octagons. Others were found playing roles associated with the image post-processing: for instance, agents were found which would execute a random or semi-random walk over an image blurring the area over which they passed (using the GP *mean* function); other individuals would pass over an image adding a particular tint by slightly altering the H channel, or a gradient of tint, or darkening or brightening an area using the GP *incdec* function. Many agents specialized at following existing edges, either thickening or adding elaboration to the curves, or filling in enclosed

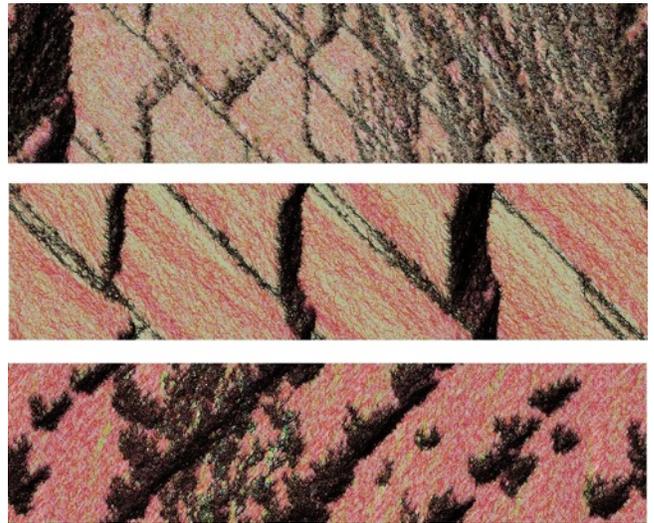


Fig. 6. The phenotype of an individual selected by one of our anonymous users (*top*), and two related individuals created via crossover and mutation.

areas. Indeed, there existed many structures and effects which were highly transferable via the genetic operators, leading to offspring that preserve specific qualities of an image, without necessarily preserving the quantitative feature values.

One such individual and some of its children are shown in Figure 6. In this instance, although the overall form of the images changes dramatically (as measured by a spatial moment on the brightness channel, say), there is consistency in the produced structures (*e.g.* thin and fat black lines) and effect (*e.g.* consistently oriented fuzzy edges, textured background).

A second critical question had to do with the scalability of the generated individuals. Due to the demands of online delivery, only modest hardware assumptions could be made, resulting in very small phenotypes. As anticipated from our earlier experiments with randomly generated genotypes, there was a great deal of similarity between phenotypes of evolved individuals regrown at different sizes. We regrew each of the selected individuals at several sizes and aspect ratios ( $800 \times 600$ ,  $900 \times 900$ ,  $1024 \times 728$ ,  $1440 \times 1080$ ), and examined the results. Six out of forty-one showed wildly different phenotypes at some size of growth: usually, this was due to a trivial phenotype. These trivial phenotypes were usually due to a lack of meeting between agents travelling in orthogonal directions, due to a change in width, height, or aspect ratio. In all other cases, however, distance in feature space was small relative to unrelated genotypes, and in visual terms, recognizable structures remained intact at all sizes. Some examples are shown in Figure 8.

Some samples of evolved outputs are shown in Figure 7. There is a great deal of diversity in the results. Different users produced images with highly distinct aesthetic styles. Indeed, the complexity of the evolved images range from near noise to subtle textures; the level of image structure ranges from highly patterned and ordered fields of varying but similar elements (*e.g.* fields of circles), to images resembling

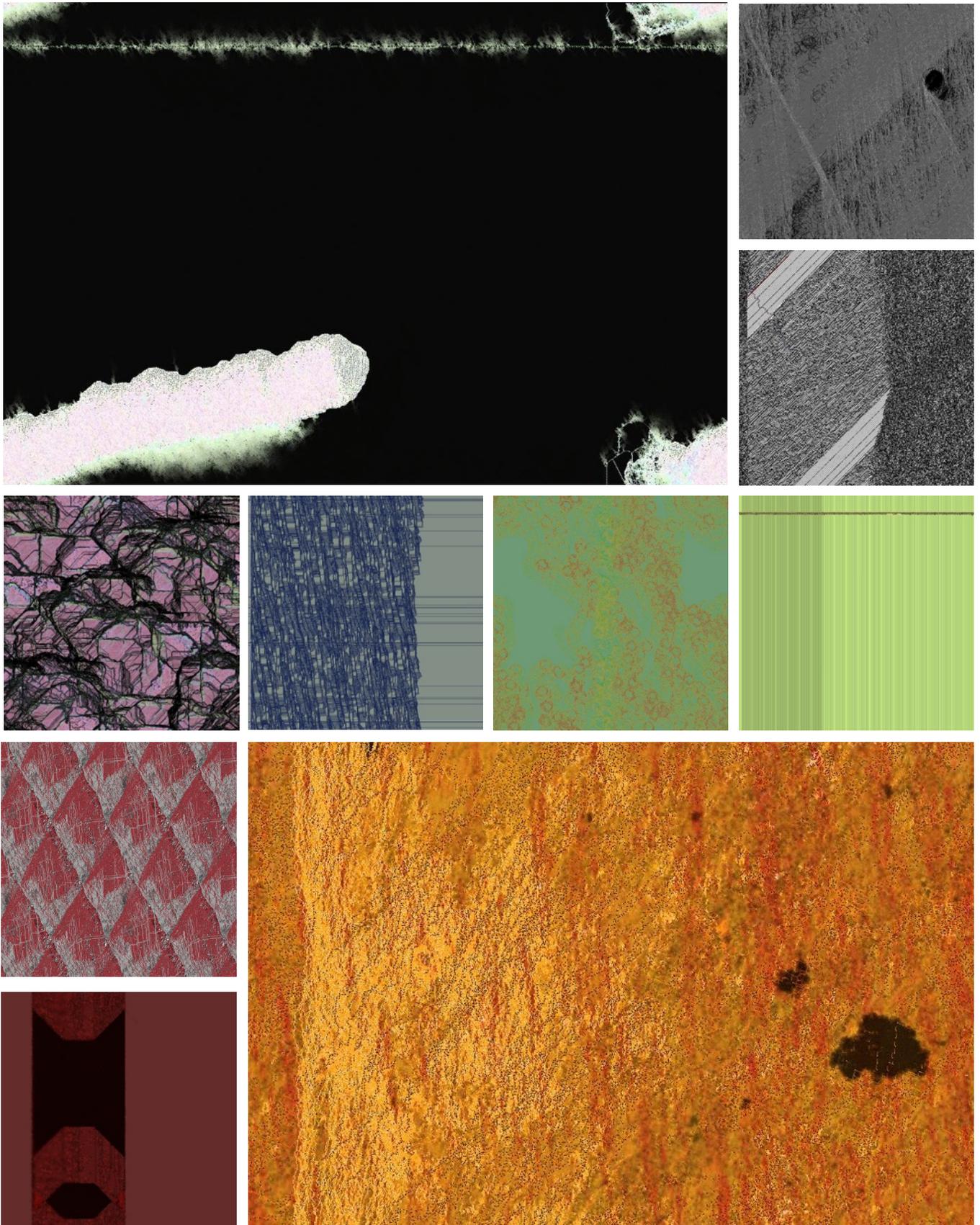


Fig. 7. A sample of outputs evolved by the authors and the anonymous artists. Best viewed in colour. More examples, sample applet output, and high-resolution versions are available at: <http://www.csse.monash.edu.au/cema/evoeco>.

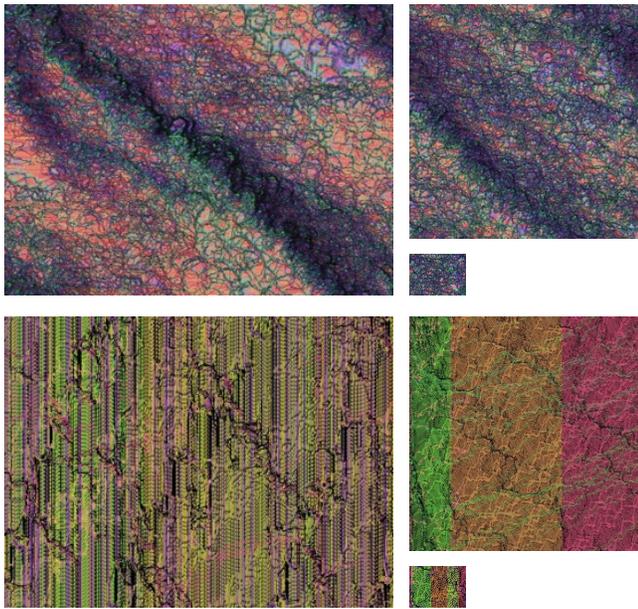


Fig. 8. Evolved individuals regrown at: the original size ( $224 \times 168$ ), along with two larger sizes ( $900 \times 900$ ,  $1440 \times 1080$ ).

pure noise, to images consisting of flat colour and simple geometric constructs. Some images exhibited a wild range of colours, while others showed subtle changes within a small and consistent palette.

This diversity of visual form is a testament to both the capacity for diverse behaviours present in an artificial ecosystem, and also in the capacity of the Dorin & Korb creativity measure to help find such regions of genotypic space.

## V. CONCLUSIONS

In this paper, we have introduced EvoEco, a novel ecosystemic art system driven by an interactive evolutionary algorithm. EvoEco was augmented by an interpretation of a theoretical definition of creativity in an attempt to further explore the diversity expected from an artificial ecosystem. The system was explored by the authors and by forty-one anonymous online users. We have argued that, in a visual sense, the system is highly evolvable: that is, there exist genetic operators on the space of EvoEco individuals that search and recombine individuals in visually meaningful ways. Further, we have argued that these individuals are capable of scaling, allowing for an exploration on a space of small (and hence, fast) images to produce individuals capable of being regenerated into high-resolution images. Results of this exploration have produced popular images reflecting many diverse aesthetic styles.

EvoEco demonstrates that ecosystemic models can evolve pleasing images and produce a wide range of aesthetic styles. This establishes it as a system capable of evaluating claims regarding computational creativity. Our current work involves doing precisely this with respect to our creativity definition. We have conducted a user survey contrasting the Dorin & Korb definition against other means of generating population

diversity, evaluating the results on the basis of user survey responses [19].

## ACKNOWLEDGEMENT

Our gratitude to all the anonymous artists who participated in our experiments. This research was supported by an Australian Research Council Discovery Grants DP0772667 and DP1094064.

## REFERENCES

- [1] D. Ashlock, "Evolutionary design in embryogeny," in *Design by Evolution: Advances in Evolutionary Design*, P. H. et al., Ed. Springer-Verlag, 2008, pp. 199–202.
- [2] K. Laland, F. Odling-Smee, and M. Feldman, "Evolutionary consequences of niche construction and their implication for ecology," *Proceedings of the National Academy of Science*, vol. 96, pp. 10242–10247, 1999.
- [3] M. Rönkkö, "An artificial ecosystem: Emergent dynamics and lifelike properties," *Artificial Life*, vol. 13, no. 2, pp. 159–187, 2007.
- [4] J. McCormack, "Eden: An evolutionary sonic ecosystem," *Advances in Artificial Life, Proceedings of the Sixth European Conference, ECAL*, vol. LNCS 2159, pp. 133–142, 2001.
- [5] A. Di Scipio, "'sound is the interface': from interactive to ecosystemic signal processing," *Organised Sound*, vol. 8, no. 3, pp. 269–277, 2003.
- [6] A. Dorin, "The virtual ecosystem as generative electronic art," in *Proceedings of 2nd European Workshop on Evolutionary Music and Art, Applications of Evolutionary Computing: EvoWorkshops 2004*, G. e. a. Raidl, Ed. Heidelberg, Germany: Springer-Verlag, 2004, pp. 467–476.
- [7] J. McCormack, "Creative ecosystems," in *Proceedings of the 4th International Joint Workshop on Computational Creativity*, A. Cardoso and G. Wiggins, Eds., 17–19 June 2007 2007, pp. 129–136.
- [8] J. McCormack and O. Bown, "Life's what you make: Niche construction and evolutionary art," in *EvoWorkshops 2009, LNCS 5484*, M. Giacobini et al., Ed. Springer-Verlag, 2009, pp. 528–537.
- [9] A. Dorin, "A survey of virtual ecosystems in generative electronic art," in *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*, J. Romero and P. Machado, Eds. Springer-Verlag, 2008.
- [10] E. Driessens and M. Verstappen, "Natural processes and artificial procedures," in *Design by Evolution: Advances in Evolutionary Design*, P. F. Hingston, L. C. Barone, and Z. Micalewicz, Eds. Springer, 2008.
- [11] M. A. Boden, *The creative mind: myths & mechanisms*. New York, N.Y.: Basic Books, 1991, no. xii, 303.
- [12] S. DiPaola and L. Gabora, "Incorporating characteristics of human creativity into an evolutionary art algorithm," *Genetic Programming and Evolvable Machines*, vol. 10, no. 2, pp. 97–110, 2009.
- [13] A. Dorin and K. Korb, "Improbable creativity," in *Proceedings of the Dagstuhl International Seminar on Computational Creativity*, J. McCormack, M. Boden, and M. d'Inverno, Eds. Springer, 2009.
- [14] T. Kowaliw, A. Dorin, and J. McCormack, "An empirical exploration of a definition of creative novelty for generative art," in *Artificial Life: Borrowing from Biology (ACAL '09)*, vol. 5865. Springer, 2009, pp. 1–10.
- [15] J. Shen, "Stochastic modelling western paintings for effective classification," *Pattern Recognition*, vol. 42, pp. 293–301, 2009.
- [16] A. Vailaya, M. Figueiredo, A. Jain, and H.-J. Zhang, "Image classification for content-based indexing," *Image Processing, IEEE Transactions on*, vol. 10, no. 1, pp. 117–130, Jan 2001.
- [17] P. Howarth and S. Rüger, "Evaluation of texture features for content-based image retrieval," in *CIVR 2004, LNCS 3115*, 2004, pp. 326–334.
- [18] T. Zerjal, Y. Xue, G. Bertorelle, R. S. Wells, W. Bao, S. Zhu, R. Qamar, Q. Ayub, A. Mohyuddin, S. Fu, P. Li, N. Yuldasheva, R. Ruzibakiev, J. Xu, Q. Shu, R. Du, H. Yang, M. E. Hurles, E. Robinson, T. Gerel-saikhan, B. Dashnyam, S. Q. Mehdi, and C. Tyler-Smith, "The genetic legacy of the mongols," *The American Journal of Human Genetics*, vol. 72, no. 3, pp. 717 – 721, 2003.
- [19] T. Kowaliw, A. Dorin, and J. McCormack, "Promoting creative design in interactive evolutionary computation," *IEEE Transactions on Evolutionary Computation*, 2011 (Under submission).