

Promoting Creative Design in Interactive Evolutionary Computation

Taras Kowaliw, Alan Dorin, Jon McCormack

Abstract—We use a new measure of creativity as a guide in an interactive evolutionary art task and tie the results to natural language usage of the term “creative”. Following previous work, we explore a tractable definition of creativity, one emphasizing the novelty of systems, and its addition to an interactive application. We next introduce a generative ecosystemic art system, *EvoEco*, an agent-based pixel-level means of generating images. *EvoEco* is used as a component of an online survey which asks users to evolve a pleasing image and then rank the success of the process and its output. Evolutionary search is augmented with the creativity measure, and compared with control groups augmented with either random search or a measure of phenotypic distance. We show that users consistently rate the creativity measure-enhanced version as more “creative” and “novel” than other search techniques. We further derive additional insights into appropriate forms of genetic representation and pattern space-traversal in an interactive evolutionary algorithm.

Index Terms—Interactive Evolution, Computational Creativity, Electronic Art, Artificial Ecosystem

I. INTRODUCTION

Interactive evolutionary computation (IEC) is the use of a human-computer interface as a component of an evolutionary optimization task. An interactive evolutionary algorithm (IEA, including interactive genetic algorithms) typically uses a human’s opinion as some component of the objective function: a powerful means of approaching design, allowing for a combination of the search capacities of evolutionary computation with a user’s expert knowledge or aesthetic criteria [30]. It is an active research area, used in many diverse domains (summarized in [30], [35]). More recent work includes the

design of ergonomic chairs [5], building layouts [33], and hearing aids [36].

IEC has a close relationship with electronic art, with many seminal works realized via IEAs. Indeed, arguably the first example of an IEA¹ was an art task, undertaken by hand rather than on computer: artist William Latham utilized a large piece of paper, a die, and a methodical drawing procedure to explore a generative space of sculptural forms, a precursor to his later machine-based work [37]. IEC has been intimately involved in the electronic arts (reviewed in [20]), and remains an active area of research [15], [29], [32].

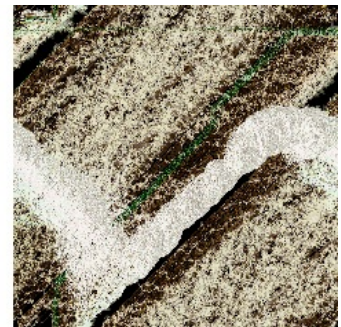
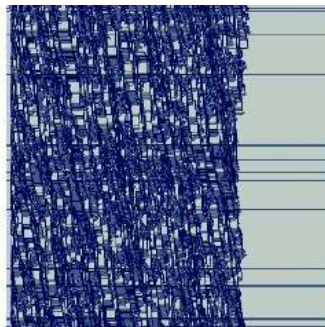
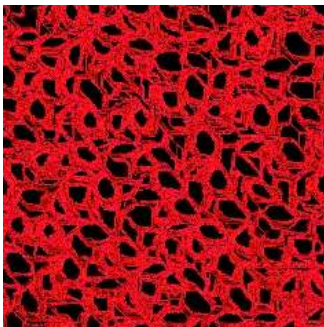
There have been several attempts to evolve art without interactivity as well, where the subjective opinion of the human operator is replaced with a fitness function designed to emulate human aesthetic preference. These attempts include the training of neural networks on successful examples of art [3]; the use of mathematics based on the complexity and compressibility of the images [23]; and the use of co-evolutionary techniques [13]. While some success has been attained by these systems, we generally believe that it is necessary, at least given the current state-of-the-art, to keep a human-in-the-loop. Indeed, this was an explicit recommendation of two of the aforementioned papers.

There are, however, several obstacles to the use of interactive evolution, a significant example being user fatigue: evolutionary computation works via the evaluation of many potential patterns, and often the number of evaluations required by certain domains or representations can overwhelm a human operator [9], [27]. There have been several approaches to ameliorating user fatigue, including: using meta-feature clustering to gradually learn and predict a user’s judgement [25]; using a partial ordering and an SVM to synthesize a subjective fitness function [21]; transforming the representation to reflect a value-based model [17]; training a

T. Kowaliw is with the Institut des systèmes complexes - Paris Île-de-France, Centre national de la recherche scientifique, 57-59 rue Lhomond, 75005, Paris, France. A. Dorin and J. McCormack are with the Faculty of Information Technology, Monash University, Clayton VIC 3800, Australia. Web: <http://www.infotech.monash.edu.au/~cema>, email: taras@kowaliw.ca

Manuscript received 29 September 2010; revised 8 February 2011; accepted 11 July 2011; current version: 21 July 2011. This research was supported by an Australian Research Council Discovery Grant DP0772667.

¹Latham’s work was approximately contemporaneous with Dawkins’ *Biomorphs* [6], also often cited as the first example of an IEA.



neural network on a database of externally-collected, highly fit examples to automatically define an iterative fitness function [24]; and expanding the role of the artist during evolution by allowing the artist several means of access to the system, including direct manipulation of the genotypes [22]. Here we look at a different strategy for reducing user fatigue by attempting to automatically detect and emphasize creative designs. Unlike approaches that require modification of the design space representation, our approach is applicable in cases where the relationship between representation and final design is complex and nonlinear (*e.g.*, generative systems); It does, however, require a collection of features adept at describing the design space.

In this paper, we consider the use of an IEA in an evolutionary art task. Specifically, we develop an artistic engine we believe capable of autonomously generating a wide array of novel images, and evolve those images via an IEA. To augment this algorithm, we introduce a measure of creative novelty, based on a new theoretical definition. Via a user study, we evaluate our system, comparing the augmented version with two control versions, and ultimately judging the success of our approach by considering the natural language responses of users.

Section II discusses some previous notions of creativity and their inclusion in automated design tasks. We present a recent theoretical definition, including our simplified version thereof, defined using a collection of features over the space of images. We also present a fast means of approximation. Section III discusses the use of ecosystem inspired methods (“ecosystemics”) in generative art. We introduce our ecosystemic process for generating images, *EvoEco*, and define an IEA using three special search techniques: the first based on our measure of creativity, a second based on phenotypic distance, and a third consisting of simple random search. Finally, Section IV presents and analyzes the user study through which we evaluate the success of *EvoEco*.

II. A TRACTABLE THEORY OF CREATIVITY

The literature on creativity is vast. A popular definition of creativity is that it involves generation of *appropriate novelty* [4]. That is, a creative system is reliably able to generate artefacts or ideas that are *new* relative to some personal or historical pretext. Not only are these artefacts or ideas new, they are also appropriate, useful, or valuable in a particular context or task.

The possibility exists that some well-chosen formal notion of creativity can be used to drive automated systems to greater utility. Indeed, some progress in this regard has already been made. DiPaola and Gabora incorporated a psychological notion of fluid thinking into a GP-based evolutionary art system, utilizing an automated fitness function based on some aesthetic principles of portrait art to produce “abstract portraits” [8].

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 (*i.e.* it is possible for a machine or algorithm to generate aesthetically pleasing images without that system

being creative). In the case of the work presented here, we introduce a creative system that assists the human aesthetic selection process to generate “aesthetically pleasing” images.

Dorin and Korb have recently introduced a new and tractable 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” [11]. Hence, a *framework* can be creative, based on its ability — according to some measure — of reliably generating novel patterns.

The Dorin / Korb notion of creativity is at odds with several popular notions. For instance, this definition considers creativity as independent of notions of value and appropriateness. According to Dorin and Korb, this eliminates the contradictions inherent in tying the *perception* of novelty and its *perceived* importance to the definition of the act of generating creative ideas. This allows for the association of creativity with an act or artifact independently of the opinions of domain experts. Thus, this definition permits acts or artifacts that break with the traditions established by existing practice, as is often later recognized to be the case where creativity is concerned. Dorin and Korb provide several further motivations for their definition and respond to some obvious criticisms.

In this section, we summarize our previously explored simplified version of the Dorin / Korb definition [18]. First, we define a suitable feature space. These features will also be used in a measure of phenotypic distance. Next, we discuss what it means for a system to have some reliable additional capacity relative to another system. Finally, we will discuss a fast approximate measure of our creativity definition, which we call *creativity lite*.

Our notion of creativity, including the approximate *creativity lite*, operates on some domain of patterns $P = \{p\}$, (here, finite-sized colour images), as described by a small set of features. We consider the creativity of *systems*: generators which accept some seed and (possibly stochastically) generate some pattern p . Our notion of creativity considers the capacity of a system S_2 to reliably generate patterns which 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 δ , phenotype $p = \delta(g) \in P$, and some stochastic genetic operator, say *mutate* : $G \rightarrow G$. Then, we can consider $(g, \delta, \textit{mutate})$ a system which generates output $\delta(\textit{mutate}(g))$. That is, individual g is viewed as a system for producing child patterns.

A. Feature Space

In previous work [18], we have explored the use of a variety of image features for the description of a space of artistic images. We considered a collection of features independently developed for use in image processing [14], [34] and content-based image retrieval [16], [38]. Since these features were developed for roles in specific tasks involving human visual perception, we have good empirical reason to believe that they approximate significant aspects of human vision. Hence, we believe this collection of mass, histogram, edge, and texture features to be an appropriate choice for measuring aspects of discernability by the human visual system.

In this same previous work, several of these features were explored for their capacity to be used in an evolutionary art system, where they were partially evaluated on their ability to identify interesting regions of phenotypic space. Here, we select three which showed potential for identifying changes we intuitively consider non-trivial, adapted for colour images. We have elected to use: $var(H(f))$, $var(S(f))$, 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 channels, and E is the entropy of the histogram. Despite success in previous applications, we do not use spatial moments, since growth in our image space — a torus — shows little respect for image regions.

Given two images, f and g , we will also define the *phenotypic distance* between them:

$$d(f, g) = ((var(H(f)) - var(H(g)))^2 + (var(S(f)) - var(S(g)))^2 + (E(B(f)) - E(B(g)))^2) \quad (1)$$

i.e., a function monotone increasing with Euclidean distance in our feature space. Thus, our measure of phenotypic distance is expected to be an approximation of some significant aspects of human vision.

B. A Creative System

Let S_1 and S_2 be systems which map from a space of input, $x \in X$, to the space of patterns P . We wish to claim that S_2 is creative (or not) relative to S_1 on the basis of what can be reliably produced by the systems in question. We interpret this as the capacity to find an interval in feature space that can be reliably populated by system S_2 but not by system S_1 . An illustration of the concept of a creative region is illustrated in Figure 1.

More formally, we assume an error-tolerance τ and confidence c . We will write that an interval of feature space is $\mathbf{r} = (r_1 \pm \delta_1, \dots, r_k \pm \delta_k)$, (where $k = 3$ in the current work), and that a point p is contained in \mathbf{r} , $p \in \mathbf{r}$, if it is contained within the bounds for all feature-space dimension. We aim to estimate the true probability of S_j generating a point in interval \mathbf{r} , $P[S_j(X) \in \mathbf{r}]$, via the frequency of sample points, written \hat{P} . Assuming our sample is representative, the (conservatively estimated) margin of error associated with this frequency is

$$m.e.(\hat{P}[S_j(X) \in \mathbf{r}]) = \frac{z}{\sqrt{n}} \quad (2)$$

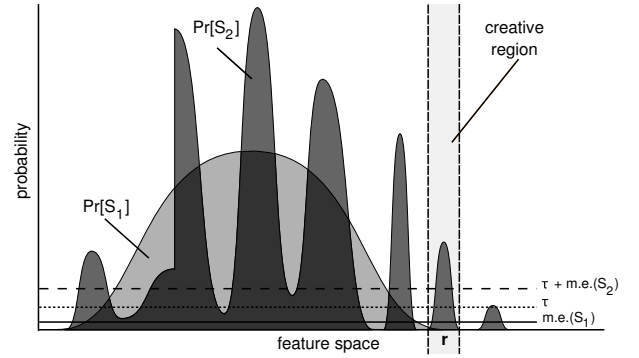


Fig. 1. An illustration of the concept of a creative region. Two hypothetical distributions are shown, S_1 in light grey, S_2 in medium grey, with their intersection highlighted in dark grey. A single “hump” exists in which $Pr[S_2(x) \in \mathbf{r}] > \tau + m.e.(S_2)$ and $Pr[S_1(x) \in \mathbf{r}]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

where z is the upper critical value for confidence-level c , and n is the sample size. Let us assume that $\tau > \frac{z}{\sqrt{n}}$, that is, that our margin of error does not dominate our error tolerance.

We state that \mathbf{r} intersects the *reliable-support* of S_j iff

$$\hat{P}[S_j(X) \in \mathbf{r}] > \tau + \frac{z}{\sqrt{n}} \quad (3)$$

Additionally, \mathbf{r} is **not** in the reliable-support of S_j iff $\hat{P}[S_j(X) \in \mathbf{r}] = 0$, with no conclusions being drawn on the region in between. In the former case, we have greater than c confidence that the probability of generating samples in the region \mathbf{r} is greater than τ , and in the latter, greater than c confidence that the probability is less than our margin of error.

Hence, provided with values for reliability τ and c , we seek to find a region \mathbf{r} which intersects the reliable-support of S_2 , but not the reliable-support of S_1 .

1) *Discovering Creative Intervals*: We attempt to find intervals in feature space that exist in the reliable support of system S_2 , but not in the reliable-support of system S_1 . Using a confidence interval of $c = 95\%$, an error tolerance of $\tau = 0.03$, and a sample size of $n = 500$, we need to find regions with at least $n(\tau + \frac{z}{\sqrt{n}}) = 59$ samples from system S_2 , and none from system S_1 . It is natural to base the minimal size of an interval on the standard deviation of the sample. Rather than include any such interval capable of supporting the mass of points required, we will instead require that a minimal length² of interval be $\beta = \frac{\sigma}{5}$.

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

$$\mathbf{b}(p) = \left(F_1(p) \pm \frac{\beta}{2}\sigma_1^j, \dots, F_k(p) \pm \frac{\beta}{2}\sigma_k^j \right) \quad (4)$$

where S_j is the system from whence sample p was drawn,

²This number was chosen such that during a search of possible values it led to less than 1% of random genomes being declared creative. That is, using a smaller value we risked declaring regions resulting from a particularly impotent application of mutation (*i.e.*, mutations which had no noticeable effect on phenotype) as creative.

and σ_i^j is the standard deviation of the system S_j 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. Finally, the best interval found is tested to see if it is in the reliable-support of S_2 .

If we can find such an interval, 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 excluded³.

2) *Creativity Lite*: Since searching for the 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 IEA tends to push the optimization toward individuals considered creative by our formal definition [18].

These measures, including *creativity lite*, are somewhat expensive computationally, depending on many searches through a potentially large memory. Since we desire the memory to describe a distribution over a feature space, we can expect the curse of dimensionality to be a factor. We note, however, that in many IEA applications the computational bottle-neck is the human user, and hence that often computational time is available.

III. EVOECO

Generative art is often concerned with increasing the (visually perceived) complexity of the design space. Via an embryogenic stage, the *richness* of expression of a simple seed can be increased. This increased capacity, however, comes at the expense of prediction, and necessarily involves the introduction of model biases [1]. Here our generative procedure is an artificial ecosystem.

Biological ecosystems are known to be an integral component of evolutionary diversity, where niche construction is known to support stable polymorphisms and unusual evolutionary dynamics, relative to non-niche enabled models [19]. Artificial ecosystems are believed to be capable of supporting

³Generally, 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 .

a rich array of emergent dynamics quite independent of evolutionary pressures. That is, in a simple particle-based environment using only static agent rules, the resulting dynamics alone are capable of generating complex patterns and life-like properties [31].

Our image generation system is based on metaphors from biological ecosystems. “Ecosystemic” models have become popular in generative art over recent years due to their ability to generate complex structures and interactions through carefully coupled feedback components [7], [10], [26], [28]. Our primary concern 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, one which 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; The animations are produced by a collection of cellular automaton-like agents operating on a grid of coloured pixels, which form the environment. Agents’ behaviours are coupled via their mutual modification of this environment [12].

Similarly, the system used to generate images in this paper, *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 world⁴. 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.

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 values⁵, 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, sufficient time for each agent to (theoretically) visit every pixel in the image. The final state of the image is its phenotype. The mapping between individual and grid to phenotype is deterministic, however different grid sizes lead to different phenotypes.

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

⁴There is also some similarity to ant-based painting (*e.g.*, [2]), which also relies on the interactions of agents in a shared colour-world. The primary difference is that *EvoEco* agent behaviours are more general than pheromone-based motion, and that *EvoEco* agent behaviours do not necessarily relate to other agents’. We instead supply heuristics through the agent input-output mappings.

⁵Typically we use a size of 224×168 , chosen for a 4:3 aspect ratio, and such that a square of 4×4 individuals can be displayed easily on a monitor of resolution 1024×768 .

can be viewed as an animation, proceeding from a simple flat colour to the final image.

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 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

- $sD \in \{0, \dots, 8\}$, is a starting direction.
- $delay \in [0, 0.5]$, and $stop \in [0.5, 1]$, which indicate when an agent should start or stop execution during its lifespan.
- (a_H, a_S, a_B) , and a_{dir} , four GP-trees, described below. These four trees are used to determine the colour that an agent draws in the current pixel, and the direction it travels next.
- g , t_1 , and t_2 are floating values $\in [0, 1]$. The former is a general-purpose agent-specific constant value, usable for any purpose, and the latter two are time periods, used in looping functions, allowing individuals to execute periodic behaviour.

An agent's first step is to collect a description of its Moore neighbourhood (see Figure 2). An agent will collect twenty-six variables, each normalized to a value in $[0, 1]$. These are listed in Table I.



Fig. 2. Moore neighbourhood with directions.

An agent's internal state variables are the colour triple (a_H, a_S, a_B) and the direction a_{dir} . All are represented by a floating point number in $[0, 1]$, a_{dir} being linearly scaled to a direction in $\{0, \dots, 8\}$. At each time step, (a_H, a_S, a_B) will determine the colour the agent draws in the current pixel, and a_{dir} the direction it travels next.

Each of these internal state variables is determined through a function of the aforementioned 26 inputs. A natural choice for doing so is tree-based genetic programming. Expecting relatively simple agent behaviours to be sufficient, and desiring to encourage evolvability and avoid bloat, we elected to use a static tree size and topology. The mapping is accomplished through four complete 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. The function set used is shown in Table II. Other choices of function, including exponentiation, absolute value, and some conditionals were excluded due to infrequent use or because they exhibited erratic behaviour in early tests. 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 state variable trees is shown in Figure 3. Some redundancy (*i.e.*, neutral code, or "junk DNA")

TABLE II
A LISTING OF THE FUNCTION SET USED IN THE GP TREES.

function	action (on two inputs, x and y)
<i>add</i>	$x + y$
<i>sub</i>	$x - y$
<i>mult</i>	xy
<i>const</i>	Returns the associated constant value.
<i>max</i>	$\max\{x, y\}$
<i>min</i>	$\min\{x, y\}$
<i>sin</i>	$\sin(x)$
<i>id</i>	x
<i>div</i>	$\frac{x}{y}$, or 1, if $y < 0.0001$.
<i>incdec</i>	$x + \frac{1}{8}$ if $y > 0.5$, $x - \frac{1}{8}$ ow.
<i>mean</i>	$\frac{x+y}{2}$

has been removed (*e.g.*, the second child of *sine* functions, unreferenced constant values, etc.).

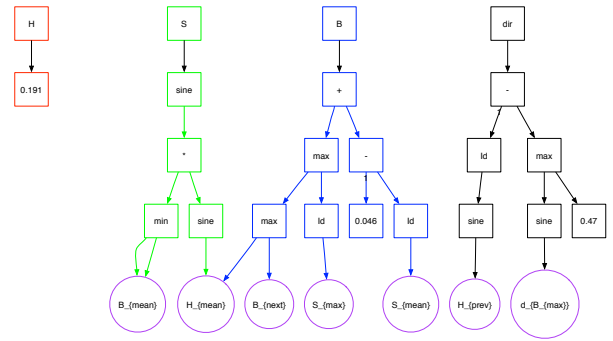


Fig. 3. An example of an agent's trees, with redundancy removed, visualized as a tree. This same individual can be written in mathematical notation as:

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

The generation of individuals via random initialization or genetic operators is detailed in Appendix A. Figure 4 shows the phenotypes produced by a sample of randomly generated individuals. We may also describe the behaviour of random initialization of such an individual in feature space: they have mean values (and standard deviations) for $var(H)$ of 0.1043 (0.1108); for $var(S)$ of 0.1007 (0.1087); and for $E(B)$ of 0.2241 (0.2622).

2) *Environmental Size*: Since the system is ecosystemic, changing the size of the environment will have potentially large consequences on the developed phenotype. This redevelopment is shown for some select randomly generated genotypes in Figure 4. 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). In some few cases, there exist some completely new patterns at some environmental size that do not exist at another (C).

To test this more formally, we generated 10,000 random

TABLE I
A LISTING OF THE 26 INPUTS COLLECTED AT EVERY TIME STEP BY ANY GIVEN AGENT.

input name(s)	description
(H_C, S_C, B_C)	The H , S and B values of pixel at the agent's current location.
(H_P, S_P, B_P)	The H , S and B values of the pixel previously visited by the agent.
$(H_{mean}, S_{mean}, B_{mean})$	The mean H , S and B values in the agent's neighbourhood.
B_{max}, B_{min}	The maximum and minimum B values in the agent's neighbourhood.
$d_{B_{max}}$, and $d_{B_{min}}$	The directions in which B_{max}, B_{min} occur.
a_g	A genetically-specified parameter.
d_P	The agent's previous direction.
d_{mD}	The direction of maximum distance (Euclidean distance in HSB space) between the current pixel and all neighbours.
$rand$	A random number chosen uniformly in $[0, 1]$.
$loop(t, a_{t_1}), loop(t, a_{t_2})$	Where t is the agent's development time, a_{t_i} is a genetically specified parameter, and $loop(t, r) = t \bmod (r \cdot 2wh)$.
(ind_H, ind_S, ind_B)	The individual's genetically-specified initial image colour.
d_{rB} and d_{rW}	A randomly chosen direction pointing at a black and white pixel, respectively, or -1 if none exists.
d_{WBN} and d_{BWN}	A randomly chosen direction pointing at a white pixel with at least one black neighbour, and a randomly chosen direction pointing at a black pixel with at least one white neighbour, respectively. In either case if no such pixel exists, a random direction is returned.

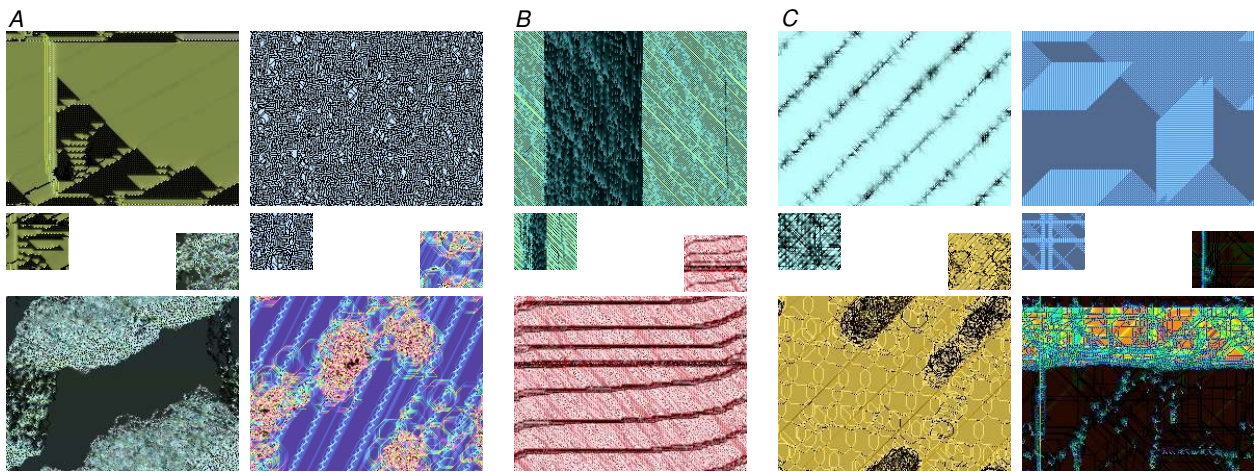


Fig. 4. Examples of select randomly-generated genotypes grown at two different sizes.

genotypes, and computed the phenotypes in environments of size 224×168 (large) and 60×54 (small). The phenotypes were contrasted using our measure of phenotypic distance. 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). It is clear that there is significant relation between the phenotypes of the same genotype, which we interpret to mean that smaller phenotypes can be used as a stand-in for larger phenotypes in our statistical tests for creativity.

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 desired 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 — *i.e.*, one single user-interaction per generation — from which any next generation individual was generated through mutation or crossover with the preceding individual in its place. Below, we describe such an IEA.

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 selected individual from the population or the

history will be referred to as *the Khan*⁶.

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 (*i.e.*, a proportion of $\frac{1}{16}$ of the population) is added to the population via one of three special search techniques, discussed below; Next, the remainder of the population is generated via *crossover*, with probability 0.33, or otherwise, via mutation of the Khan (hence, we expect 0.6075 of the population to be mutations 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).

C. Search Types

We explored three search strategies, random-search (**RAND**), creativity-search (**CS**), and phenotypic distance-search (**PD**). In each case, one of the sixteen children in the population was generated by the search mechanism. Every user of *EvoEco* was randomly assigned a search technique at the beginning of their run.

In the case of creative search, sixteen individuals were generated from the preceding population (or randomly, in the first generation). For each of those generated potential individuals, ten children were created via mutation. The potential individual with the maximal *creativity lite* score over its ten children was selected for inclusion in the new population.

In the case of phenotypic distance-search, 160 individuals were generated via mutation from the current population. The one which maximized phenotypic distance to the memory was chosen.

In the case of random search, a completely new individual was initialized and added to the population. 160 individuals were still computed in the background and ignored to ensure that the running time was consistent between the search techniques, *i.e.*, that users would not select the RAND technique simply because it was faster.

IV. USER STUDY

A. Methodology

The *EvoEco* system was instantiated as a Java applet, and linked from a webpage. The webpage was advertized via email, flyers, 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. Indeed, our advertising was aimed at reaching an artistically-literate audience.

Users were presented with a screen of the sixteen individuals, developing over time, along with a history panel showing individuals previously chosen. The history panel allows users to potentially revisit earlier, potentially successful images from the evolutionary tree, allowing the possibility of escaping from an “evolutionary dead end” at any stage in the evolutionary process. Users could select any of the sixteen or more options as the Khan, possibly before animation of the phenotype was complete. In these cases, the state of the phenotype at the time of clicking was used as the evaluated image for the individual. Users were allowed to perform interactive evolution for as long as desired.

Once the user finished, they were taken to a survey, a series of close-ended multiple choice questions regarding their gender, age, education level, residence, involvement with electronic arts, and whether or not they had previously encountered *EvoEco*. These questions were included as a measure of how representative our sample was. Next, the survey included six five-point Likert-style scales:

The software used your selections to generate a final image (shown at the top left). That final image was:

- | | | |
|----------------|-----|-----------------|
| 1. Appealing | ... | 5. Unappealing |
| 1. Creative | ... | 5. Not creative |
| 1. Interesting | ... | 5. Dull |

As the program proceeded, the software generated sets of images for your consideration. The suggested images were:

- | | | |
|--------------|-----|-----------------|
| 1. Appealing | ... | 5. Unappealing |
| 1. Creative | ... | 5. Not creative |
| 1. Novel | ... | 5. Repetitive |

Users were next asked the yes/no/uncertain question “did you feel that you could control the quality of the images through your selections?” Finally, they were asked if any technical difficulties interfered with their enjoyment of the system.

These survey results were logged, along with the genome of the final individual and some summary statistics of usage.

B. Results

Approximately 40% of respondents who started the *EvoEco* applet did not complete the survey (although more visited the website without starting the applet at all). Users who answered “no” to the question of whether they were encountering *EvoEco* for the first time were excluded. Additionally, approximately 5% of responses were excluded out of hand⁷.

As expected, the sample was not representative of the population as a whole. The users were largely young (appr. 50% were aged 18 – 35), male (66%), urban (70%), and overwhelmingly well educated (91% indicating college or professional training). Approximately half indicated some involvement with the electronic arts. Many of these biases are consistent with the placement of the survey in a gallery in a highly urban centre, and its advertisement via academic forums.

⁷A few of these were excluded because users explicitly said they experienced technical difficulties. Others were excluded on the basis that the user undertook less than three generations (often zero generations) and filled out the survey. We are uncertain why this behaviour occurred, but a theory is that some users tried the system, exited without completing the survey, then returned later to complete it.

⁶Named in honour of Genghis Khan, who geneticists believe to be a direct male-lineage ancestor of approximately 0.5% of the world's population [39]. It's good to be the Khan!

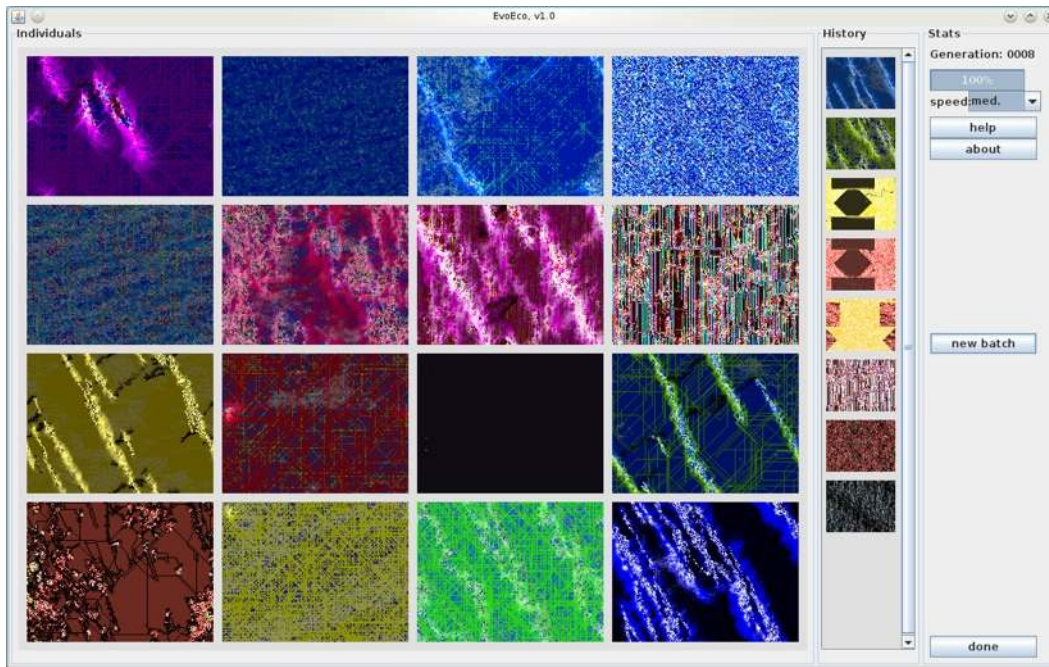


Fig. 5. A screenshot of the EvoEco Applet. On the left is the population of sixteen individuals; towards the right is the history, showing the previous eight choices of Khan; on the far right are the user controls.

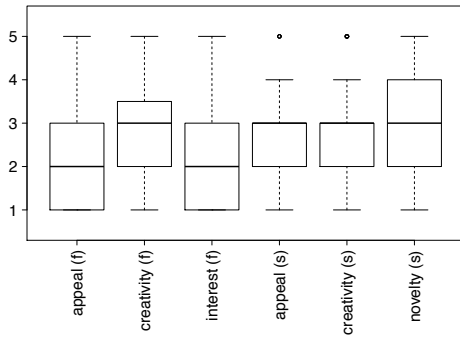


Fig. 6. User responses to questions, where (f) refers to the final image, and (s) refers to the suggested images. Lower is better.

We obtained a total of 96 usable responses, consisting of 30 RAND responses, 41 CS responses, and 25 PD responses. User responses to the Likert-scales, overall, are summarized in Figure 6. Some examples of resulting phenotypes are shown in Figure 8.

Figure 7 shows a histogram of the number of generations (*i.e.*, how long users were willing to play with the system before clicking “done”). The median number was 9.5, and the mean 12.95 (s.d. 9.44). It is evident that the majority of users’ runs lasted for approximately the median time length, while a sizeable minority used the system for substantially longer. There is no strong correlation between the reported appeal, creativity, or novelty of either the final nor suggested images with the number of generations, in all cases, the absolute value of the Spearman correlation is less than 0.1.

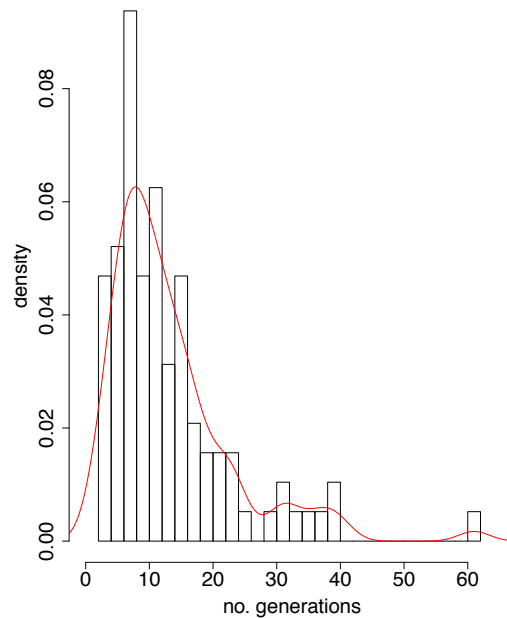


Fig. 7. Histogram of the number of generations undertaken by users.

1) *Selection Method:* We tracked the selections made by each user, including the means by which the selected individual was generated. Hence, we have a measure of the popularity of the modes of generating new population members. However, as in most evolutionary algorithms, the various means have differing probabilities of selection. Additionally, users have the option of generating a population via *population spawn* (*i.e.*, complete re-initialization) or by selecting an indi-

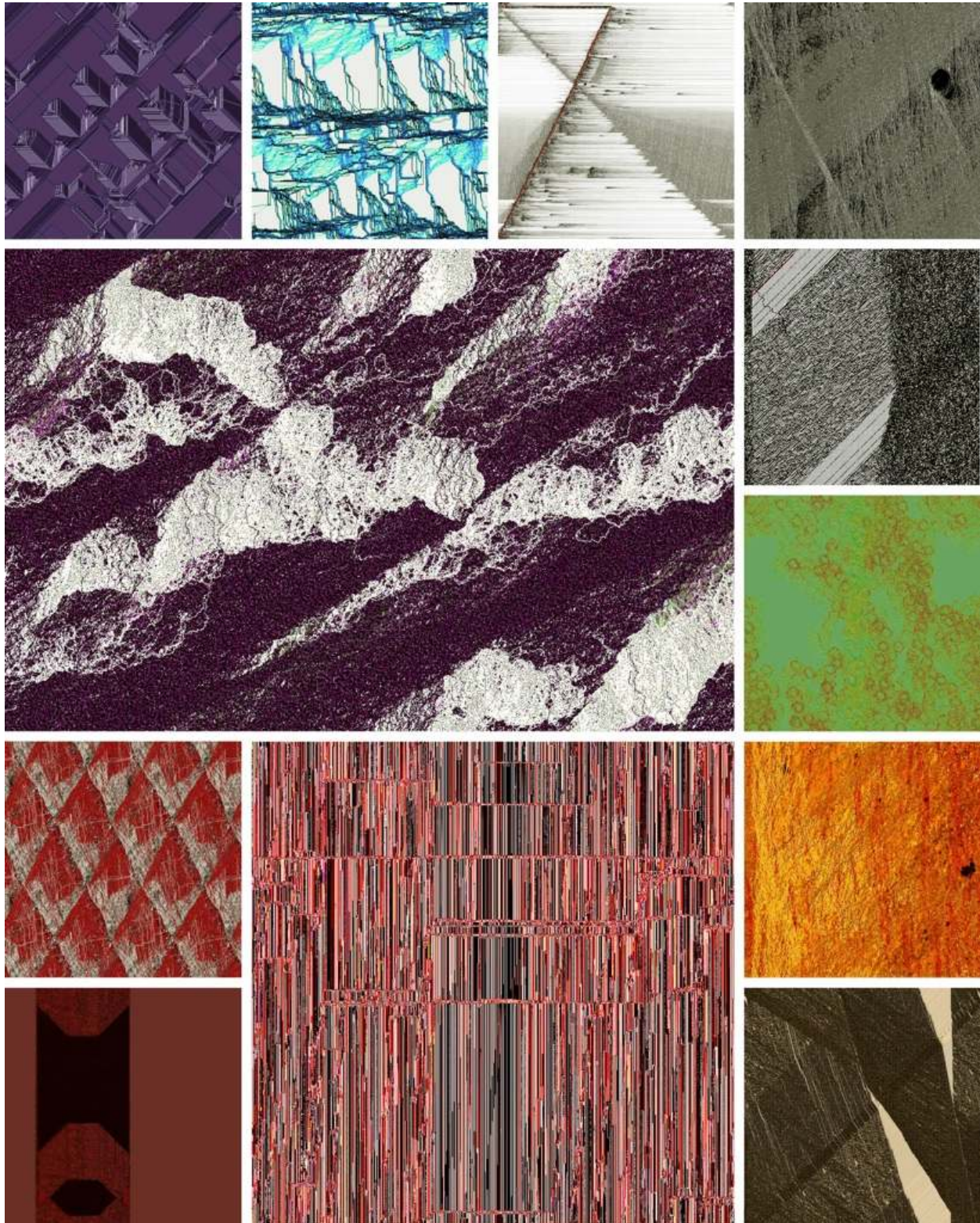


Fig. 8. Select examples produced by the authors and ninety-six anonymous users. Some images, here and throughout the paper, have been subjected to minor colour modifications to encourage visibility during grayscale printing. See <http://www.csse.monash.edu.au/~cema/evoeco> for original and high-quality colour examples.

vidual from the *history*. Hence, we expect completely random selection to choose crossover, mutation, and other genetic operators with slightly less probability than 0.33, 0.6075, and 0.0625, respectively. These values are marked with reference lines in the following plots.

An important note regarding creativity-search is that the measure does not select for a novel individual directly, but instead selects for an individual which has the capacity to generate novel children. Hence, individuals selected by creativity search need not be particularly interesting on their own, but their addition to the population can still have beneficial effects, via the children they spawn.

Figure 9 shows the various proportions of selected individual generation methods over runs organized by search type. From this figure a few trends emerge.

Firstly, we note that selection of individuals generated by crossover is approximately at expected level, indicating that crossover generates popular children. Mutation, on the other hand, is selected significantly less than is presented in populations.

Secondly, we note that of the special search types, only the RAND technique is selected with any noticeable frequency. This lack of selection indicates that both techniques generate unpopular outputs, at least in the immediate generation.

Finally, we note that users rarely select from the history panel, however, given that the UI separates the history panel from the main population, we cannot exclude UI-based factors in this result.

2) *Genetic Programs*: At the end of every run the frequency of the function and input types present in the population was recorded. Below, we present the frequencies of these various inputs and functions, where an increase in frequency shows that that particular gene type has been selected for by subjective evolution. Since our data comes from many independent runs, we generally make the assumption that the relative presence of particular inputs or GP functions in the population implies active selection, namely, that functions which are components of pleasing results will be relatively well represented, and that functions which do not contribute to the generation of pleasing results (or worse, which inhibit pleasing results) will be relatively poorly represented (We note, however, that the contribution of any particular GP function or input is likely indirect, and hence epistatic effects might affect the generalizability of these results to other systems where other sub-components have been excluded). Due to mutation and random search, we do not expect any values to ever fall entirely to zero.

Figure 10 shows the proportions of input types present in the final generations. We have split the input types into two groups, those which we expect to be useful in the generation of the colour, and those expected to be useful in the generation of the direction. Since there are three trees for colour (a_H, a_S, a_B) and only one for direction (a_{dir}), we naturally expect colour-related inputs to be referenced more. The most widely used inputs utilized the H and S colour dimensions. Surprisingly, the genetically-defined individual colour was rarely used. We also note that the edge-following behaviours d_{WBN} and d_{BWN} are the most frequently used

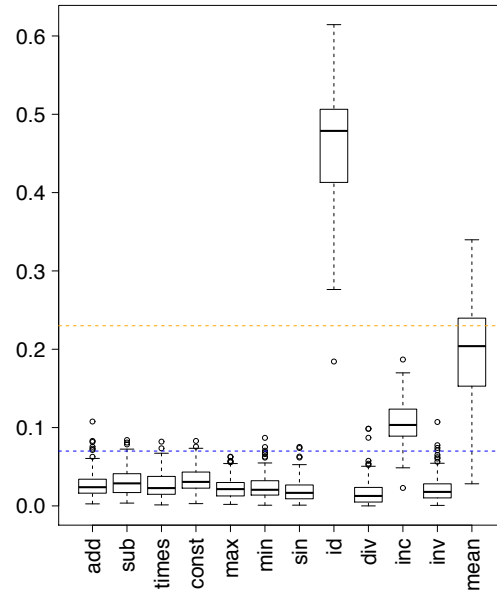


Fig. 11. Absolute proportions of GP-functions in the final population.

direction inputs.

Figure 11 shows the proportions of function types present in the final generations. Note that due to the distribution used in initialization and mutation, we would expect pure chance to return probabilities of approximately 0.07 for each function save *id*, with probability 0.23. The *id* function is used substantially, unsurprisingly given its role as a structural component in the trees. Both the *mean* and *incdec* functions are represented significantly more than would be expected via random chance, however, indicating their role in popular individuals. Of the remaining functions, none are significantly less represented than the median. We consider this indication that all were at least partially selected for, indicating that our informal choices during the design phase were generally well founded.

3) *User Preferences by Search Type*: Our primary question concerned users' stated preferences for the system conditioned by search type. In Table III we show the scores on the questions broken down by the search mechanism. To the right, we show the significance by which the mean responses differ, as determined by Welch's t-test. We will consider a p-value of less than 0.05 as highly significant, and 0.1 as somewhat significant.

Our first note is that phenotypic distance, generally, is an impediment to evolutionary success. The PD technique was ranked somewhat worse than the RAND technique in terms of the novelty of the suggested images, and significantly worse than the CS technique in three of six scales. This is intuitive, since the phenotypic distance measure serves largely to push populations away from the direction selected by the users, effectively undoing the effects of evolution.

As noted, the CS technique significantly outperforms the PD technique. Additionally, it is ranked better than the RAND technique in both the novelty of the final image, and the cre-

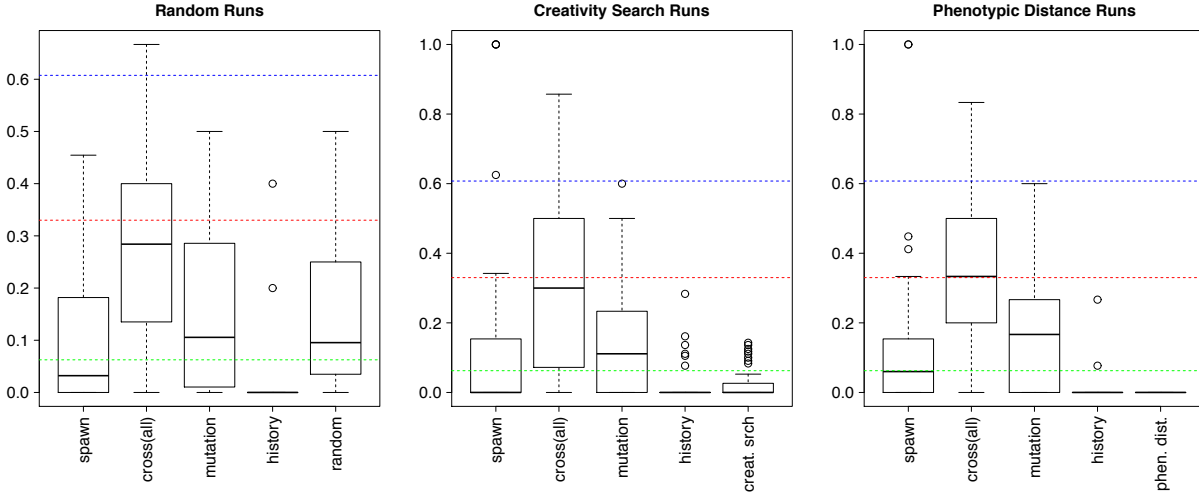


Fig. 9. Means of generation of selected individuals for (left) random-search runs, (middle) creativity-search runs, and (right) phenotypic distance runs.

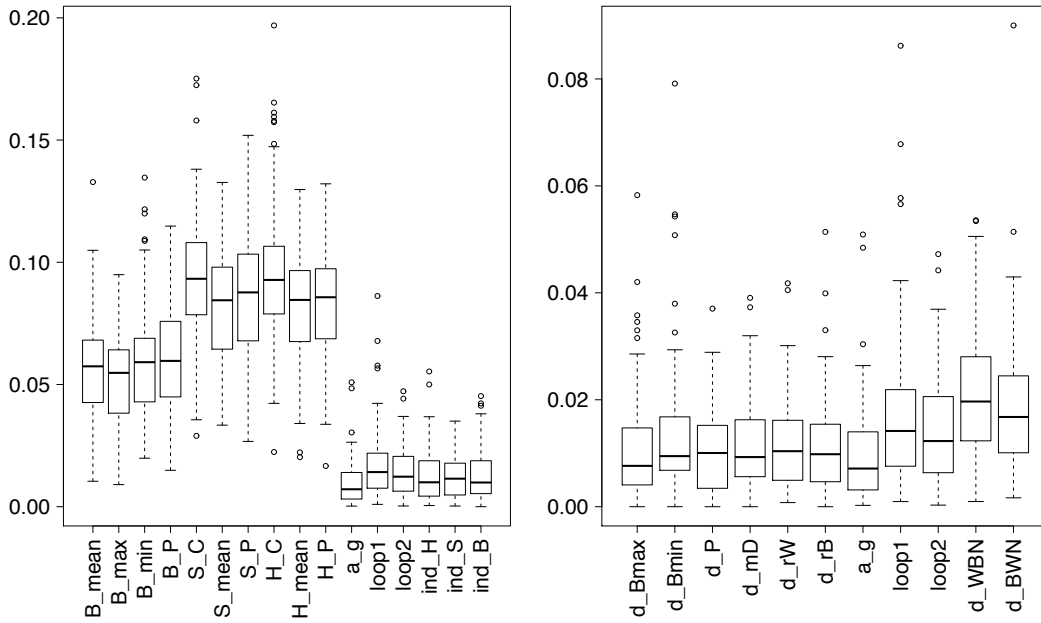


Fig. 10. Absolute proportions of all (left) colour-related and (right) direction-related inputs in the population.

TABLE III
BREAKDOWN OF USER RESPONSES ORGANIZED BY SEARCH TYPE. LOWER IS BETTER.

question	mean			p-value			
	RAND	CS	PD	CS vs RAND	CS vs PD	RAND vs PD	
Final image:	appealing	2.31	2.10	2.40	0.43	0.26	0.77
	creative	2.76	2.58	2.86	0.92	0.23	0.61
	novel	2.34	1.89	2.58	0.09	0.02	0.47
Suggested images:	appealing	2.73	2.40	2.67	0.26	0.32	0.83
	creative	2.77	2.23	3.13	0.08	< 0.01	0.27
	novel	2.72	2.42	3.43	0.41	< 0.01	0.06

ativity of the suggested images. We further note an abundance of more informal evidence supporting the use of creativity-search. The mean value for *all* responses was minimized for the CS technique. Further, the proportion of users who answered positively to the question “did you feel that you could control the quality of the images through your selections” was higher for CS (55%) than for the other two techniques (31% for RAND and 40% for PD).

Hence, we conclude that the use of the *creativity lite* measure has improved the performance of our IEA with respect to natural language notions of novelty and creativity.

4) *Comparison between Creativity Definition and User Responses*: We next chose to investigate the correlation between users’ stated responses and the formal Dorin / Korb creativity definition. To do so, we contrasted user responses on evolved individuals (in the context of the evolutionary run that produced them) against the computed creativity of those same individuals against a randomly generated memory. The comparison of an individual against a random memory is a comparison of the level of specialization of that individual’s children, as opposed to the sorts of behaviour generated by trivial genomes. Our hypothesis is that users are capable of discerning the difference, either due to their developed familiarity with the behaviours of randomly generated individuals via the preceding exploration of the system, or, perhaps, by a more universal familiarity of the sorts of behaviours that are likely output by simple machines.

For each survey response, the final individual was extracted. Five hundred mutated children were computed and compared to a memory of 15,000 randomly generated individuals. If a creative region is found via the technique detailed in Section II-B1, then the individual was labelled creative by our definition. Figure 12 shows an example of an individual deemed creative by both the user and the definition.

We next computed the Spearman correlation ρ between these labels and the user-stated rankings. For most rankings, we did not obtain significant relation (p-values associated with the correlations were high). In the case of the correlation with the novelty of the suggested images, we obtained $\rho = 0.253$ with a p-value of 0.05. In the case of the correlation with the creativity of the final image, we obtained a weaker result: $\rho = 0.144$ with a p-value of 0.2.

We interpret these results to mean that there exists a weak relation between users’ stated opinion of the creativity and novelty of discovered individuals and the score by our technical definition.

We must note that the survey itself left the ratings of creativity and interest as purposefully ambiguous: creative relative to the starting point? To a user’s intuition with the tool at hand? To web-based art more generally? To even more universal standards? Our measurements are based relative to other users under similar ambiguity, and hence useful as a guide to performance in a generative art task. To tie this to a more unambiguously universal natural language notion of “creativity”, however, requires more research.

V. SUMMARY

In this paper, we have analyzed users’ stated impressions of the “creativity” and “novelty” of art produced via an ecosystemic IEA.

We have provided several results that likely extend to the design of interactive electronic art systems generally, including: (a) that crossover is a useful operator in ecosystemic systems; (b) that the GP functions *mean* and *incdec* are popular choices in an GP-based task; and (c) that in a pixel-based local neighbourhood, edge-following behaviours and inputs based on the H and S channels are important choices. In the case of the use of particular GP functions and inputs, we cannot exclude the possibility of epistatic effects, meaning that change of context may change the usefulness of the functions in question.

We also analyzed the use of three strategies to augment the evolutionary search, reaching strong conclusions. Firstly, that the introduction of a phenotypic distance function is a liability in evolutionary art tasks. In this case, phenotypic distance significantly worsened the performance of the system relative to our other techniques, including a random search. Secondly, that the inclusion of a simplified Dorin / Korb measure of creativity significantly improves a user’s stated opinion of the “creativity” and “novelty” of the system. We have provided additional informal evidence to suggest that it results in a better outcome generally.

Finally, we have shown that there exists a weak correlation between our technical definition of creativity and users’ stated opinions. This may be unambiguously read as saying that an interpreted version of the Dorin and Korb definition captures some aspect of the natural language usage of the term in a generative art task — this is an important matter for a value-free definition!

APPENDIX

A. Individual Initialization

We recall that an individual is the $(k + 4)$ -tuple:

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

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, 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 $\in \{0, \dots, 26\}$.
- $2^{d-1} - 1$ non-leaf nodes representing functions in the tree, represented by integers $\in \{0, \dots, 11\}$.

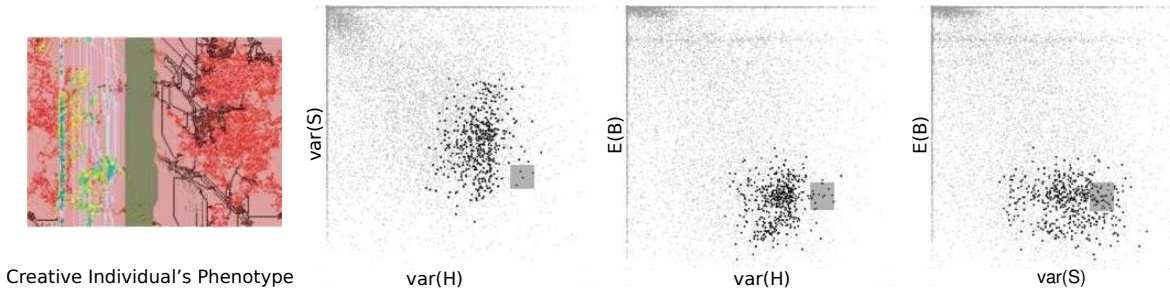


Fig. 12. Example of an individual deemed creative by both the user and the Dorin / Korb definition, along with an illustration of (two-dimensional projections of) feature space. The memory data points are drawn in light grey, the children of the individual in black, and the creative region is overlaid as a medium-grey rectangle.

- $2^{d-1} - 1$ 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. In these cases, each function has a probability of being chosen of 0.07, 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, and hence to be of greater importance.

B. Genetic Operators

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.

We can apply *crossover* to a pair of agents, a^1 and a^2 , by performing sub-tree crossover on a particular tree. 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 some individual $I^1 = (k^1, (ind_H^1, ind_S^1, ind_B^1), a_1^1, \dots, a_k^1)$ and a probability of mutation *probMut*, we can generate a new individual I^2 via *mutation*:

```

 $k^2 = k^1$ 
 $ind_X^2 = ind_X^1$ 
if ( $rand < probMut$ ) then
   $ind_X^2 = ind_X^1 + \epsilon(0.2)$ 
end if
for  $i \in \{1, \dots, k^2\}$  do
   $a_i^2 = mutate(a_i^1, probMut)$ 
end for
if ( $rand < probMut \wedge k^2 < 15$ ) then
   $k^2 = k^2 + 1$ 
   $a_{k^2}^2 = \text{new random agent}$ 
else if ( $rand < probMut \wedge k^2 < 15$ ) then
   $k^2 = k^2 + 1$ 

```

```

 $a_{k^2}^2 = clone(a_{urand(\{1, \dots, k^1\})}^1)$ 
else if ( $rand < 2 \cdot probMut \wedge k^2 > 2$ ) then
   $k^2 = k^2 - 1$ 
  delete  $a_{urand(\{1, \dots, k^1\})}^1$  and re-index
end if

```

where $\epsilon(z)$ is a function that generates Gaussian noise with variance z . Increase and decrease of size are equally likely by this operator, but we note that the smaller initialization maximum (six) compared to overall maximum number of agents (fifteen) implies growth in the number of agents over time independent of selection pressures. Two forms of agent addition were included to encourage both duplication of desirable properties and increase in complexity.

Given two individuals, $I^1 = (k^1, (ind_H^1, ind_S^1, ind_B^1), a_1^1, \dots, a_{k^1}^1)$ and $I^2 = (k^2, (ind_H^2, ind_S^2, ind_B^2), a_1^2, \dots, a_{k^2}^2)$, 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

```

REFERENCES

- [1] D. Ashlock. Evolutionary design in embryogeny. In P.F. Hingston *et al.*, editor, *Design by Evolution: Advances in Evolutionary Design*, pages 199–202. Springer-Verlag, 2008.
- [2] S. Aupetit, V. Bordeau, N. Monmarché, M. Slimane, and G. Venturini. Interactive evolution of ant paintings. In A. Tyrrell, editor, *CEC'03: Proceedings of the 2003 Congress on Evolutionary Computation*, pages 1376–1383, 2003.
- [3] S. Baluja, D. Pomerleau, and T. Jochem. Towards automated artificial evolution for computer-generated images. *Connection Science*, 6(1):325–354, 1994.
- [4] M.A. Boden. *The creative mind: myths & mechanisms*. Number xii, 303. Basic Books, New York, N.Y., 1991.
- [5] A.M. Brintrup, J. Ramsden, H.H. Takagi, and A. Tiwari. Ergonomic chair design by fusing qualitative and quantitative criteria using interactive genetic algorithms. *Evolutionary Computation, IEEE Transactions on*, 12(3):343–354, june 2008.
- [6] R. Dawkins. *Climbing Mount Improbable*. W.W. Norton & Company, 1996.
- [7] A. Di Scipio. ‘sound is the interface’: from interactive to ecosystemic signal processing. *Organised Sound*, 8(3):269–277, 2003.
- [8] S. DiPaola and L. Gabora. Incorporating characteristics of human creativity into an evolutionary art algorithm. *Genetic Programming and Evolvable Machines*, 10(2):97–110, 2009.

- [9] A. Dorin. Aesthetic fitness and artificial evolution for the selection of imagery from the mythical infinite library. In Kelemen and Sosik, editors, *Advances in Artificial Life, Proc. 6th European Conference on Artificial Life*, volume LNAI 2159, pages 659–668. Springer Verlag, 2001.
- [10] A. Dorin. The virtual ecosystem as generative electronic art. In G.R. et. al. Raidl, editor, *Proceedings of 2nd European Workshop on Evolutionary Music and Art, Applications of Evolutionary Computing: EvoWorkshops 2004*, pages 467–476, Heidelberg, Germany, 2004. Springer-Verlag.
- [11] A. Dorin and K. Korb. Improbable creativity. In J. McCormack, M. Boden, and M. d'Inverno, editors, *Proceedings of the Dagstuhl International Seminar on Computational Creativity*. Springer, 2009.
- [12] E. Driessens and M. Verstappen. Natural processes and artificial procedures. In P. F. Hingston, L. C. Barone, and Z. Micalewicz, editors, *Design by Evolution: Advances in Evolutionary Design*. Springer, 2008.
- [13] G. Greenfield. Co-evolutionary methods in evolutionary art. In Juan Romero and Penousal Machado, editors, *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*, pages 357–380. Springer Berlin Heidelberg, 2007.
- [14] R. Haralick. Statistical and structural approaches to texture. *Proceedings of the IEEE*, 67:786–804, 1979.
- [15] D. Hart. Toward greater artistic control for interactive evolution of images and animation. *Applications of Evolutionary Computing*, pages 527–536, 2007.
- [16] P. Howarth and S. Rüger. Evaluation of texture features for content-based image retrieval. In *CIVR 2004, LNCS 3115*, pages 326–334, 2004.
- [17] F.C. Hsu and P. Huang. Providing an appropriate search space to solve the fatigue problem in interactive evolutionary computation. *New Generation Computing*, 23(2):115–127, 2005.
- [18] 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)*, volume 5865, pages 1–10. Springer, 2009.
- [19] K.N. Laland, F.J. Odling-Smee, and M.W. Feldman. Evolutionary consequences of niche construction and their implication for ecology. *Proceedings of the National Academy of Science*, 96:10242–10247, 1999.
- [20] M. Lewis. Evolutionary visual art and design. In J. Romero and P. Machado, editors, *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*, pages 3–37. Springer, 2008.
- [21] X. Llorà, K. Sastry, D.E. Goldberg, A. Gupta, and L. Lakshmi. Combating user fatigue in igas: partial ordering, support vector machines, and synthetic fitness. In *GECCO '05: Proceedings of the 2005 conference on Genetic and evolutionary computation*, pages 1363–1370. ACM, 2005.
- [22] E. Lutton. Evolution of fractal shapes for artists and designers. *International Journal on Artificial Intelligence Tools*, 15(4):651–672, 2006.
- [23] P. Machado and A. Cardoso. All the truth about nevar. *Applied Intelligence*, 16:101–118, January 2002.
- [24] P. Machado, J. Romero, and B. Manaris. Experiments in computational aesthetics: An iterative approach to stylistic change in evolutionary art. In J. Romero and P. Machado, editors, *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*, pages 381–415, 2008.
- [25] A.T. Machwe and I.C. Parmee. Reducing user fatigue within an interactive evolutionary design system using clustering and case-based reasoning. *Engineering Optimization*, 41(9):871–887, 2009.
- [26] J. McCormack. Eden: An evolutionary sonic ecosystem. *Advances in Artificial Life, Proceedings of the Sixth European Conference, ECAL, LNCS 2159*:133–142, 2001.
- [27] J. McCormack. Open problems in evolutionary music and art. In *Applications on Evolutionary Computing*, volume 3449, pages 428–436. Springer-Verlag, 2005.
- [28] J. McCormack. Creative ecosystems. In A. Cardoso and G.A. Wiggins, editors, *Proceedings of the 4th International Joint Workshop on Computational Creativity*, pages 129–136, 17–19 June 2007 2007.
- [29] N. Monmarché, I. Mahnich, and M. Slimane. Artificial art made by artificial ants. In J. Romero and P. Machado, editors, *The Art of Artificial Evolution: A Handbook on Evolutionary Art and Music*, pages 227–247. Springer, 2008.
- [30] I.C. Parmee. Evolutionary design search, exploration and optimisation. In *GECCO '07: Proceedings of the 2007 GECCO conference companion on Genetic and evolutionary computation*, pages 3508–3536. ACM, 2007.
- [31] M. Rönkkö. An artificial ecosystem: Emergent dynamics and lifelike properties. *Artificial Life*, 13(2):159–187, 2007.
- [32] J. Secretan, N. Beato, D.B. D'Ambrosio, A. Rodriguez, A. Campbell, and K.O. Stanley. Picbreeder: Collaborative interactive evolution of images. *Leonardo*, 41(1):98–99, 2008.
- [33] A. Serag, S. Ono, and S. Nakayama. Using interactive evolutionary computation to generate creative building designs. 13(1):246–250, 2008.
- [34] J. Shen. Stochastic modelling western paintings for effective classification. *Pattern Recognition*, 42:293–301, 2009.
- [35] H.H. Takagi. Interactive evolutionary computation: fusion of the capabilities of ec optimization and human evaluation. *Proceedings of the IEEE*, 89(9):1275–1296, sep 2001.
- [36] H.H. Takagi and M. Ohsaki. Interactive evolutionary computation-based hearing aid fitting. *Evolutionary Computation, IEEE Transactions on*, 11(3):414–427, June 2007.
- [37] S. Todd and W. Latham. *Evolutionary Art and Computers*. Academic Press Ltd., 1992.
- [38] A. Vailaya, M.A.T. Figueiredo, A.K. Jain, and Hong-Jiang Zhang. Image classification for content-based indexing. *Image Processing, IEEE Transactions on*, 10(1):117–130, Jan 2001.
- [39] 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*, 72(3):717 – 721, 2003.

AUTHOR BIOGRAPHIES

Taras Kowaliw (M '10) received the B.Sc. in mathematics from the University of Toronto in 2000, and the M.Sc. and Ph.D. degrees in computing from Concordia University in Montréal in 2003 and 2007.

He has been a Postdoctoral Fellow with the Department of Computer Science at Memorial University of Newfoundland and a Research Associate with the Faculty of Information Technology at Monash University. Presently, he is a postdoctoral fellow with the Institut des systèmes complexes - Paris Île-de-France, Centre national de la recherche scientifique. His research interests include machine learning, artificial life, computer vision, and the electronic arts.

Alan Dorin received the B.Sc. in mathematics (Monash University, 1991), Comp. Sci. Honours (Monash University, 1992), the Post. Grad. Dip. Animation and Interactive Multimedia (R.M.I.T, 1995) and the Ph.D. in Comp. Sci. (Monash University, 1999).

He is currently a Senior Lecturer in Computer Science in the Faculty of Information Technology at Monash University, where he co-directs the Centre for Electronic Media Art (CEMA), and is an electronic media artist. His interests include artificial life, electronic art, computer animation, computer music, virtual and organic biology and ecology, self-assembling systems, the history and philosophy of science and art.

Jon McCormack received the B.Sc. in applied mathematics and computer science from Monash University in 1987, the Graduate Diploma of Art from Swinburne University in 1986, and the Ph.D. in computer science from Monash University in 2004.

He is currently Associate Professor in Computer Science, co-director of CEMA at Monash University and visiting research fellow at Goldsmiths, University of London. His research interests include generative evolutionary systems, computational creativity, artificial life, L-systems and developmental models. His artworks are held in the collections of leading museums in the USA, Europe and Australia.

CEMA is an interdisciplinary research centre established to explore new collaborative relationships between computing and the creative arts.